Zbornik 28. mednarodne multikonference

INFORMACIJSKA DRUŽBA – IS 2025

Zvezek C

Proceedings of the 28th International Multiconference

INFORMATION SOCIETY – IS 2025

Volume C

Odkrivanje znanja in podatkovna skladišča - SiKDD
Data Mining and Data Warehouses - SiKDD

Urednika / Editors

Dunja Mladenić, Marko Grobelnik

http://is.ijs.si

6. oktober 2025 / 6 October 2025
Ljubljana, Slovenia

Urednika:

Dunja Mladenić
Department for Artificial Intelligence
Jožef Stefan Institute, Ljubljana

Marko Grobelnik
Department for Artificial Intelligence
Jožef Stefan Institute, Ljubljana

Založnik: Institut »Jožef Stefan«, Ljubljana
Priprava zbornika: Mitja Lasič, Vesna Lasič, Lana Ze[illegible]ak
Oblikovanje naslovnice: Vesna Lasič

Dostop do e-publikacije:
http://library.ijs.si/Stacks/Proceedings/InformationSociety

Ljubljana, oktober 2025

Informacijska družba
ISSN 2630-371X

PREDGOVOR MULTIKONFERENCI INFORMACIJSKA DRUŽBA 2025

28. mednarodna multikonferenca *Informacijska družba* se odvija v času izjemne rasti umetne inteligence, njenih aplikacij in vplivov na človeštvo. Vsako leto vstopamo v novo dobo, v kateri generativna umetna inteligenca ter drugi inovativni pristopi oblikujejo poti k superinteligenci in singularnosti, ki bosta krojili prihodnost človeške civilizacije. Naša konferenca je tako hkrati tradicionalna znanstvena in akademsko odprta, pa tudi inkubator novih, pogumnih idej in pogledov.

Letošnja konferenca poleg umetne inteligence vključuje tudi razprave o perečih temah današnjega časa: ohranjanje okolja, demografski izzivi, zdravstvo in preobrazba družbenih struktur. Razvoj UI ponuja rešitve za številne sodobne izzive, kar poudarja pomen sodelovanja med raziskovalci, strokovnjaki in odločevalci pri oblikovanju trajnostnih strategij. Zavedamo se, da živimo v obdobju velikih sprememb, kjer je ključno, da z inovativnimi pristopi in poglobljenim znanjem ustvarimo informacijsko družbo, ki bo varna, vključujoča in trajnostna.

V okviru multikonference smo letos združili dvanajst vsebinsko raznolikih srečanj, ki odražajo širino in globino informacijskih ved: od umetne inteligence v zdravstvu, demografskih in družinskih analiz, digitalne preobrazbe zdravstvene nege ter digitalne vključenosti v informacijski družbi, do raziskav na področju kognitivne znanosti, zdrave dolgoživosti ter vzgoje in izobraževanja v informacijski družbi. Pridružujejo se konference o legendah računalništva in informatike, prenosu tehnologij, mitih in resnicah o varovanju okolja, odkrivanju znanja in podatkovnih skladiščih ter seveda Slovenska konferenca o umetni inteligenci.

Poleg referatov bodo okrogle mize in delavnice omogočile poglobljeno izmenjavo mnenj, ki bo pomembno prispevala k oblikovanju prihodnje informacijske družbe. »Legende računalništva in informatike« predstavljajo domači »Hall of Fame« za izjemne posameznike s tega področja. Še naprej bomo spodbujali raziskovanje in razvoj, odličnost in sodelovanje; razširjeni referati bodo objavljeni v reviji *Informatica*, s podporo dolgoletne tradicije in v sodelovanju z akademskimi institucijami ter strokovnimi združenji, kot so ACM Slovenija, SLAIS, Slovensko društvo Informatika in Inženirska akademija Slovenije.

Vsako leto izberemo najbolj izstopajoče dosežke. Letos je nagrado *Michie-Turing* za izjemen življenjski prispevek k razvoju in promociji informacijske družbe prejel **Niko Schlamberger**, priznanje za raziskovalni dosežek leta pa **Tome Eftimov**. »Informacijsko limono« za najmanj primerno informacijsko tematiko je prejela odsotnost obveznega pouka računalništva v osnovnih šolah. »Informacijsko jagodo« za najboljši sistem ali storitev v letih 2024/2025 pa so prejeli Marko Robnik Šikonja, Damir Vreš in Simon Krek s skupino za slovenski veliki jezikovni model GAMS. Iskrene čestitke vsem nagrajencem!

Naša vizija ostaja jasna: prepoznati, izkoristiti in oblikovati priložnosti, ki jih prinaša digitalna preobrazba, ter ustvariti informacijsko družbo, ki koristi vsem njenim članom. Vsem sodelujočim se zahvaljujemo za njihov prispevek — veseli nas, da bomo skupaj oblikovali prihodnje dosežke, ki jih bo soustvarjala ta konferenca.

Mojca Ciglarič, predsednica programskega odbora
Matjaž Gams, predsednik organizacijskega odbora

FOREWORD TO THE MULTICONFERENCE INFORMATION SOCIETY 2025

The 28th International Multiconference on the Information Society takes place at a time of remarkable growth in artificial intelligence, its applications, and its impact on humanity. Each year we enter a new era in which generative AI and other innovative approaches shape the path toward superintelligence and singularity — phenomena that will shape the future of human civilization. The conference is both a traditional scientific forum and an academically open incubator for new, bold ideas and perspectives.

In addition to artificial intelligence, this year's conference addresses other pressing issues of our time: environmental preservation, demographic challenges, healthcare, and the transformation of social structures. The rapid development of AI offers potential solutions to many of today's challenges and highlights the importance of collaboration among researchers, experts, and policymakers in designing sustainable strategies. We are acutely aware that we live in an era of profound change, where innovative approaches and deep knowledge are essential to creating an information society that is safe, inclusive, and sustainable.

This year's multiconference brings together twelve thematically diverse meetings reflecting the breadth and depth of the information sciences: from artificial intelligence in healthcare, demographic and family studies, and the digital transformation of nursing and digital inclusion, to research in cognitive science, healthy longevity, and education in the information society. Additional conferences include Legends of Computing and Informatics, Technology Transfer, Myths and Truths of Environmental Protection, Knowledge Discovery and Data Warehouses, and, of course, the Slovenian Conference on Artificial Intelligence.

Alongside scientific papers, round tables and workshops will provide opportunities for in-depth exchanges of views, making an important contribution to shaping the future information society. *Legends of Computing and Informatics* serves as a national »Hall of Fame« honoring outstanding individuals in the field. We will continue to promote research and development, excellence, and collaboration. Extended papers will be published in the journal *Informatica*, supported by a long-standing tradition and in cooperation with academic institutions and professional associations such as ACM Slovenia, SLAIS, the Slovenian Society Informatika, and the Slovenian Academy of Engineering.

Each year we recognize the most distinguished achievements. In 2025, the Michie-Turing Award for lifetime contribution to the development and promotion of the information society was awarded to **Niko Schlamberger**, while the Award for Research Achievement of the Year went to **Tome Eftimov**. The »Information Lemon« for the least appropriate information-related topic was awarded to the absence of compulsory computer science education in primary schools. The »Information Strawberry« for the best system or service in 2024/2025 was awarded to Marko Robnik Šikonja, Damir Vreš and Simon Krek together with their team, for developing the Slovenian large language model GAMS. We extend our warmest congratulations to all awardees.

Our vision remains clear: to identify, seize, and shape the opportunities offered by digital transformation, and to create an information society that benefits all its members. We sincerely thank all participants for their contributions and look forward to jointly shaping the future achievements that this conference will help bring about.

Mojca Ciglarič, Chair of the Program Committee
Matjaž Gams, Chair of the Organizing Committee

KONFERENČNI ODBORI
CONFERENCE COMMITTEES

International Programme Committee

Vladimir Bajic, South Africa
Heiner Benking, Germany
Se Woo Cheon, South Korea
Howie Firth, UK
Olga Fomichova, Russia
Vladimir Fomichov, Russia
Vesna Hljuz Dobric, Croatia
Alfred Inselberg, Israel
Jay Liebowitz, USA
Huan Liu, Singapore
Henz Martin, Germany
Marcin Paprzycki, USA
Claude Sammut, Australia
Jiri Wiedermann, Czech Republic
Xindong Wu, USA
Yiming Ye, USA
Ning Zhong, USA
Wray Buntine, Australia
Bezalel Gavish, USA
Gal A. Kaminka, Israel
Mike Bain, Australia
Michela Milano, Italy
Derong Liu, Chicago, USA
Toby Walsh, Australia
Sergio Campos-Cordobes, Spain
Shabnam Farahmand, Finland
Sergio Crovella, Italy

Organizing Committee

Matjaž Gams, chair
Mitja Luštrek
Lana Zemljak
Vesna Koricki
Mitja Lasič
Blaž Mahnič

Programme Committee

Mojca Ciglarič, chair
Bojan Orel
Franc Solina
Viljan Mahnič
Cene Bavec
Tomaž Kalin
Jozsef Györkös
Tadej Bajd
Jaroslav Berce
Mojca Bernik
Marko Bohanec
Ivan Bratko
Andrej Brodnik
Dušan Caf
Saša Divjak
Tomaž Erjavec
Bogdan Filipič
Andrej Gams
Matjaž Gams
Mitja Luštrek
Marko Grobelnik
Nikola Guid
Marjan Heričko
Borka Jerman Blažič Džonova
Gorazd Kandus
Urban Kordeš
Marjan Krisper
Andrej Kuščer
Jadran Lenarčič
Borut Likar
Janez Malačič
Olga Markič
Dunja Mladenič
Franc Novak
Vladislav Rajkovič
Grega Repovš
Ivan Rozman
Niko Schlamberger
Gašper Slapničar
Stanko Strmčnik
Jurij Šilc
Jurij Tasič
Denis Trček
Andrej Ule
Boštjan Vilfan
Baldomir Zajc
Blaž Zupan
Boris Žemva
Leon Žlajpah
Niko Zimic
Rok Piltaver
Toma Strle
Tine Kolenik
Franci Pivec
Uroš Rajkovič
Borut Batagelj
Tomaž Ogrin
Aleš Ude
Bojan Blažica
Matjaž Kljun
Robert Blatnik
Erik Dovgan
Špela Stres
Anton Gradišek

KAZALO / TABLE OF CONTENTS

Odkrivanje znanja in podatkovna skladišča – SiKDD / Data Mining and Data Warehouses - SiKDD....1

PREDGOVOR / FOREWORD 3

PROGRAMSKI ODBORI / PROGRAMME COMMITTEES 5

Semantic Prompting for Large Language Models in Biomedical Named Entity Recognition / Calcina Erik, Novak Erik, Mladenić Dunja....7

LLM Based Approach to Extracting Smells in Slovenian Corpora / Brank Janez, Novalija Inna, Mladenić Dunja, Grobelnik Marko 11

BetweenTheLines - Cross Source News Analysis / Trajkov Georgi, Grobelnik Marko, Grobelnik Adrian Mladenić....15

Identifying Social Self in Text: A Machine Learning Study / Caporusso Jaya, Purver Matthew, Pollak Senja ..19

WinWin Meets – Investigating the Future of Online Meetings / Žust Martin, Grobelnik Marko, Guček Alenka, Grobelnik Adrian Mladenić....25

Predicting Ski Jumps Using State-Space Model / Hegler Živa, Camlek Neca, Jelenčič Jakob, Grobelnik Marko, Mladenić Dunja 29

Predicting milling overload based on sensor data: a graph-based approach / Krumpak Roy, Rožanec Jože M., Mladenić Dunja, Guo Zhenyu, Song Tao, Roman Dumitru, Novalija Inna, Ma Xiang....33

Short and Long Term Bike Rental Forecasting / Kocjančič Oskar, Žnidaršič Martin 37

Predicting Traffic Intensity on Motorway Sections / Kladnik Matic, Mladenić Dunja 41

Empowering Youth for Smart Cities with AI Solutions to Community and Urban Challenges in the Context of SDG 11 / Zaouini Mustafa, Costa João Pita, Rahmani Yousef, Kassis Rayan, Stopar Luka, Souss Sohaib, Lamgari Asmai, Mochariq Ouidad....45

Automated First-Reply Generation for IT Support Tickets Using Retrieval-Augmented Generation and Multi-Modal Response Synthesis / Jeršek Domen, Kenda Klemen, Frattini Matteo, Klančič Rok 49

A Machine-Learning Approach to Predicting the Pronunciation of Pre-Consonant l in Standard Slovene / Čibej Jaka....53

Sequencing News Articles with Large Language Models within Enterprise Risk Management Context / Debeljak Žiga, Mladenić Dunja, Kenda Klemen 57

Graph-Based Feature Engineering for DeFi Security Incident Severity Prediction / Pavlova Daria, Novalija Inna, Mladenić Dunja....61

Evolving Neural Agents in Simulated Ecosystems / Ćetković Marija, Tošić Aleksandar, Vake Domen 65

Designing AI Agents for Social Media / Sittar Abdul, Smiljanic Mateja, Guček Alenka 69

Explaining Temporal Data in Manufacturing using LLMs and Markov Chains / Šturm Jan, Škrjanc Maja, Topal Oleksandra, Novalija Inna, Mladenić Dunja, Grobelnik Marko 73

Active Learning for Power Grid Security Assessment: Reducing Simulation Cost with Informative Sampling / Leskovec Gašper, Mylonas Costas, Kenda Klemen....77

Supporting Material Reuse in Drone Production / Cek Rok, Topal Oleksandra, Leonardi Linda, Forcolin Marherita, Kenda Klemen 82

Temporal Dynamics and Causal Feature Integration for Predictive Maintenance in Manufacturing Systems: ACausality-Informed Framework / Hosseini Seyed Iman, Kenda Klemen, Mladenić Dunja....86

Using Interactive Data Visualization for DeFi Market Analysis / Pavlova Daria....90

A Hybrid Lexicon-Machine Learning Approach to Macedonian Sentiment Analysis / Kochovska Sofija, Kavšek Branko, Vičič Jernej 94

Building an AI-Ready Data Infrastructure Towards a SDG-focused Observatory for the Brazilian Amazon / Costa João Pita, Polzer Mirozlav, Barrionuevo Leonardo, Veiga João Cândia 98

Towards a format for describing networks, NetsJSON / Batagelj Vladimir, Pisanski Tomaž, Savnik Iztok, Slavec Ana, Bašić Nino....102

Automating Numba Optimization with Large Language Models: A Case Study on Mutual Information / Kozamernik Lučka, Jakomin Martin, Škrlj Blaž, Urbančič Jasna....106

Topological Exploration of Embedded GitHub Repository Data Using Mapper / Hrib Ivo, Zajec Patrik....110

CO2 Monitoring for Energy-Efficient Workloads in Kubernetes: A Data Provider for CO2-Aware Migration / Hrib Ivo, Topal Oleksandra, Šturm Jan, Škrjanc Maja 114

Beyond Surveys: Adolescent Profiling via Ecological Momentary Assessment and Mobile Sensing / Dobša Jasminka, Korenjak-Černe Simona, Novak Miranda, Pandur Maja Buhin, Šutić Lucija 118
Brazil's First AI Regulatory Sandbox: Towards Responsible Innovation / Oliveira Cristina Godoy, Veiga João Cândia, Sancin Vasilka, Costa João Pita, Silva Rafael Meira, Dine Masa Kovic, Anjos Lucas Costa dos, Marcilio Thiago Gomes, Silva Anthony Novaes 122

Indeks avtorjev / Author index 127

Zbornik 28. mednarodne multikonference

INFORMACIJSKA DRUŽBA – IS 2025

Zvezek C

Proceedings of the 28th International Multiconference

INFORMATION SOCIETY – IS 2025

Volume C

Odkrivanje znanja in podatkovna skladišča - SiKDD
Data Mining and Data Warehouses - SiKDD

Urednika / Editors

Dunja Mladenić, Marko Grobelnik

http://is.ijs.si

6. oktober 2025 / 6 October 2025
Ljubljana, Slovenia

PREDGOVOR

Tehnologije, ki se ukvarjajo s podatki so močno napredovale. Iz prve faze, kjer je šlo predvsem za shranjevanje podatkov in kako do njih učinkovito dostopati, se je razvila industrija za izdelavo orodij za delo s podatkovnimi bazami in velikimi količinami podatkov, prišlo je do standardizacije procesov, povpraševalnih jezikov. Ko shranjevanje podatkov ni bil več poseben problem, se je pojavila potreba po bolj urejenih podatkovnih bazah, ki bi služile ne le transakcijskem procesiranju ampak tudi analitskim vpogledom v podatke. Pri avtomatski analizi podatkov sistem sam pove, kaj bi utegnilo biti zanimivo za uporabnika – to prinašajo tehnike odkrivanja znanja v podatkih (knowledge discovery and data mining), ki iz obstoječih podatkov skušajo pridobiti novo znanje in tako uporabniku nudijo novo razumevanje dogajanj zajetih v podatkih. Slovenska KDD konferenca SiKDD, pokriva vsebine, ki se ukvarjajo z analizo podatkov in odkrivanjem znanja v podatkih: pristope, orodja, probleme in rešitve.

Dunja Mladenić in Marko Grobelnik

FOREWORD

Data driven technologies have significantly progressed. The first phases were mainly focused on storing and efficiently accessing the data, resulted in the development of industry tools for managing large databases, related standards, supporting querying languages, etc. After the initial period, when the data storage was not a primary problem anymore, the development progressed towards analytical functionalities on how to extract added value from the data; i.e., databases started supporting not only transactions but also analytical processing of the data. In automatic data analysis, the system itself tells what might be interesting for the user - this is brought about by knowledge discovery and data mining techniques, which try to obtain new knowledge from existing data and thus provide the user with a new understanding of the events covered in the data. The Slovenian KDD conference SiKDD covers topics dealing with data analysis and discovering knowledge in data: approaches, tools, problems and solutions.

Dunja Mladenić and Marko Grobelnik

PROGRAMSKI ODBOR / PROGRAMME COMMITTEE

Janez Brank, Jožef Stefan Institute, Ljubljana

Jasminka Dobša, Faculty of Organization and Informatics, University of Zagreb

Alenka Guček, Jožef Stefan Institute, Ljubljana

Branko Kavšek, University of Primorska, Koper

Klemen Kenda, Qlector, Ljubljana

Bojana Mikelenić, Faculty of Humanities and Social Sciences, University of Zagreb

Elham Motamedi Mohammadabadi, Jožef Stefan Institute, Ljubljana

Irena Nančovska Šerbec, Faculty of Education, University of Ljubljana

Erik Novak, Jožef Stefan Institute, Ljubljana

Inna Novalija, Jožef Stefan Institute, Ljubljana

Joao Pita Costa, Quintelligence, Ljubljana

Jože Rožanec, Jožef Stefan Institute, Ljubljana

Abdul Sitar, Jožef Stefan Institute, Ljubljana

Luka Stopar, SolvesAll, Ljubljana

Blaž Škrlj, Teads, Ljubljana

Jan Šturm, Jožef Stefan Institute, Ljubljana

Oleksandra Topal, Jožef Stefan Institute, Ljubljana

Semantic Prompting for Large Language Models in Biomedical Named Entity Recognition

Erik Calcina
Jožef Stefan Institute
Jožef Stefan International Postgraduate School
Jamova cesta 39
Ljubljana, Slovenia

Erik Novak
Jožef Stefan Institute
Jožef Stefan International Postgraduate School
Jamova cesta 39
Ljubljana, Slovenia

Dunja Mladenić
Jožef Stefan Institute
Jožef Stefan International Postgraduate School
Jamova cesta 39
Ljubljana, Slovenia

Abstract

Extracting structured medical information from unstructured clinical text remains a challenge for biomedical research and decision support. Recent advances in large language models (LLMs) suggest that prompt-based methods could provide a promising alternative to traditional supervised approaches for Named Entity Recognition (NER) in the biomedical domain. This study investigates whether adding semantic descriptions of entity labels can improve NER performance on clinical texts. Using a dataset of annotated case reports, we evaluate model performance in zero-shot, few-shot, and fine-tuned settings. Results show that semantic prompts enhance accuracy in low-supervision scenarios, while offering limited benefit once models are fine-tuned.

Keywords

Named entity recognition, large language models, semantic prompting, prompt engineering, medical domain, biomedicine

1 Introduction

Biomedical texts present a critical challenge for automated analysis. Clinical case reports, patient records, and related narratives are written in free text rather than in structured formats. While they contain essential medical knowledge, their unstructured nature makes it necessary to extract and organize information for systematic use in research and clinical decision support. Doing this manually is costly, time-consuming, and challenging to scale. Therefore, an automated approach to extract relevant information is required.

Named entity recognition (NER) models enable the identification and classification of clinically relevant entities, such as biological structures, diagnostic procedures, or symptoms. Recent advances in large language models (LLMs) show strong generalizing abilities, identifying relevant entities in both zero-shot and few-shot settings. However, in the biomedical domain, performance can be hindered by specialized terminology and subtle entity distinctions. To address this, we propose enriching prompts with semantic descriptions of entity labels, providing models with explicit context to improve their understanding of the task.

This study investigates the impact of semantically enhanced prompting in biomedical named entity recognition using large language models. We evaluate the effect of enriching entity labels with semantic descriptions on model performance across zero-shot, few-shot, and fine-tuned scenarios, using the MACCROBAT2020 dataset [3]. The contributions of this paper are threefold. First, we introduce the use of semantically enhanced prompts for biomedical NER by enriching entity labels with descriptions. Second, we provide a systematic evaluation of semantic prompting across zero-shot, few-shot, and fine-tuned scenarios, assessing its effectiveness under different levels of supervision. Third, we apply a statistical validation method, McNemar's test, to rigorously assess the reliability of observed performance differences between baseline and semantically enhanced prompts.

The remainder of the paper is structured as follows: Section 2 contains the overview of the related work. Next, we present the methodology in Section 3, and describe the experiment setting in Section 4. The experiment results are found in Section 5, followed by a discussion in Section 6. Finally, we conclude the paper and provide ideas for future work in Section 7.

2 Related Work

This section focuses on the related work on named entity recognition in biomedicine, as well as the use of semantic descriptions in prompting.

2.1 Prompting with semantic context

PromptNER introduced the idea of augmenting few-shot prompts with entity definitions, leading to substantial gains in F1 score on benchmarks like CoNLL, GENIA, and FewNERD, improving performance by 4–9 points compared to standard prompting [2]. Extending this idea, PromptNER unifies locating and typing into a single enriched prompt, enabling phrase extraction and entity classification simultaneously [7]. Similarly, the biomedical NER study demonstrated that "on-the-fly" inclusion of concept definitions enhances performance (+15% F1) in low-data settings [5].

2.2 Iterative and zero-shot semantic prompting

Recent work in zero-shot NER explores iterative prompt refinement to align model outputs with precise entity definitions. EvoPrompt uses an evolving definition-based framework to better distinguish between similar entity types, yielding improvements across benchmarks [9]. In a broader context, some studies found that while directly injecting semantic parses into LLM inputs can degrade performance, carefully designed semantic "hints" embedded in prompts can reliably boost outcomes [1].

2.3 Domain-specific prompt optimization

FsPONER optimizes few-shot prompts for industrial NER tasks by using semantic entity–enhanced meta prompts and task-specific exemplar selection, yielding F1 improvements of 5 to 13 points

Information Society 2025, 6–10 October 2025, Ljubljana, Slovenia

https://doi.org/10.70314/is.2025.sikdd.3

in domain benchmarks [8]. In the biomedical domain, MPE³ integrates ontology-derived label semantics into prompts, improving performance in few-shot NER scenarios [10].

Prior research has shown that enriching prompts with semantic context and label definitions can significantly boost LLM performance in both few-shot and zero-shot NER. Our work provides a systematic evaluation in the biomedical domain. By examining multiple supervision settings, benchmarking several model families, and validating differences through McNemar's test, we offer a comprehensive assessment of when semantically enriched prompts provide benefits.

3 Methodology

This study evaluates the impact of incorporating semantic information into prompts on the performance of LLMs in biomedical NER tasks. Three distinct approaches were employed: zero-shot prompting, few-shot prompting, and fine-tuning.

Zero-shot prompting. In the zero-shot setting, models were prompted to perform NER without any prior exposure to labeled examples. Two types of prompts were utilized: *baseline prompt*, a standard instruction to identify and classify entities without additional context, and *semantically enhanced prompt*, which includes detailed descriptions for each entity label, offering explicit semantic context to guide the model's understanding and classification.

Few-shot prompting. The few-shot approach involved providing the models with a limited number of annotated examples (k-shots) before performing NER on new texts. Similar to the zero-shot setting, both baseline and semantically enhanced prompts were employed to assess the influence of semantic information.

Fine-tuning. Fine-tuning was conducted to adapt the pre-trained LLMs to the specific biomedical NER task. Two fine-tuning strategies were explored: *standard fine-tuning*, where models are fine-tuned using the original dataset annotations without additional semantic information, and *semantically enhanced fine-tuning*, which fine-tunes models on data where annotations were supplemented with semantic descriptions of each entity label.

4 Experiment Setting

This section describes the experiment setting, which includes the dataset and prompt preparation, the fine-tuning procedure used, the evaluation metrics, and the statistical significance test description.

4.1 Dataset

The experiments were conducted using the MACCROBAT2020 [3] dataset, which comprises 200 clinical case reports sourced from PubMed Central. In total, it contains 4,542 sentences with an average of 22.7 sentences per document, which includes manual annotations of biomedical entities, events, and relations, provided in brat standoff format[1]. For this study, we focused on the five most frequent entity labels within the dataset. These are BIOLOGICAL STRUCTURE, DIAGNOSTIC PROCEDURE, LAB VALUE, SIGN SYMPTOM, and DETAILED DESCRIPTION supplemented by the AGE and SEX labels. The inclusion of AGE and SEX was motivated by their prevalence and clarity within clinical narratives, providing a basis for evaluating model performance on both complex and straightforward entity types.

Each document was segmented into individual sentences by splitting on full stops. Subsequently, each sentence, along with its associated entity annotations, was transformed into a JSON format to facilitate processing by the language models.

[1] https://brat.nlplab.org/standoff.html

4.2 Semantically enhanced prompts

To enhance the semantic understanding of entity labels, detailed descriptions were crafted for each. These descriptions were derived by combining information from the MACCROBAT2020 dataset documentation and definitions from the Oxford English Dictionary [6]. The integration of these sources was performed manually, ensuring that the descriptions were both accurate and contextually relevant.

Prompts were structured as plain text instructions, guiding the model to identify and classify entities within the provided sentences. For the semantically enhanced prompts, the detailed entity descriptions were included to provide additional context. Models were instructed to output their responses in a JSON format, explicitly focusing on the labels component. Below we present an example of the entity description, specifically for the label AGE.

Baseline prompt: `The age of the patient.`
Semantic enhanced prompt: `The duration of time a patient has lived, expressed numerically (e.g., '65-year-old', '20 years old') or categorically (e.g., 'newborn', 'teenage'), representing their age at the time of presentation.`

This added context is intended to improve the model's ability to distinguish and extract nuanced biomedical entities more accurately.

4.3 Fine-tuning procedure

Fine-tuning is carried out using parameter-efficient techniques, where only lightweight adapter modules are trained instead of modifying the full model. This strategy reduces memory usage, mitigates catastrophic forgetting, and accelerates training.

To further improve efficiency, models are quantized to 4-bit precision. Fine-tuning is supervised and focuses on the generated outputs; all non-target tokens (e.g., system prompts, input context) are masked during loss computation. This ensures that training adapts the model to the expected JSON label output format rather than to the input content or prompt structure.

4.4 Evaluation metrics

To evaluate entity recognition performance, we use two F1-based metrics. The Exact F1 score measures strict matches, requiring predicted entities to align perfectly with the reference text and label. The Relaxed F1 score allows partial matches, counting predictions as correct if they include the true entity as a substring with the correct label.

4.5 McNemar statistical significance test

While Exact and Relaxed F1 scores quantify the magnitude of performance differences, they do not establish whether these differences are statistically reliable. The McNemar test [4] complements the Exact F1 metric by verifying whether observed improvements can be attributed to the semantically enhanced

prompts rather than random variation. Following standard NER practice, we treat Exact F1 as the primary endpoint and therefore apply McNemar's test only to exact match predictions.

Let b denote the number of cases correctly predicted by the semantically enhanced model but missed by the baseline, and c the number of cases correctly predicted by the baseline but missed by the semantically enhanced model. Only discordant pairs (b, c) contribute to the test; agreements do not affect the statistic. Using the continuity-corrected version of the test, the statistic is computed as

$$\chi^2 = \frac{(|b - c| - 1)^2}{b + c},$$

which follows a chi-squared distribution with one degree of freedom. The corresponding p-value allows us to test the null hypothesis H_0: the two models have equal marginal probabilities (i.e., performance differences are due to chance). Conventionally, $p < 0.01$ is considered statistically significant.

5 Results

This section presents model performance under three experimental conditions: zero-shot, few-shot, and fine-tuned prompting. For each condition, we compare the impact of semantically enhanced prompts against standard prompts using Exact and Relaxed F1 scores on a subset of clinically relevant entity types.

5.1 Zero-shot prompting

Table 1 reports the Exact and Relaxed F1 scores for models evaluated in the zero-shot setting using semantically enhanced prompts. Without semantic descriptions, most models struggled to generate outputs in the required JSON format, and valid scores could not be computed. Even with semantically enhanced prompts, Meta-Llama-3.1-8B consistently failed to produce structured responses.

Among the evaluated models, Llama-3.1-8B-Instruct achieved the highest Exact F1 score, while txgemma-9b-chat attained the best Relaxed F1 score. Llama-3.2-3B-Instruct and DeepSeek-Qwen-7B also demonstrated non-trivial performance in both metrics. These results suggest that semantically enhanced prompts can effectively compensate for the absence of training examples in zero-shot scenarios by providing clearer task guidance and improving structured prediction output.

Table 1: Exact and Relaxed F1 scores in the zero-shot setting with semantically enhanced prompts. Bolded values indicate the highest score in each column. Results without valid JSON output are marked with /.

Model	Exact F1 Semantics	Relaxed F1 Semantics
Llama-3.1-8B-Instruct[2]	**0.2310**	0.3708
Meta-Llama-3.1-8B[3]	/	/
Llama-3.2-3B-Instruct[4]	0.1620	0.3254
DeepSeek-Qwen-7B[5]	0.1592	0.3217
txgemma-9b-chat[6]	0.2181	**0.4245**

[2]https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct
[3]https://huggingface.co/meta-llama/Llama-3.1-8B
[4]https://huggingface.co/meta-llama/Llama-3.2-3B-Instruct
[5]https://huggingface.co/deepseek-ai/DeepSeek-R1-Distill-Qwen-7B
[6]https://huggingface.co/google/txgemma-9b-chat

5.2 Few-shot prompting

Table 2 summarizes the Exact and Relaxed F1 scores for few-shot prompting. The addition of semantic information consistently improved model performance across most models. Notably, txgemma-9b-chat achieved the highest Exact F1 score 0.3288 and Relaxed F1 score 0.4998 with semantic prompting, compared to 0.2732 and 0.4469 without.

Both Llama-3.1-8B-Instruct and Llama-3.2-3B-Instruct showed improvements in both Exact and Relaxed F1 scores when provided with semantically enhanced prompts. For instance, Llama-3.1-8B-Instruct improved from 0.2509 to 0.3005 (Exact) and from 0.3526 to 0.3948 (Relaxed), while Llama-3.2-3B-Instruct increased from 0.2300 to 0.2439 (Exact) and from 0.3769 to 0.3948 (Relaxed). These gains highlight the benefit of enriching prompt instructions when training data is limited. However, not all models responded positively. For example, Meta-Llama-3.1-8B experienced a drop in Exact F1 from 0.2698 to 0.2210 and in Relaxed F1 from 0.3537 to 0.2799, indicating that semantically enhanced prompts do not universally improve performance and may be less effective for some models.

To assess the reliability of these differences, we conducted McNemar tests on Exact paired predictions. The tests revealed that performance differences between baseline and semantically enhanced prompts were statistically significant for all models except Llama-3.2-3B-Instruct. It is important to note, however, that significance here indicates that the two variants produce systematically different predictions, but does not itself imply improvement. For instance, while the difference for Meta-Llama-3.1-8B was highly significant, the semantically enhanced model in fact performed worse in terms of F1 scores.

5.3 Fine-tuned performance

In the fine-tuning scenario, results were more nuanced. As shown in Tables 2, most models performed strongly even without semantic enhancements. For instance, Meta-Llama-3.1-8B attained the highest Exact F1 score (0.7099) with semantic input, only slightly outperforming its baseline (0.7076), and this difference was not statistically significant ($p \approx 0.64$).

Some models, such as Llama-3.1-8B-Instruct and Llama-3.2-3B-Instruct, even showed small performance drops when semantic descriptions were included, with McNemar tests confirming that these differences were not significant ($p \approx 0.75$ and $p \approx 0.88$). This suggests that in settings where the model is already exposed to sufficient task specific supervision, additional prompt-level context may offer limited benefit or even introduce redundancy.

In contrast, TxGemma-9B-Chat exhibited the most notable improvement, with Exact and Relaxed F1 scores increasing from 0.6837 to 0.7092 and from 0.7483 to 0.7686, respectively; the McNemar test confirmed this difference as statistically significant ($p \approx 9.7 \times 10^{-5}$). By comparison, DeepSeek-Qwen-7B also showed a significant difference ($p \approx 6 \times 10^{-3}$), but in this case the semantically enhanced model performed worse (Exact F1: 0.7013 → 0.6879).

5.4 Overall observations

The largest performance improvements from semantically enhanced prompts appeared in zero-shot and few-shot settings, where gains in F1 scores were often statistically significant. In contrast, fine-tuned models showed smaller and mixed effects:

Table 2: Exact (left) and Relaxed (right) F1 scores for selected labels in few-shot and fine-tuned settings, with and without semantically enhanced prompts. Bolded values indicate the highest score in each column. We use symbols ○ and ● to denote whether the differences between using the baseline or semantically enhanced prompts are statistically significant (●) or not (○) according to the McNemar test at a significance level of $p = 0.01$.

Model	Exact F1				Relaxed F1			
	Few-Shot		Fine-Tuned		Few-Shot		Fine-Tuned	
	/	Semantic	/	Semantic	/	Semantic	/	Semantic
Llama-3.1-8B-Instruct	0.2509	0.3005 ●	0.7053	0.7004 ○	0.3526	0.3948	0.7660	0.7645
Meta-Llama-3.1-8B	0.2698	0.2210 ●	**0.7076**	**0.7099** ○	0.3537	0.2799	**0.7670**	**0.7765**
Llama-3.2-3B-Instruct	0.2300	0.2439 ○	0.6881	0.6867 ○	0.3769	0.3948	0.7629	0.7622
DeepSeek-Qwen-7B	0.1423	0.2270 ●	0.7013	0.6879 ●	0.2465	0.3891	0.7584	0.7521
txgemma-9b-chat	**0.2732**	**0.3288** ●	0.6837	0.7092 ●	**0.4469**	**0.4998**	0.7483	0.7686

for most, differences were not significant, though TxGemma-9B-Chat benefited reliably while DeepSeek-Qwen-7B showed a significant decrease. These results indicate that semantic prompting is most effective in low-resource conditions, while its impact under full supervision is limited and model-dependent.

6 Discussion

This section discusses the experiment findings and highlights the advantages and disadvantages of the different approaches.

6.1 Model pretraining and domain adaptation

TxGemma-9B-Chat, based on the Gemma 2 architecture and further fine-tuned on therapeutic development data, outperformed general-purpose models in a few-shot scenario. This suggests that domain-specific pretraining can significantly improve performance when supervision is limited. However, in the full fine-tuning setting, its advantage diminished. In fact, general models like Meta-Llama-3.1-8B achieved comparable but slightly better results, indicating that once sufficient task-specific supervision is provided, prior domain specialization offers limited additional benefit.

6.2 Prompt quality matters

The structure and clarity of prompts are critical to model performance. Poorly designed prompts often resulted in JSON formatting errors or reduced accuracy, particularly in zero-shot and few-shot settings. While adding semantic context improves task understanding by making objectives and entity definitions more explicit, excessive length or ambiguity can offset these gains.

6.3 Prompt length vs. model response

Semantic enrichment inevitably increases prompt length, which can slow response time and raise computational overhead. It may also overwhelm smaller models when excessive detail is included. In practical applications, this must be weighed against the potential gains in entity extraction accuracy.

7 Conclusion

This study investigated the impact of a semantically enhanced prompt design on LLM-based NER in the clinical domain. Our experiments on the MACCROBAT2020 dataset demonstrated that adding semantic label descriptions significantly improves model performance in zero-shot and few-shot scenarios, with notable gains in both Exact and Relaxed F1 scores. In contrast, fine-tuned models already exposed to task-specific data showed only marginal improvement.

Future work could explore adaptive semantic prompting strategies, such as ontology-driven label enrichment, and further investigate the trade-offs between prompt length and inference efficiency. Additionally, this method could be tested on larger datasets and across different models to assess its generalizability.

In summary, semantically enhanced prompts offer a straightforward yet effective way to boost clinical NER performance in low-data regimes, but their impact diminishes as models are exposed to more supervised training.

Acknowledgements

This work was supported by the Slovenian Research Agency. Funded by the European Union. UK participants in Horizon Europe Project PREPARE are supported by UKRI grant number 10086219 (Trilateral Research). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Health and Digital Executive Agency (HADEA) or UKRI. Neither the European Union nor the granting authority nor UKRI can be held responsible for them. Grant Agreement 101080288 PREPARE HORIZON-HLTH-2022-TOOL-12-01.

References

[1] Kaikai An, Shuzheng Si, Yuchi Wang, et al. 2024. Rethinking semantic parsing for large language models. *arXiv preprint arXiv:2409.14469*.
[2] Dhananjay Ashok and Zachary C. Lipton. 2023. Promptner: prompting for named entity recognition. *arXiv preprint arXiv:2305.15444*.
[3] J. Harry Caufield, Yichao Zhou, Yunsheng Bai, David A. Liem, Anders O. Garlid, Kai-Wei Chang, Yizhou Sun, Peipei Ping, and Wei Wang. 2019. A comprehensive typing system for information extraction from clinical narratives. *medRxiv*. Preprint. DOI: 10.1101/19009118.
[4] Quinn McNemar. 1947. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12, 2, (June 1947), 153–157. DOI: 10.1007/bf02295996.
[5] Monica Munnangi, Sergey Feldman, Byron C. Wallace, et al. 2024. On-the-fly definition augmentation of llms for biomedical ner. *arXiv preprint arXiv:2404.00152*.
[6] 2025. Oxford english dictionary. https://www.oed.com/. Accessed: 2025-06-17. (2025).
[7] Yongliang Shen, Zeqi Tan, Shuhui Wu, et al. 2023. Promptner: prompt locating and typing for named entity recognition. In *ACL (Long Papers)*.
[8] Yongjian Tang, Rakebul Hasan, and Thomas Runkler. 2024. Fsponer: few-shot prompt optimization for named entity recognition. *arXiv preprint arXiv:2407.08035*.
[9] Zeliang Tong, Zhuojun Ding, and Wei Wei. 2025. Evoprompt: evolving prompts for enhanced zero-shot named entity recognition. In *COLING*.
[10] Yuwei Xia, Zhao Tong, Liang Wang, et al. 2023. Learning meta-prompt with entity-enhanced semantics for few-shot ner. *SSRN*.

LLM Based Approach to Extracting Smells in Slovenian Corpora

Janez Brank
Jožef Stefan Institute
Ljubljana, Slovenia
janez.brank@ijs.si

Inna Novalija
Jožef Stefan Institute
Ljubljana, Slovenia
inna.koval@ijs.si

Dunja Mladenić
Jožef Stefan Institute
Ljubljana, Slovenia
dunja.mladenic@ijs.si

Marko Grobelnik
Jožef Stefan Institute
Ljubljana, Slovenia
marko.grobelnik@ijs.si

Abstract

This paper presents a comparative study of automatic smell detection in Slovenian cultural heritage texts using both keyword-based search and large language model (LLM) inference. We process a portion of the dLib.si corpus from the late 19th and early 20th centuries, analyzing over 1.6 million text segments for olfactory references. The keyword method leverages an expert-curated list of smell terms, while the LLM method applies semantic inference via prompt-engineered queries. We compare the methods in terms of detection density, temporal trends, and agreement overlap. Additionally, we visualize the semantic landscape of extracted smell terms using t-SNE and unsupervised clustering with auto-generated labels. Our findings reveal limited overlap between methods, a shared rise in smell mentions over time, and distinct semantic clusters ranging from industrial to culinary and bodily smells. This study highlights the value of combining symbolic and neural approaches for nuanced sensory mining in digital heritage corpora.

Keywords

LLM, Artificial Intelligence, Cultural Heritage, Text Mining

1 Introduction

Olfactory perception is an essential yet underexplored dimension in the analysis of historical texts, particularly within the cultural heritage domain. Smells, though intangible, play a critical role in shaping memory, atmosphere, and cultural meaning. However, their representation in written sources is often subtle, indirect, or metaphorical. This challenge becomes more pronounced in historical corpora such as 19th- and early 20th-century Slovenian publications, where evolving linguistic practices and cultural norms affect how sensory information is encoded.

This paper explores automatic smell detection in Slovenian cultural heritage texts using two complementary strategies: (1) a keyword-based approach derived from an expert-curated list of smell-related expressions and their morphological variants, and (2) large language model (LLM) - based semantic inference using prompt-engineered queries via the Together.ai platform. We process a subset of the dLib.si digital library corpus of Slovenian texts, divided into temporal buckets, and evaluate the performance, overlap, and divergence between the two methods.

To facilitate large-scale analysis, we produce and analyze over 1.6 million document-query pairs, extracting smell mentions, classifying them by agreement type, and visualizing their distributions both temporally and semantically. Our goals are twofold: (i) to quantify the representational density of olfactory references in the corpus, and (ii) to better understand how computational methods can surface subtle cultural patterns that evade traditional keyword search alone.

This work contributes toward a richer modeling of sensory information in digital heritage collections and highlights the value of combining symbolic and neural methods for text mining in the cultural heritage domain.

2 Related Work

Recent years have seen increased interest in the computational modeling of olfactory expressions in historical and cultural texts. A prominent initiative in this space is the Odeuropa project [7], which focused on identifying, curating, and semantically linking smell-related content in European heritage corpora. Large-scale initiatives, such as the Odeuropa project, have produced the European Olfactory Knowledge Graph and tools like the Smell Explorer to trace historical olfactory knowledge across 400 years of European sources [7, 5]. Research on sensory perception in NLP has traditionally focused on the visual and auditory modalities, while olfaction remains relatively underexplored. Annotation frameworks such as the Olfactory Event Frame and guidelines for labeling sources, qualities, and experiences [6] provide structured resources for information extraction from historical and literary corpora. Traditional approaches to olfactory semantics rely on fixed lexicons such as the Dravnieks Atlas [1] and the DREAM challenge descriptors [3]. For morphologically complex and

Information Society 2025, Ljubljana, Slovenia

https://doi.org/10.70314/is.2025.sikdd.5

low-resource languages such as Slovene, monolingual models like SloBERTa [10] and seq-to-seq models like SloT5 [9] demonstrate that tailoring architectures to linguistic structure improves performance over multilingual baselines. A wide range of Slovene corpora underpins these modeling efforts. Gigafida 2.0, a reference corpus of 1.1 billion tokens covering contemporary written Slovene, provides a large-scale foundation for model pretraining and evaluation [4]. For user-generated content, the JANES corpus supplies richly annotated Slovene social media text, including normalization and NER [2]. Unlike prior studies that primarily focus on annotation frameworks, fixed olfactory lexicons, or large-scale multilingual heritage initiatives such as Odeuropa, our work provides the first comparative evaluation of keyword-based and LLM-based smell detection specifically for Slovenian cultural heritage corpora, highlighting the interplay between symbolic coverage and neural semantic inference.

3 Corpora and Preprocessing

For the experiments presented in this paper, we used texts from the Slovenian Digital Library (dLib.si). Initially we downloaded, from the Library's website, all documents from the period 1870–1919 for which OCRed text was available and whose language was marked as Slovene in the metadata there. In terms of content, this covers nearly all books, newspapers, magazines etc. published in Slovene during that period. From this corpus we then randomly selected 7 % of the documents from each year for further processing; thus the selected subset maintains the same distribution over time, genre, etc. as the full corpus. This resulted in a dataset of approx. 366 thousand documents with a total of 105 million words.

4 Methodology

This section outlines the analytical pipeline used to detect, compare, and interpret smell-related expressions in Slovenian cultural heritage texts. Our approach combines large language model inference, keyword-based retrieval, temporal and density statistics, and unsupervised semantic clustering.

4.1 Comparative Evaluation of Detection Methods

In order to identify olfactory expressions, we employed two complementary strategies:

- **LLM-based Extraction:** Each document was split into passages and processed using a LLM.[1] The model returned a list of potential smell-related words or phrases, structured in JSON format. In cases of formatting failure, raw strings or exception messages were recorded.
- **Keyword-Based Search:** A manually curated index of smell-related expressions, including morphologically inflected forms, was used for direct string matching within each passage.[2]

[1]The Llama-3.3-70B-Instruct-Turbo-Free model, accessed via Together.ai.
[2]This index has been kindly provided by Mojca Ramšak and is based on her work on the anthropology of smell [8].

For each passage, we recorded both LLM and keyword results. We classified outcomes into four categories: *LLM Only*, *Keyword Only*, *Both*, or *None*. Additionally, we computed the **Jaccard similarity** J between the two result sets:

$$J(A, B) = |A \cap B| \,/\, |A \cup B|,$$

where A is the set of LLM-based results and B is the set of keyword-based results. This metric enabled quantitative comparison of coverage and intersection across detection methods.

4.2 Temporal Distribution of Smell Mentions

We extracted the year of publication from each document's metadata. For each year, we aggregated:

- Total LLM-detected smell terms
- Total keyword-detected smell terms
- Number of processed queries

These aggregates were used to generate yearly time series, revealing longitudinal patterns in olfactory expression across the corpus. This temporal analysis supports hypotheses about cultural shifts, such as increasing industrial or bodily smell discourse over time.

4.3 Semantic Typology via Clustering of Smell Terms

To explore latent smell categories, we constructed a semantic typology using the following steps:

- **Term Extraction:** We extracted the 500 most frequent smell-related terms from the combined LLM and keyword results.
- **Vectorization:** Terms were embedded using TF-IDF vectors over character-level n-grams (`char_wb` with range 2–4), capturing morphological similarity.
- **Dimensionality Reduction:** The high-dimensional vectors were projected to two dimensions using **t-SNE** (perplexity = 30), yielding a visual semantic landscape.
- **Clustering:** We applied ***k*-means clustering** (with $k = 8$) to the t-SNE coordinates. For each cluster, the top 5 TF-IDF terms were used to generate semantic labels (e.g., *"Herbs & Cooking"*, *"Pharmaceutical Smells"*).
- **Visualization:** The clusters were visualized with color-coded labels and representative terms. Interactive versions were built using `plotly`.

This typology enables data-driven classification of smell discourse and provides interpretable categories for cultural and linguistic analysis.

4.4 Document-Level Smell Density Analysis

To assess the distribution of olfactory content across documents, we computed the *smell density* as the ratio of detected terms to queries per document:

$$\text{Density}_{\text{LLM}} = \frac{\#\text{ LLM terms}}{\#\text{ queries}}$$

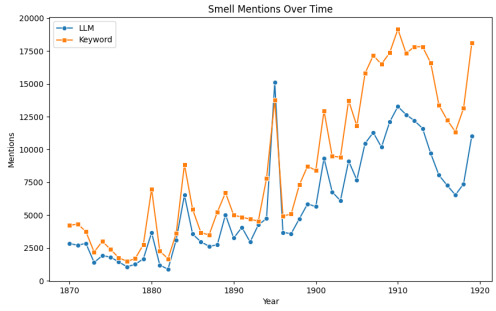


Figure 1: Yearly trends in smell term mentions. Keyword-based detection consistently returns higher frequencies than the LLM, but both show similar growth patterns.

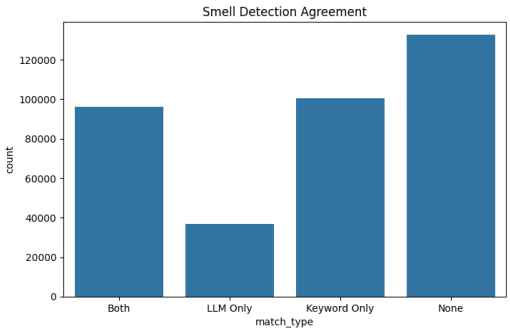


Figure 2: Detection agreement between LLM and keyword methods. Most passages are matched by one method only, with a significant number showing no detection. The overlap ("Both") occurs in fewer than one-third of cases.

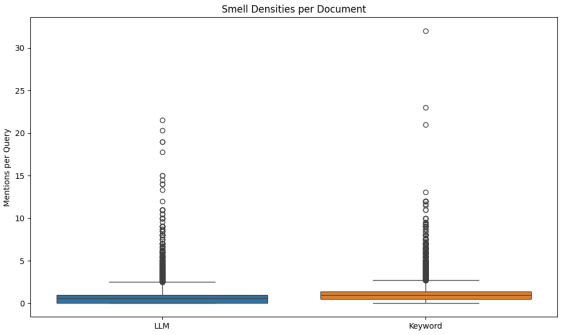


Figure 3: Smell term density per document. While outliers exist for both methods, keyword-based detection generally identifies a higher density of smell references per query.

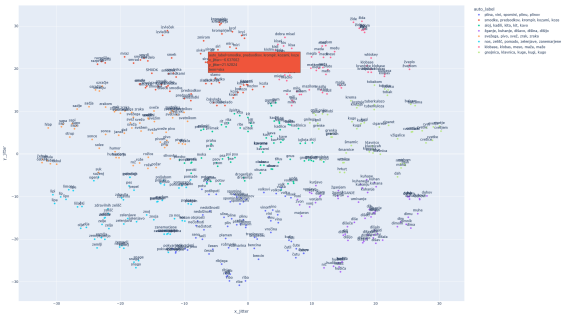


Figure 4: t-SNE semantic landscape of smell terms, clustered by character-level similarity and automatically labeled using top TF-IDF terms per group. The visualization reveals coherent groups such as food, ritual, body, and chemical references.

$$\text{Density}_{\text{Keyword}} = \frac{\#\text{ Keyword terms}}{\#\text{ queries}}$$

This metric enabled identification of smell-rich and smell-sparse texts. Density distributions were visualized using boxplots and descriptive statistics, facilitating selection of representative or outlier texts for deeper qualitative analysis.

5 Evaluation and Results

We evaluated complementary approaches to detecting olfactory references in historical corpora: a keyword-based method and an LLM-based classifier. The results highlight both convergences and divergences in performance across time, document density, and semantic coverage.

Figure 1 shows yearly frequencies of smell-related mentions from 1870 to 1920. While keyword-based detection consistently yields higher absolute counts than the LLM, both methods exhibit similar growth trajectories.

Agreement analysis between the two methods (Figure 2) reveals substantial divergence. Only about one-third of passages are identified by both approaches. A large portion is captured exclusively by the keyword method, while the LLM contributes a smaller but meaningful number of unique detections. A significant subset of passages registers no olfactory detection at all, probably because most documents don't mention smell-related topics in the first place.

Figure 3 illustrates the distribution of smell term density per document. Keyword-based detection generally produces higher densities of references, whereas the LLM outputs are sparser but potentially more semantically filtered. Both distributions exhibit long-tailed outliers, where certain documents contain disproportionately high concentrations of olfactory mentions.

To further analyze lexical diversity, we applied t-SNE to embed and cluster smell-related terms (Figure 4). The resulting semantic landscape reveals coherent groupings that align with cultural domains, including food, ritual, body, and chemical references. These clusters highlight the variety of olfactory expressions and suggest that both methods capture complementary facets of the semantic space. The LLM appears particularly adept at recognizing context-dependent terms, while the keyword method anchors clusters in explicit lexical cues.

Overall, the keyword-based approach provides broader coverage and higher frequencies, but at the cost of noise and overcounting. The LLM method, while more conservative, contributes precision and captures context-sensitive

olfactory references that keywords may overlook. The combination of both thus provides a richer and more balanced representation of olfactory discourse in historical texts.

6 Discussion

Our analysis reveals several key insights into olfactory representations in Slovenian cultural heritage texts and the methodological implications of combining LLM-based and keyword-based detection.

First, both detection strategies show meaningful trends over time, with a noticeable increase in smell-related references around the turn of the 20th century. This may reflect broader urbanization, industrialization, and shifts in public health discourse, which intensified the cultural significance of air quality, hygiene, and olfactory environments.

Second, although keyword-based detection consistently returned more hits, the LLM-based method surfaced a distinct set of semantically inferred mentions. As the agreement analysis shows, only a minority of mentions (∼24 %) were matched by both methods. One possible explanation of this would be if neural inference captures more nuanced or contextually implied smell references, such as metaphorical use ("a whiff of suspicion") or implied odors in narrative scenes.

Third, density analysis suggests that LLMs return more sparse but targeted mentions, while keyword detection produces broader but sometimes noisier coverage. This difference is critical for researchers deciding between high recall and high precision when exploring sensory data in historical texts.

Finally, the t-SNE landscape of smell terms uncovered semantically coherent clusters — e.g., medicinal substances, industrial emissions, festive foods, and bodily decay - and allowed us to generate meaningful auto-labels using top TF-IDF terms. Such visualizations provide a valuable tool for cultural historians to engage with thematic patterns across large-scale textual datasets.

Overall, our findings underscore the value of hybrid approaches to cultural text analysis. By comparing symbolic and neural perspectives, we gain both coverage and subtlety, enabling a deeper reconstruction of sensory worlds encoded in the archives.

7 Conclusion and Future Work

We conducted a dual-method analysis of olfactory references in Slovenian historical texts, revealing how keyword search and LLM-based inference each contribute unique perspectives to sensory data mining. Our results show that while the keyword method offers broad lexical coverage, the LLM can detect more subtle, implied, or metaphorical references often overlooked by surface-level matching.

Furthermore, t-SNE clustering of smell terms revealed rich thematic structures — such as food, medicine, pollution, and ritual — highlighting the semantic complexity of olfactory language.

Together, these results demonstrate the complementary strengths of symbolic and neural approaches for enriching digital humanities research, especially in domains like historical sensory studies where annotation is sparse and vocabulary is diffuse.

Several promising directions remain open for further exploration. First, we plan to expand the dataset to cover all documents in the dLib.si corpus, enabling more robust longitudinal and regional analyses. Second, we aim to improve LLM prompts to better handle nested or narrative contexts, including smells embedded in metaphor, irony, or emotional framing.

Another avenue involves extending the classification of smell mentions into functional categories (e.g., pleasant vs. unpleasant, natural vs. artificial, bodily vs. environmental) using additional LLM-based postprocessing. We also intend to explore multilingual smell detection, comparing Slovene with other Central European languages to study cultural convergence and divergence in olfactory discourse.

Finally, we hope to integrate our smell detection pipeline into public digital heritage platforms, providing curators, historians, and linguists with new tools for sensory exploration of archival materials.

Acknowledgements

This work was supported by the Slovenian Research Agency under the project J7-50233.

References

[1] Andrew Dravnieks. 1992. *Atlas of Odor Character Profiles*. ASTM International, (Feb. 1992). ISBN: 978-0-8031-0456-3. DOI: 10.1520/DS61-EB.

[2] Darja Fišer, Nikola Ljubešić, and Tomaž Erjavec. 2020. The janes project: language resources and tools for slovene user generated content. *Language Resources and Evaluation*, 54, 1, pp. 223–246. Retrieved Aug. 27, 2025 from https://www.jstor.org/stable/48740864.

[3] Andreas Keller et al. 2017. Predicting human olfactory perception from chemical features of odor molecules. *Science*, 355, (Feb. 2017), eaal2014. DOI: 10.1126/science.aal2014.

[4] Simon Krek, Špela Arhar Holdt, Tomaž Erjavec, Jaka Čibej, Andraz Repar, Polona Gantar, Nikola Ljubešić, Iztok Kosem, and Kaja Dobrovoljc. 2020. Gigafida 2.0: the reference corpus of written standard Slovene. eng. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*. Nicoletta Calzolari et al., editors. European Language Resources Association, Marseille, France, (May 2020), 3340–3345. ISBN: 979-10-95546-34-4. https://aclanthology.org/2020.lrec-1.409/.

[5] P. Lisena, T. Ehrhart, and R. Troncy. European olfactory knowledge graph. Zenodo. DOI: 10.5281/zenodo.10709703.

[6] Stefano Menini, Teresa Paccosi, Serra Sinem Tekiroğlu, and Sara Tonelli. 2023. Scent mining: extracting olfactory events, smell sources and qualities. In *Proceedings of the 7th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*. Stefania Degaetano-Ortlieb, Anna Kazantseva, Nils Reiter, and Stan Szpakowicz, editors. Association for Computational Linguistics, Dubrovnik, Croatia, (May 2023), 135–140. DOI: 10.18653/v1/2023.latechclfl-1.15.

[7] ODEUROPA Project Consortium. 2021–2023. ODEUROPA: negotiating olfactory and sensory experiences in cultural heritage practice and research. https://odeuropa.eu/. EU Horizon 2020 research and innovation programme, grant agreement No. 101004469. Royal Netherlands Academy of Arts and Sciences (KNAW) Humanities Cluster et al., (2021–2023).

[8] Mojca Ramšak. 2025. *Antropologija vonja*. AMEU-ISH, Ljubljana.

[9] Matej Ulčar and Marko Robnik-Šikonja. 2023. Sequence-to-sequence pretraining for a less-resourced slovenian language. *Frontiers in Artificial Intelligence*, 6.

[10] Matej Ulčar and Marko Robnik-Šikonja. 2021. Sloberta: slovene monolingual large pretrained masked language model. In *SiKDD*.

BetweenTheLines - Cross Source News Analysis

Georgi Trajkov
geotrajkov0@gmail.com
Jožef Stefan Institute
Ljubljana, Slovenia

Marko Grobelnik
marko.grobelnik@ijs.si
Jožef Stefan Institute
Ljubljana, Slovenia

Adrian Mladenic Grobelnik
adrian.m.grobelnik@ijs.si
Jožef Stefan Institute
Ljubljana, Slovenia

Abstract

Different news outlets covering the same event often emphasize, omit, or frame facts differently, making cross-source comparison essential for understanding media bias and information diversity. Large language models (LLMs) can automate this analysis, but simple single-LLM prompt approaches tend to underperform when processing large amounts of data [1]. Platforms like Ground News [2] and Event Registry [3] provide publisher and article-level bias scores but cannot track how individual claims and entities are portrayed by articles. The fundamental challenge is determining whether LLM prompt architecture affects accuracy when classifying claim presence across multiple news sources. We show that a multi-prompt LLM architecture reduces classification errors 7-fold (from 33.0% to 4.67%) compared to single-prompt approaches. Our pipeline first extracts all claims and entities from articles collectively, then evaluates each article separately for claim presence (confirmed/contradicted/partial/absent) and entity sentiment. This decomposition virtually eliminates false positives, major errors dropped from 28.0% to 0.79% across 797 manually validated claim-publisher pairs from Slovene news. The results demonstrate that task decomposition, not LLM sophistication, drives accuracy in cross-source analysis. This finding enables scalable media monitoring at $0.01 per event, making systematic bias detection accessible to journalists and researchers worldwide.

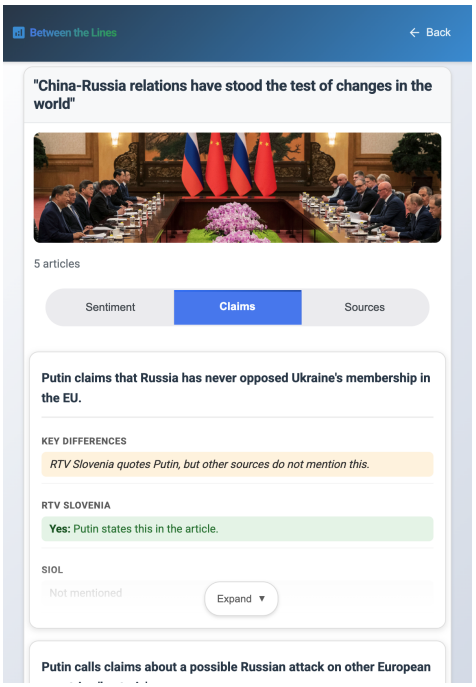


Figure 1: Analyzed event in BetweenTheLines mobile webapp, showing the claims tab

1 Introduction

Different news sources (publishers) covering the same event (groups of articles reporting on the same story) often cover facts differently. While existing platforms like Event Registry [3] and Ground News [2] provide valuable bias indicators and sentiment scores, they do not track how specific entities (People, Organizations, Countries) and claims (Factual Claims) within articles are portrayed across publishers. Getting insight into these differences is usually time-consuming for the user.

Thus we present BetweenTheLines, (Figure 1) a system that automatically identifies claims and entities in an event, and tracks their portrayal in each individual publisher. For example, when analyzing political coverage, we can see how the same entity is portrayed differently by 2 publishers, and how one publisher omitted a claim while the other did not.

Our key technical contribution is demonstrating that multi-prompt LLM architecture outperforms single-stage approaches for this task.

SiKDD 2025, Ljubljana, Slovenia

https://doi.org/10.70314/is.2025.sikdd.26

2 Related Work

Cross-source news analysis is an under-discussed area of research which is important for understanding media bias, information diversity, and narrative framing across different outlets. This section reviews existing approaches to cross-source news analysis, event aggregation systems, and LLM-based content analysis pipelines.

2.1 Cross-Source News Analysis Platforms

Ground News represents a prominent platform for cross-source news comparison, classifying publishers along the left-right political spectrum. The platform has gained widespread adoption in educational institutions, with libraries at Harford Community College [4] and West Virginia University [5] integrating it into their media literacy curricula. For each news event, Ground News allows users to compare coverage by publisher on aggregate. While these aggregated summaries can reveal different emphases across the political spectrum, the platform does not provide article-by-article comparisons or track how specific entities and claims are portrayed between articles.

2.2 Event-Centric News Aggregation

Event Registry [6, 3] pioneered event-centric news aggregation by clustering articles from multiple publishers around identified news events. The platform provides article-level sentiment scores using VADER sentiment analysis [7] and allows filtering

by various parameters including language, location, and publisher credibility. Each article has a sentiment score, a level of granularity above Ground News. Still there is no analysis for how specific entities and claims within those articles are portrayed.

Our work builds upon Event Registry's foundation, by combining its event-based aggregation, with more granular entity and claim analyses through LLM processing. Unlike Ground News's publisher-level political bias ratings or Event Registry's article sentiment scores, we provide fine-grained analysis of how specific entities and claims are portrayed differently across publishers.

3 Application and analysis Architecture

3.1 Application architecture

"BetweenTheLines is a news-analysis web app 1, developed with Claude Code [8]." The backend is built using Flask [9] and PostgreSQL [10]. It uses Event Registry [6, 3] analysis service for event and article fetching, and integrates both Google Gemini [11] and OpenAI [12] LLMs.

3.2 Analysis Service overview

The analysis service consists of two modules, claims analysis and sentiment analysis, with more thorough exploration of the former due to it's less subjective nature. Figure 2 illustrates our three-stage LLM pipeline.

Stage 1: Extraction. We begin by sending all articles from an event to a single LLM call. This extracts two lists (Table 1) for entities and claims that appear in the articles.

Stage 2: Classification. With the lists from stage 1, a parallel LLM call is made twice for each publisher, once for claims, once for entities. The calls return categorized data. Claims are categorized by presence, and entities by sentiment. The results of these categorizations are referred to as entity-publisher and claim-publisher pairs.

Stage 3: Key Differences. Summarizes how different publishers covered each claim or entity. This requires one LLM call per claim/entity, running in parallel.

The final results are structured into a tabular or card format, depending on device, where users can compare coverage across publishers at a glance (Figure 1).

3.3 Language

We decided for all prompts to be in Slovene, and to analyze only Slovene articles. This came after empirically observing a decrease in errors when the language of the prompts and articles was the same. It also language consistency for evaluation.

All showcased prompts and results are originally Slovene, and were translated to English for the paper.

3.4 Event Filtering

Events and articles are fetched from the Event Registry API[3].

Articles are then filtered to only retain the newest article for each unique publisher in an event. To retain only the most relevant events, we discard any events with less than 3 articles.

To prevent context overloading maximum article limit is 10. Then final article list is prepared for each event, and the title, body, publisher name, and article link is stored for every article.

3.5 Extraction

Extraction for an event is done after filtering, in a single LLM call to gpt-4o-mini [13], in which the contents of all articles are included in the prompt along with instructions for extracting 2 lists (Table 1) in JSON format: entities for sentiment analysis and claims for claims analysis.

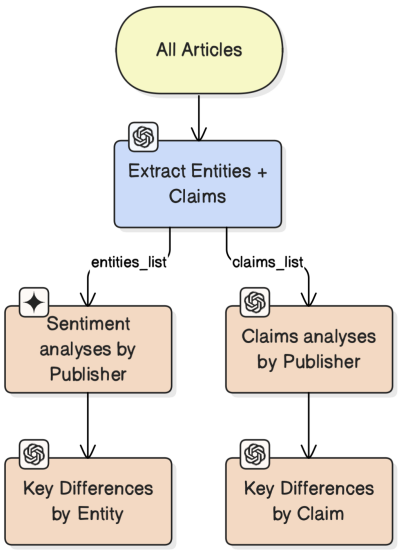


Figure 2: three Stage process flow of analysis. Extracted results lead to multiple parallel LLM calls.

The prompt focuses on extracting 8-15 claims and 8-15 entities that are central to the story, explicitly excluding news publishers unless they are the subject of the news story:

```
Analyze all these news articles and extract two comprehensive
     lists in JSON format:
  1. All significant CLAIMS made across all articles
  2. All important ENTITIES (people, organizations, countries, etc
       .) mentioned across all articles
```

A 2-step extraction process was also tested, where each article is prompted for claims and entities contained in it, and then the results are aggregated. However, this led to very large lists with duplicate names written differently (e.g., USA vs United States Government vs United States), for little performance gain.

Another issue we faced was the publisher names themselves being in the entities list, even in situations where they are not a direct part of the article. This led us to add additional rules in the extraction prompt to not include them:

```
-EXCLUDE news publishers/sources (like BBC, CNN, Reuters, etc.)
      UNLESS they are actually subjects of the news story itself
- Focus on entities that are the SUBJECT of the news, not the
     source reporting it
```

Entities	Claims
Vladimir Putin	Putin claims that Russia has never opposed Ukraine's membership in the EU.
Xi Jinping	Putin calls claims about a possible Russian attack on other European countries "hysteria."
Russia	Putin says that Russia is forced to respond to the West's attempt to take over the post-Soviet space.
China	Putin and Trump discussed the security of Ukraine.
Ukraine	Putin and Xi signed about 20 agreements in the fields of energy, aviation, artificial intelligence, and agriculture.

Table 1: Example of first 5 entities and claims received from extraction prompt for Russia–China summit.

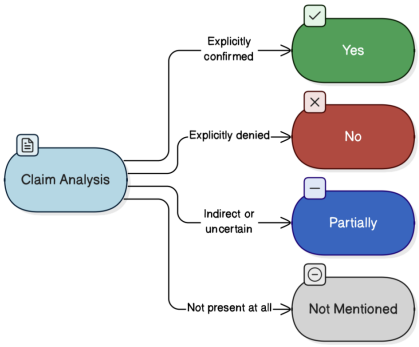


Figure 3: Claims analysis decision tree, 4 options depending on whether and how a claim is mentioned

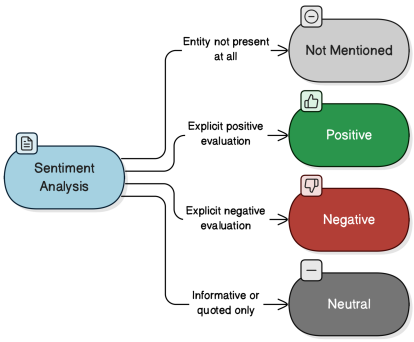


Figure 4: Decision tree in sentiment analysis

3.6 Claims Analysis

Claim analysis starts after the extraction step returns a claims list. It consists of multiple parallel LLM calls, each analyzing a single article against the claims list, using 4 categorizations for whether the article confirms the claim: Yes, Partially, No and Not mentioned, as depicted in Figure 3.

False negatives were the biggest problem we faced with claim analysis. Originally, there were only 3 claim categories; however, due to too many "not mentioned" results, we added a fourth partial classification that led to significant improvements. To further reduce false negatives without adding false positives, we tightened the categorization rules for the Not mentioned category, to default to Partial instead when answer is unclear.

Portion of the rule-set that helped improve results:

```
Before selecting "Not mentioned", you MUST check the following
transformations/hints:
- paraphrases/synonyms; hypernyms/hyponyms; abbreviations/acronyms;
  coreferences (pronouns, descriptive references)
- numbers/units/conversions; relative dates -> absolute;
  geographic hypernyms (e.g. EU -> country)
- sections: title, introduction, body, subtitles, captions,
     tables/graphs,
  quotes/indirect statements
- negations, questions, conditionals, predictions/hypotheses
Rule to reduce false negatives:
- If in doubt between "Partial" and "Not mentioned", choose "Partial"
```

3.7 Sentiment Analysis

The sentiment analysis proceeds in parallel with claims analysis after receiving the entity list (Figure 1) from the extraction. It is structured in a manner very similar to the claims analysis, it calls the LLM once per publisher, and it has 4 categorizations (Figure 4): Positive, Negative, Neutral, and Not Mentioned. Accuracy assessment is harder due to subjective interpretation. The module uses gemini-2.5-flash-lite [14] due to empirical observation of better results, every other LLM call uses gpt-4o-mini [13].

LLMs struggle with implicit criticism conveyed through selective quoting. For instance, when Mladina [15] quoted Trump praising himself as "smart" and suggesting people want a dictator, the LLM classified sentiment as positive, missing the article's critical intent to portray authoritarianism.

To account for this weakness, we added more constraints and rules in the prompts:

```
Important: OUTCOME ≠ SENTIMENT

- Do not mark "Positive" because the entity wins/makes a profit,
     without explicit value judgement of the entity.
- Do not mark "Negative" because the entity loses/has a bad result
     , without explicit value judgement of the entity.
```

```
Mandatory decision steps (before choosing a label):
- First identify the role of the entity's mention: SPEAKER /
     TARGET / MENTIONED WITHOUT ROLE
- Then look for META-EVALUATION of the entity (adjectives,
     evaluative verbs, framing before/after the quote, editorial
     tone).
- If the entity is only a SPEAKER without meta-evaluation, choose
     "Neutral".
```

This resulted in false negatives and positives reducing significantly, however it also came with the tradeoff of having a much higher incidence of neutral classification, even when it is slightly positive or negative.

3.8 Key Differences

The final step of the pipeline is the generation of the key differences (Figure 5). It uses the claims/sentiment categorizations from the previous step as input. It works for both Claims and Sentiment analysis in an almost identical manner; we will use claims for explanation in this example. A parallel LLM call is made once per every claim in the analysis, containing all claim-publisher pairs of the claim.

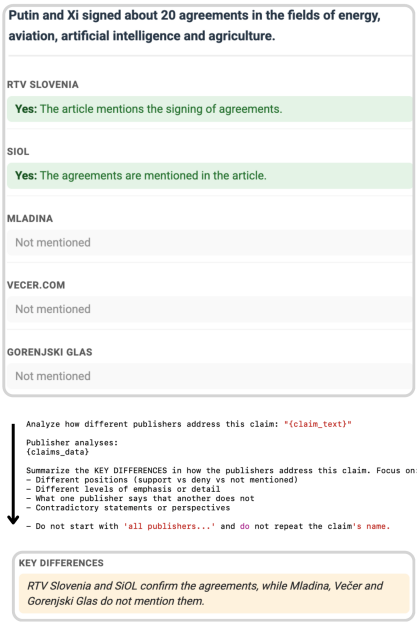


Figure 5: Key difference generation for claim from Russia-China Summit

	Hvar snakebite		Putin prepared to meet Zelenski		Carpaccio's Mary Returns to Piran		Giorgio Armani dies		Russia–China summit		Weighted avg	
	Single	**Multi**	**Single**	**Multi**	**Single**	**Multi**	**Single**	**Multi**	**Single**	**Multi**	**Single**	**Multi**
Publishers	7		7		9		7		5		—	
Claims	9	15	9	14	8	15	8	15	8	12	—	—
Error rate	25.4%	3.80%	30.15%	3.06%	38.9%	6.3%	37.5%	7.62%	32.5%	0%	**33.0%**	**4.67%**
Major errors	25.4%	1.90%	14.28%	0%	37.5%	0%	30.4%	1.91%	32.5%	0%	**28.0%**	**0.79%**
Rows affected	100%	20%	88.8%	7.14%	100%	33.3%	87.5%	33.3%	100%	0%	**95.3%**	**21.5%**

Table 2: Single-stage (left) vs. multi-stage (right) per event. Final column shows weighted averages. For error rates and major errors, weights = number of claim-publisher pairs tested per pipeline. For rows affected, weights = number of claims (rows) per pipeline. Note that weights differ between pipelines due to different extraction results.

4 Evaluation

4.1 Manual Testing

To test our hypothesis that the multi-stage pipeline is superior to a single-stage pipeline (where all articles and instructions are included in a single one prompt LLM call), we conducted a comparison of claim analysis results spanning 797 claim-publisher pairs, of which 294 are from single-stage pipeline and 503 from multi-stage pipeline. Both single and multi-stage results were generated across the same 5 control news events.

Quantitative testing was not done for sentiment due to time constraints, combined with increased difficulty due to level of subjectiveness.

Each claim-publisher pair was manually reviewed for correctness. We classified errors into two categories: minor errors (positive or not mentioned classified as partial) and major errors (false positives/negatives). Results were grouped by event to enable direct comparison between the two architectures on identical data. Weighted averages were calculated, using claim-publisher pair counts for error rates, and distinct claim counts for rows affected (Row refers to a distinct claim, and it's corresponding claim-publisher pairs).

4.2 Results

The multi-stage pipeline achieved 4.67% error rate versus the 33.0% error rate of the single-stage pipeline.2

The results table 2 shows results across the five test news events. Each percentage represents the proportion of claim-publisher pairs that were incorrectly classified. For example, in "Russia-China summit" with 5 publishers, single-stage misclassified 32.5% of all claim-publisher pairs while multi-stage achieved 0% error.

Major errors (false positives/negatives) are critical misclassifications where claims are marked "confirmed" when absent or "not mentioned" when present. Minor errors involve "partial" misclassifications. The multi-stage pipeline reduced major errors from 28.0% to 0.79%.

Rows affected shows the percentage of claims with at least one error across publishers. Single-stage produced errors in 95.3% of claims versus 21.5% for multi-stage, demonstrating more localized error patterns.

The improvement was consistent across all five news events. The most dramatic gain was the 35-fold reduction in major errors.

5 Discussion

Our results demonstrate that LLM prompt architecture fundamentally impacts LLM classification accuracy in cross-source news analysis. Significant error reduction validates task decomposition as a critical design principle for complex NLP pipelines.

While the multi-stage pipeline (Figure 2) requires more API calls (8+ versus one), costs remain manageable at $0.008-0.015 per event with both modules enabled. The accuracy improvement justifies this modest cost increase, especially considering manual verification would require expensive human labor. Considering that an event only needs to be analyzed once with no variable cost, this offers a lot of potential for analysis at scale.

Sentiment analysis struggles with irony and implicit criticism, as shown in the Mladina [15] example where selective quoting conveyed negativity despite positive surface language.

Future work includes comprehensive user testing with journalists and researchers, optimization of current modules, and expansion to other languages. We plan structured evaluations to understand how different user groups interpret and act upon cross-source comparisons.

Acknowledgments

The research described in this paper was supported by the TWON project, funded by the European Union under Horizon Europe, grant agreement No 101095095.

References

[1] Yushi Bai et al. 2023. Longbench: a bilingual, multitask benchmark for long context understanding. *arXiv preprint arXiv:2308.14508*.

[2] [n. d.] Ground news - breaking news headlines and media bias. Ground News. Retrieved Sept. 7, 2025 from https://ground.news/.

[3] [n. d.] Event registry api documentation. Event Registry. Retrieved Sept. 7, 2025 from https://eventregistry.org/documentation.

[4] [n. d.] Case study: ground news at harford community college - a collaborative mission to modernize media literacy. Library Up. Retrieved Sept. 7, 2025 from https://www.libraryup.org/news-1/case-study-ground-news-at-harford-community-college.

[5] [n. d.] Ground news - media bias and news comparison. West Virginia University Libraries. Retrieved Sept. 7, 2025 from https://libguides.wvu.edu/c.php?g=1204801&p=8818927.

[6] Gregor Leban, Blaz Fortuna, Janez Brank, and Marko Grobelnik. 2014. Event registry: learning about world events from news. In *Proceedings of the 23rd International Conference on World Wide Web* (WWW '14 Companion). ACM, Seoul, Korea, 107–110. ISBN: 978-1-4503-2745-9. doi:10.1145/2567948.2577024.

[7] C.J. Hutto and Eric Gilbert. 2014. Vader: a parsimonious rule-based model for sentiment analysis of social media text. In *Proceedings of the Eighth International AAAI Conference on Weblogs and Social Media*. AAAI Press, 216–225.

[8] [n. d.] Claude code. Anthropic. Retrieved Sept. 7, 2025 from https://claude.ai/code.

[9] Armin Ronacher. [n. d.] Flask. Retrieved Sept. 7, 2025 from https://flask.palletsprojects.com/.

[10] [n. d.] Postgresql. PostgreSQL Global Development Group. Retrieved Sept. 7, 2025 from https://www.postgresql.org/.

[11] [n. d.] Gemini api. Google. Retrieved Sept. 7, 2025 from https://ai.google.dev/.

[12] [n. d.] Openai. OpenAI. Retrieved Sept. 7, 2025 from https://openai.com/.

[13] [n. d.] Gpt-4o mini. OpenAI. Retrieved Sept. 7, 2025 from https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/.

[14] [n. d.] Gemini 2.5 flash lite. Google. Retrieved Sept. 7, 2025 from https://openrouter.ai/google/gemini-2.5-flash-lite-preview-06-17.

[15] [n. d.] Trump bi ministrstvo za obrambo preimenoval v ministrstvo za vojno. Mladina. Retrieved Sept. 7, 2025 from https://www.mladina.si/243046/trump-bi-ministrstvo-za-obrambo-preimenoval-v-ministrstvo-za-vojno/.

Identifying Social Self in Text: A Machine Learning Study

Jaya Caporusso
jaya.caporusso@ijs.si
Jožef Stefan Institute
Jožef Stefan International Postgraduate School
Ljubljana, Slovenia

Matthew Purver
Jožef Stefan Institute
Ljubljana, Slovenia
Queen Mary University of London
London, UK

Senja Pollak
Jožef Stefan Institute
Ljubljana, Slovenia

Abstract

The Self encompasses many aspects, such as the Social Self. Identifying them in text is relevant for many purposes, including mental-health research. As part of a larger project aimed at automatically detecting Self-aspects in written language, in this study we annotate and employ a dataset of diary entries to classify the presence or absence of Social Self. We train three classifiers—Support Vector Machine (SVM), Naïve Bayes, and Logistic Regression—on either learned or predefined features. The best-performing model is the SVM trained on predefined LIWC features based on a previous study. We further apply feature importance methods, and examine which features make the biggest contribution to the classification models. The most informative feature across models trained on learned features is the word "we", while the LIWC category "social referents" emerges as the most important feature for models trained on predefined features.

Keywords

social self, machine learning, classification, feature importance

1 Introduction

A central aspect of human experience, the Self is a complex, multi-aspect phenomenon [3]. Its aspects—encompassing, for example, personal narratives [18] and social interactions [2]—correlate with other relevant constructs, such as mental-health conditions [17]. While the various Self-aspects reflect in the individual's language [14], Natural Language Processing (NLP) studies rarely explore them and employ them in-depth.

This work is part of a larger project aimed at developing models to automatically identify Self-aspects in text, with applications in mental-health-research and empirical phenomenology [5]. Due to the sensitive nature of the domains of application, we attempt an approach that allows both interpretability and ground-truth basis, opting for classical machine learning (ML) models. In this study, we focus on one Self-aspect: the Social Self (SS), defined as *the Self as it is shaped and/or perceived when in an interaction or relationship of sorts with other people or entities to whom we attribute qualities of inner life* [4]. We aim to investigate how this is represented in diary entries and whether these representations can be reliably identified using machine learning. Additionally, we explore which linguistic features are most predictive of these aspects. Identifying SS in text is valuable, as, e.g., disturbances in the SS are closely linked to mental health conditions [7]. This project involves labelling—with a mixed approach involving human annotators and large language models (LLMs)—diary entry instances as binary (representing or not) SS, with the purpose of investigating the correlation between SS and textual features. We train and compare three classifiers (i.e., Support Vector Machine (SVM), Naïve Bayes (NB), and Logistic Regression (LR)) to predict SS using either 1) learned features (i.e., TF-IDF unigrams and bigrams) or 2) predefined features (i.e., Linguistic Inquiry and Word Count (LIWC; [1]) lexicon categories (see [4]). We use the mentioned classifiers instead of LLMs (e.g., GPT-4) because our focus is on employing interpretable features and understanding their contribution to predictions—an aspect less directly accessible in generative models. We conduct feature importance analysis to explore these contributions further. The code is available at https://github.com/jayacaporusso/SELFtext upon request.

Information Society 2025, Ljubljana, Slovenia

https://doi.org/10.70314/is.2025.sikdd.2

2 Related Work

Studies that address the correlation between text and the traits and states of the text's author often utilise the Linguistic Inquiry and Word Count (LIWC), a text analysis software developed to analyse linguistic and psychosocial constructs connected to various textual aspects [1] (e.g., [9]). Various studies have found Self states to be associated with linguistic features, e.g., depression with first-person singular pronouns [15]. This has been employed in classification tasks (e.g., [6]). In a previous study, after labelling a dataset with a mixed approach employing human annotation and LLMs, we analysed which LIWC-22 features characterise Reddit posts including Self as an Agent, Bodily Self, and SS [4]. Specifically, we showed that the **presence of SS** is correlated with LIWC categories including, among the others, *emotion* and *time related terms*. In contrast, the **absence of SS** is correlated with, e.g., *technology* and *negative emotions*. In this work, we employ this knowledge to build SS classifiers on predefined features and compare them with classifiers trained on learned features.

3 Research Questions

In this study, we aim to address the following main research questions (RQs). **RQ1**: How does a SS classifier trained on predefined features perform compared to a SS classifier trained on learned features? **RQ2**: Among the algorithms employed, which one performs better for our task? **RQ3**: Which features are more relevant for the classification of SS?

3.1 Labelling

In our study, we use a publicly available dataset in English [11] comprising 1,473 text samples (sub-entries; average length: 507.6 characters, 100.6 words) from 500 personal journal entries (500 anonymous subjects). We augment the dataset with binary labels for SS, as following addressed.

For labelling, we employ a mixed approach (see [4]) that combines human annotation with the large language model (LLM)

gemma2 [16]. The instructions for manual annotation are provided in the Appendix A. Two human annotators label the first 105 instances of the dataset. This is needed to calculate inter-annotator agreement with the LLM annotations. We instruct gemma2 to label the data three times, providing three different personalisations (see [10]): expert in phenomenology, cognitive psychology, or social psychology. Additionally, we provide them with definitions of SS, instructions to annotate it, examples of a text instance where it is present, a text instance where it is absent, and explanations of why this is so. These can be extracted from the instructions for manual annotation. Each gemma2 model performs a one-shot, binary classification for each self-aspect. We calculate majority voting with the resulting labels and compute the inter-annotator agreement between each pair among the human and the LLM annotators by calculating Cohen's Kappa coefficient. This results in Cohen's Kappa coefficients of 0.80 (human annotators), 0.89 (first annotator vs. gemma2), and 0.84 (second annotator vs. gemma2). In the further steps, we use the majority voting labels. The class balance (calculated on the majority voting) is 50.3% (SS present) vs 49.7% (SS not present).

4 Classification

The text is preprocessed, converting it to lowercase and removing punctuation and extra whitespace. We extract learned and predefined features. We then train three classifiers for each set of features: an SVM, a NB, and a LR model.

4.1 Feature Engineering

We are interested in comparing the performance of models trained on learned vs pre-defined features. In this study, we choose to employ TF-IDF calculated on unigrams and bigrams as learned features, and the LIWC features identified as being related to the presence or absence of SS in Caporusso et al. [4].

4.1.1 Learned Features. To extract learned features, we employ TfidfVectorizer, applying TF-IDF weighting to unigrams and bigrams. Restricting the representation to unigrams and bigrams, a common choice in exploratory text classification, efficiently displays feature importance, balancing interpretability and computational efficiency. We limit the feature space to the 1000 n-grams that, based on their TF-IDF scores, are the most informative. This ensures computational efficiency. In this process, we choose not to exclude stopping words. Indeed, for the purpose of our study, they do not merely constitute noise but might play a key role in distinguishing text instances reporting on SS.

4.1.2 Predefined Features. We analyse the presence of all the LIWC-22 [1] categories and subcategories, and subsequently only considered the LIWC features of interest. Specifically, as predefined features, we employ the LIWC features that Caporusso et al. [4] identified as being related to the presence and absence of SS (see 2), for example *authenticity*, *social referents*, and *the pronoun I*. For each of them, LIWC-22 provides scores relative to the text length. All LIWC features were standardised using Z-score normalisation to ensure comparability across different feature scales. This is particularly important for models like SVM and LR, which are sensitive to feature magnitudes. Missing values (NaNs) are handled using mean imputation.

4.2 Models

The models are trained and evaluated using 10-fold cross-validation to assess their performance. Specifically, we train three models on the learned features and three models on the predefined features. The models are of three different kinds: SVM, NB, and LR, all commonly used in text classification tasks. We employ default hyperparameters. For the SVM, we use Linear kernel. For LR, we apply L2 regularisation, which adds a penalty term to the model's objective function, minimising overfitting. For NB, MultinomialNB was used for learned features, while GaussianNB was used for predefined features, which consist of continuous numerical values derived from linguistic analysis. MultinomialNB assumes that features represent discrete frequency counts, while GaussianNB assumes that feature distributions follow a normal distribution, making it appropriate for continuous data.

5 Evaluation

Similarly to the training process, the models are evaluated using 10-fold cross-validation. All the models perform reasonably well, with the SVM model trained on predefined features outperforming them all (RQ1 and RQ2). The metrics (precision, recall, and F1-score: mean and STD) across folds are reported in Table 1. They match the macro average scores. The confusion matrices for each model are presented in Figures 3 and 4 in the Appendix B. These highlight that models trained on predefined features generally perform better at distinguishing between classes, with the SVM and LR models achieving higher accuracy for both Class 0 and Class 1. However, NB trained on predefined features struggles with a higher rate of false positives for Class 0. The models trained on learned features have slightly lower performance, with higher misclassification rates for Class 1 predictions. After performing a Friedman test across folds (statistic = 44.26, p-value = 0.00), we find a statistically significant difference in model performances. We therefore conduct Wilcoxon signed-rank tests with Bonferroni correction to identify significant pairwise differences between models. LR with learned features performed significantly better than NB with learned features ($p = 0.03$); SVM with predefined features outperforms NB with learned features ($p = 0.03$); LR with predefined features outperforms NB with learned features ($p = 0.03$); SVM with predefined features performs significantly better than NB with predefined features ($p = 0.03$); LR with predefined features outperforms NB with predefined features ($p = 0.03$). The results are displayed in Figure 5 in the Appendix B.

	Precision	Recall	F1-Score
SVM_TFIDF	0.83 (0.03)	0.81 (0.03)	0.81 (0.03)
NB_TFIDF	0.80 (0.03)	0.79 (0.03)	0.79 (0.03)
LR_TFIDF	0.82 (0.03)	0.82 (0.03)	0.82 (0.03)
SVM_LIWC	**0.83 (0.03)**	**0.83 (0.03)**	**0.83 (0.03)**
NB_LIWC	0.76 (0.4)	0.75 (0.04)	0.75 (0.04)
LR_LIWC	0.83 (0.03)	0.83 (0.03)	0.82 (0.03)

Table 1: Evaluation Metrics (Mean and STD)

6 Feature Importance

We employ different feature importance methods tailored to each model's learning mechanism to ensure that feature rankings are meaningful and aligned with the way each algorithm processes data. For the SVM models, we choose Linear SVM Coefficients because they directly represent feature importance in the decision boundary and are computationally efficient to extract. This method is fast and directly interpretable without requiring additional computations, but it does not capture feature interactions

or non-linearity. For the NB models, we choose Permutation Importance. NB does not have meaningful coefficients, and this method provides a model-agnostic way to assess how each feature affects predictions. This method allows the interpretation of feature contributions without relying on the model's internal parameters, but it is computationally expensive and can be sensitive to correlated features. For the LR models, we choose SHAP (SHapley Additive exPlanations [12]) Values, because they provide both global and instance-level feature attributions while considering feature interactions, making them more informative than raw coefficients. SHAP accounts for feature dependencies and offers a nuanced interpretation of how features contribute to individual predictions, but its computations can be slow and the results depend on the reference distribution used. Using SHAP for the SVM would be unnecessary because it would give similar results as the coefficients but less directly and with added computational cost, while SHAP's dependency assumptions conflict with NB's independence assumption. The contribution of each feature to the classification decision is indicated with a feature importance score. These are computed differently depending on the method: in Linear SVM Coefficients, they are derived from the absolute magnitude of the learned weights; in Permutation Importance, they are measured by assessing the decrease in model performance when a feature's values are randomly shuffled; while in SHAP, they quantify the contribution of each feature to the predicted classification probability by distributing the model's output among the input features.

6.1 SVM: Linear SVM Coefficients

For SVM, feature importance is determined using Linear SVM Coefficients. This method is chosen because linear SVM explicitly learns a set of coefficients as part of its optimisation process, making feature importance inherently interpretable. Additionally, since the SVM model is optimised to find the maximum margin, features with the largest coefficients contribute the most to defining this separation, allowing for a clear ranking of feature relevance. The resulting importance scores are based on the absolute magnitude of the learned coefficients, and like them, they can be any real value. While the importance scores' scale depends on the range of the input features, higher numbers indicate a stronger influence on classification. The top-3 features for the SVM models are *family, we*, and *with* (TF-IDF) and *social referents, I*, and *personal pronouns* (LIWC) (RQ3).

6.2 Naïve Bayes: Permutation Importance

For NB, we choose Permutation Importance because it provides a robust way to assess feature significance in probabilistic models that do not generate explicit importance scores. By quantifying the dependence of the model's predictions on each feature, Permutation Importance allows for an intuitive understanding of which features are most influential in the NB classification process. The scores produced are relative, and their scale depends on the model's performance metric; a larger value indicates that the feature has a greater impact on classification accuracy. The top-3 features for the NB models are *us, birthday*, and *her* (TF-IDF) and *social referents, social*, and *she/he* (LIWC) (RQ3).

6.3 Logistic Regression: SHAP Values

LR calculates the probability of a given outcome using a linear combination of input features, but SHAP offers a more granular and interpretable way of explaining these predictions. This method is chosen because it provides a comprehensive, intuitive, and theoretically grounded measure of feature importance, making it well-suited for interpreting the decision-making process of a probabilistic model like LR. In this study, we reduce the SHAP computation sample size from 50 to 20 to improve efficiency while maintaining representative feature importance insights. SHAP scores are measured in the same scale as the model's output and sum to the difference between the model's output and the expected output across all features. They can be positive (probability of classification increased) or negative (probability of classification decreased). Their magnitude reflects the strength of the feature's influence on the classification decision. The top-3 features for the SVM models are *with, we*, and *my* (TF-IDF) and *social referents, Social*, and *I* (LIWC) (RQ3).

6.4 Overall feature importance

To determine the top-20 most important features across all models trained on learned features and across all models trained on predefined features, we aggregate the feature importance scores from each model and sum them across all models. This is done to show which features are consistently influential regardless of the model; however, due to differences in how each method computes importance, the aggregated scores should be viewed as indicative rather than absolute measures of feature relevance. The top-10 features for the models trained on learned features are displayed in Figure 1, while those for the models trained on predefined features in Figure 2 (RQ3). Additionally, we identify unique features for each model, defined as those that appear in the top-10 for a specific model but not in others. Following, we report those referring to models trained on learned features.

- **SVM**: *my, team, she, our, he, we, with, friend, with my, their.*
- **Naïve Bayes**: *team, they are, he was, us, birthday, she was, of our, with her, person, spending time.*
- **Logistic Regression**: *my, she, our, and, good, he, my family, we, it, sleep.*

Following, we report those referring to models trained on predefined features.

- **SVM**: *sexual, Dic, Social, socrefs, feeling, we, Affect, Drives, insight, WC.*
- **Naïve Bayes**: *Dic, Social, socrefs, number, moral, feeling, we, focuspast, Drives, illness.*
- **Logistic Regression**: *Dic, Social, socrefs, pronoun, Analytic, feeling, we, Affect, focuspast, Drives.*

This helps us shed light on how different algorithms interpret the data; some overlap in the reported features occurs because the different algorithms, despite using distinct mechanisms to estimate importance, converge on similar cues that are consistently predictive of SS. We calculate the correlation between feature importance rankings across the different models by computing the Pearson correlation coefficient between the feature importance scores of each pair of models, using their respective importance values across all features. This is displayed in Figures 6 and 7 in the Appendix C. A high positive correlation indicates similar feature rankings and vice versa. The highest correlation is measured between SVM and LR models, while the lowest between NB and LR for models trained on learned features, and between SVM and NB for models trained on predefined features.

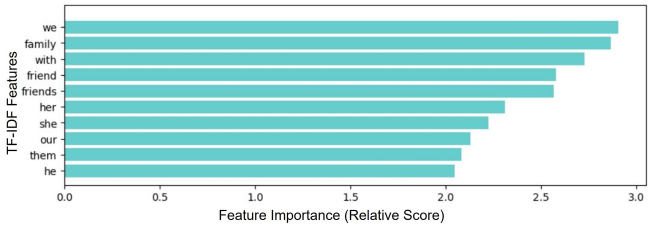


Figure 1: Top-10 Features for TF-IDF Models

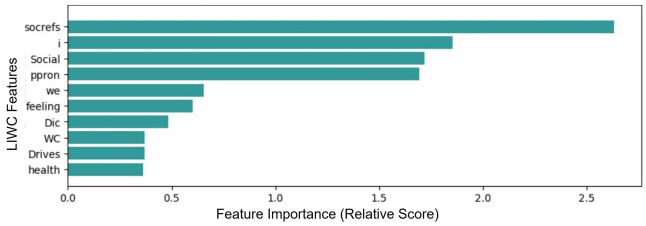


Figure 2: Top-10 Features for LIWC Models

7 Discussion

Our results indicate that the models trained on predefined features (LIWC) generally outperform those trained on learned features (TF-IDF n-grams), with the SVM model achieving the highest classification performance (RQ1-2). This suggests that LIWC features, which encapsulate linguistic and psychological constructs, provide a structured and interpretable representation of textual patterns related to SS. In contrast, TF-IDF captures surface-level word frequency distributions, which may be more susceptible to noise and context variability, limiting its predictive power for capturing abstract constructs like SS. Furthermore, our results support the findings by Caporusso et al. [4] regarding LIWC features correlated with SS. Notably, models trained on TF-IDF features tend to exhibit higher aggregated feature importance scores compared to those trained on LIWC. This could be attributed to the fact that TF-IDF operates on a larger and more granular feature space, capturing subtle variations in word usage. As a result, many features contribute partially to model decisions, leading to a higher sum of importance values across all features. In contrast, LIWC features are more constrained and predefined, leading to more concentrated but lower cumulative importance scores. This suggests that while TF-IDF captures a broader spectrum of textual variations, LIWC provides a more targeted and structured linguistic representation. Many of the features identified as relevant for the classification of SS (e.g., *we* and *social referents*) intuitively align with the nature of SS (RQ3).

8 Limitations and Future Work

This study serves as a pilot for the interpretable classification of different Self aspects in text, focusing on SS. Several areas for improvement remain. Clearer annotation guidelines are needed for consistency. The choice of restricting to linear models, LIWC features, and unigrams/bigrams was appropriate for this exploratory study prioritising interpretability; however, it inevitably limits performance and representational richness. In future work, we plan to complement this approach with more powerful models and richer feature sets (e.g., embeddings). Here we wanted to compare models trained on learned vs predefined features, but we plan to train models on both. While in this study we did not perform hyperparameter optimisation, we will do so in the future. We aim to train a neural network for multi-class classification, enabling simultaneous prediction of SS and other Self-aspects, allowing for a more comprehensive analysis of self-representation in text. In the future, we plan to employ different datasets and implement Demšar's evaluation method [8]. Our long-term goal is to be able, given a text instance, to determine what Self aspects are present and how they are expressed, in an explainable manner. To do so, it is not only necessary to extend our work to other Self-aspects, but to move beyond a binary classification for each of them. Work on the ontology underpinning future studies is ongoing [13].

9 Acknowledgments

We acknowledge Špela Rot's assistance and the financial support from the Slovenian Research Agency for research core funding for the programme Knowledge Technologies (No. P2-0103) and from the projects CroDeCo (J6-60109), Shapes of Shame in Slovene Literature (J6-60113), and Natural Language Processing for Corpus Analysis in the Medical Humanities (BI-VB/25-27-021). JC is a recipient of the Young Researcher Grant PR-13409.

References

[1] Ryan L Boyd, Ashwini Ashokkumar, Sarah Seraj, and James W Pennebaker. 2022. The development and psychometric properties of liwc-22. *Austin, TX: University of Texas at Austin*, 10.

[2] Marilynn B Brewer. 2002. Individual self, relational self, and collective self: partners, opponents, or strangers. (2002).

[3] Jaya Caporusso. 2022. Dissolution experiences and the experience of the self: an empirical phenomenological investigation (master's thesis). university of vienna. *Advisor: Assist. Prof. Dr. Maja Smrdu*.

[4] Jaya Caporusso, Boshko Koloski, Maša Rebernik, Senja Pollak, and Matthew Purver. 2024. A phenomenologically-inspired computational analysis of self-categories in text. In *Proceedings of JADT 2024*. Vol. 1, 169–178.

[5] Jaya Caporusso, Matthew Purver, and Senja Pollak. 2025. A computational framework to identify self-aspects in text. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop)*. Jin Zhao, Mingyang Wang, and Zhu Liu, editors. Association for Computational Linguistics, Vienna, Austria, (July 2025), 725–739. ISBN: 979-8-89176-254-1. DOI: 10.18653/v1/2025.acl-srw.47.

[6] Jaya Caporusso, Thi Hong Hanh Tran, and Senja Pollak. 2023. Ijs@ lt-edi: ensemble approaches to detect signs of depression from social media text. In *Proceedings of the Third Workshop on Language Technology for Equality, Diversity and Inclusion*, 172–178.

[7] Christopher G Davey and Ben J Harrison. 2022. The self on its axis: a framework for understanding depression. *Translational Psychiatry*, 12, 1, 23.

[8] Janez Demšar. 2006. Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine learning research*, 7, 1–30.

[9] Lewis R Goldberg. 2013. An alternative "description of personality": the big-five factor structure. In *Personality and Personality Disorders*. Routledge, 34–47.

[10] Boshko Koloski, Nada Lavrač, Bojan Cestnik, Senja Pollak, Blaž Škrlj, and Andrej Kastrin. 2024. Aham: adapt, help, ask, model harvesting llms for literature mining. In *International Symposium on Intelligent Data Analysis*. Springer, 254–265.

[11] X Alice Li and Devi Parikh. 2019. Lemotif: an affective visual journal using deep neural networks. *arXiv preprint arXiv:1903.07766*.

[12] Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30.

[13] Luka Oprešnik, Tia Križan, and Jaya Caporusso. 2025. Building an ontology of the self: sense of agency and bodily self. In *Proceedings of Information Society 2025*. Cognitive Science. DOI: 10.70314/is.2025.cogni.8.

[14] James W Pennebaker, Matthias R Mehl, and Kate G Niederhoffer. 2003. Psychological aspects of natural language use: our words, our selves. *Annual review of psychology*, 54, 1, 547–577.

[15] Stephanie Rude, Eva-Maria Gortner, and James Pennebaker. 2004. Language use of depressed and depression-vulnerable college students. *Cognition & Emotion*, 18, 8, 1121–1133.

[16] Gemma Team et al. 2024. Gemma 2: improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.

[17] David HV Vogel, Mathis Jording, Peter H Weiss, and Kai Vogeley. 2024. Temporal binding and sense of agency in major depression. *Frontiers in psychiatry*, 15, 1288674.

[18] Dan Zahavi. 2007. Self and other: the limits of narrative understanding. *Royal Institute of Philosophy Supplements*, 60, 179–202.

A Instructions for Labelling: Social Self

In the column relative to Social Self, insert:

- **0**: if the Social Self is not present.
- **1**: if the Social Self is present.

Following, we provide a definition of Social Self [4], instructions, and examples of a text instance where it is present and a text instance where it is not present, taken from the dataset to be labelled:

Definition: The Self as it is shaped and/or perceived when in an interaction or relationship of sorts with other people or entities to whom we attribute qualities of an inner life.

Instructions

For *Social Self* to be present in a text instance it is not enough for the text instance to contain references to other people and/or entities, but it has to contain mentions of the author's interactions with them, influence on them, or influence they have on the author. This can be even minimal, e.g., in the form of referring to a person as *my sister*, or by using the first-person plural pronoun instead of the singular one.

Examples

A.0.1 Text instance containing Social Self: "My family was the most salient part of my day, since most days the care of my 2 children occupies the majority of my time. They are 2 years old and 7 months and I love them, but they also require so much attention that my anxiety is higher than ever. I am often overwhelmed by the care they require, but at the same, I am so excited to see them hit developmental and social milestones."

Explanation of text instance with Social Self present: In this text instance, the author report on other people they are in some sort of relationship with, and about some aspects of their relationship and how they make the author feel.

A.0.2 Text instance not containing Social Self: "Yoga keeps me focused. I am able to take some time for me and breathe and work my body. This is important because it sets up my mood for the whole day."

Explanation of text instance with Social Self not present: In this text instance, the author does not report on any person, animal, or other entities to whom we attribute qualities of inner life.

General Notes While a certain Self-aspect might not be prominently present in a text instance in its entirety, if it is present in a part of the text instance to be labelled, then it has to be labelled as present in the text instance. A given text instance can have none of the Self-aspects present, one of them present and two of them non-present, two present and one non-present, or all three of them present—any combination is possible.

B Evaluation

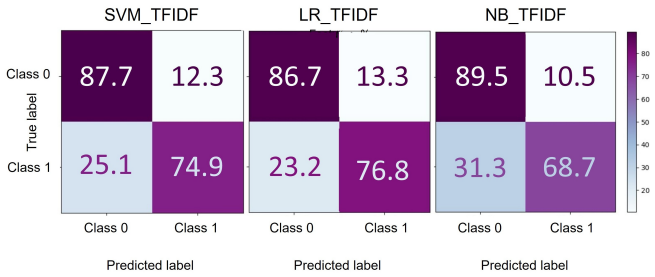


Figure 3: Confusion Matrices: Models Trained on Learned Features (TF-IDF)

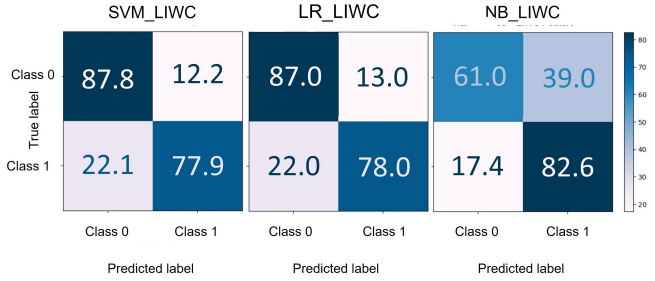


Figure 4: Confusion Matrices: Models Trained on Predefined Features (LIWC)

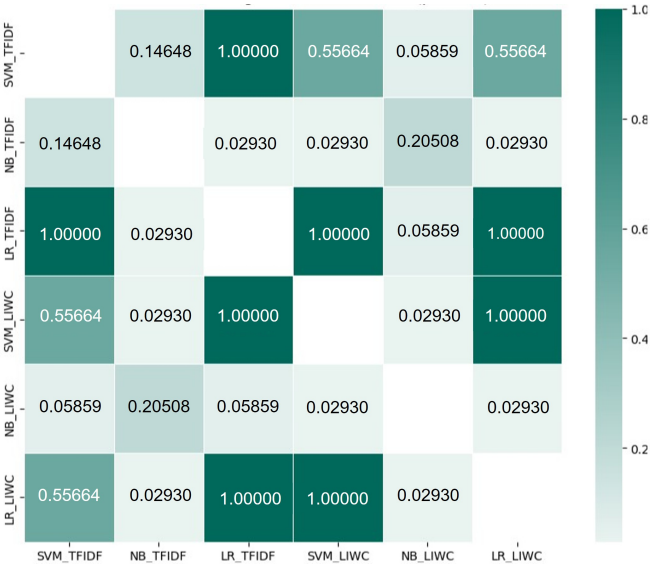


Figure 5: Pairwise Wilcoxon Signed-Rank Test Results (p-values)

C Feature Importance

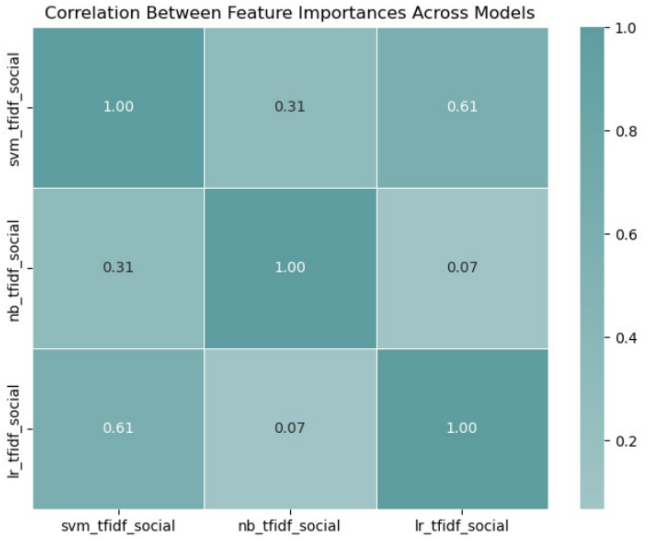


Figure 6: Correlation Between Feature Importance Across Models Trained on Learned Features (TF-IDF)

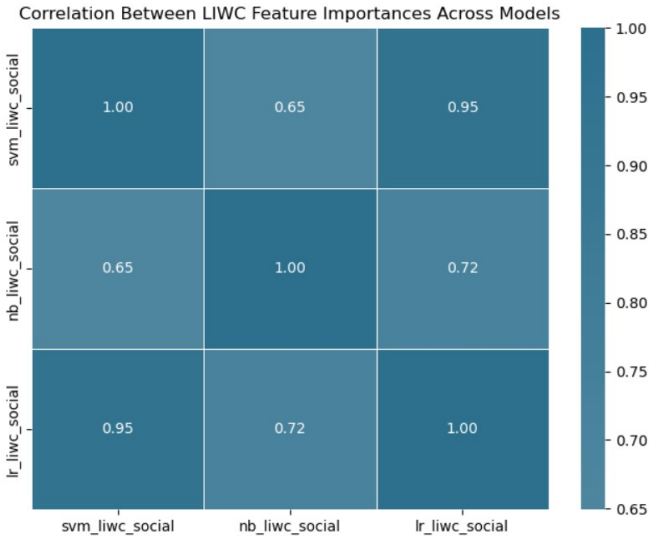


Figure 7: Correlation Between Feature Importance Across Models Trained on Pre-Defined Features (LIWC)

WinWin Meets – Investigating the Future of Online Meetings

Martin Žust
marti.zust@gmail.com
Jožef Stefan Institute
Ljubljana, Slovenia

Marko Grobelnik
marko.grobelnik@ijs.si
Jožef Stefan Institute
Ljubljana, Slovenia

Alenka Guček
alenka.gucek@ijs.si
Jožef Stefan Institute
Ljubljana, Slovenia

Adrian Mladenic Grobelnik
adrian.m.grobelnik@ijs.si
Jožef Stefan Institute
Ljubljana, Slovenia

Abstract

Video conferencing is now central to modern collaboration, yet its functionality remains largely limited to passive audio–visual communication. Despite growing investment in artificial intelligence (AI), it is unclear which features truly enhance meetings and how users will adopt them. Here we present WinWin Meets, a Jitsi-based prototype that integrates Whisper transcription and GPT-4o processing to deliver real-time summaries, visual mind maps, and goal-oriented advice. Testing with 16 participants showed strong interest in summaries and mind maps, moderate interest in in-meeting guidance, and a preference for add-on integration. Market research confirmed low organic demand for advanced AI features, with users prioritizing reliable improvements such as automated notes. These results highlight a gap between experimental enthusiasm and everyday adoption, pointing to opportunities for targeted, industry-specific integrations that combine reliability with intelligent support.

Keywords

video conferencing, AI agent, testing, market research, zoom, negotiation, transcription, summarization, advice, meeting notes, AI innovations

1 Introduction

As artificial intelligence advances rapidly, its potential to transform everyday digital tools, particularly video conferencing, has become increasingly apparent. Platforms such as Zoom, Google Meet, and Microsoft Teams have become standard, yet their functionality remains focused on basic communication. A new need is arising for next-generation conferencing, including intelligent assistants, automatic summarization, content analysis, and contextual support. These next-generation systems go beyond passive audio and video transmission to actively support users with intelligent features and real-time analysis [1].

Previous research reveals both promise and challenges. Proactive AI meeting assistants can improve efficiency but need to balance autonomy with what users are willing to accept [1]. Meanwhile, studies of speech-based technology underscore the difficulty of extracting useful outcomes from nuanced group interactions [2]. These perspectives suggest that AI's success in meetings depends on technical feasibility and sensitivity to human collaboration.

With remote meetings now central to how we work, these systems directly impact productivity, collaboration, and organizational culture. This paper explores which functionalities could define the future of video conferencing and how AI may contribute. We combine market trend and user preference analysis, reviews of online discussions, and experimental testing of the WinWin Meets prototype. We explore which features matter to users, examine how AI can support meetings, and assess the potential to improve efficiency, clarity, and structure in digital communication.

2 Background and Related Work

2.1 Overview of Current Video Conferencing Solutions

The video conferencing market is currently dominated by a few major players. Zoom, Microsoft Teams, and Google Meet together account for approximately 94% of global market share, with Zoom alone holding around 56% [3]. While all three platforms are actively investing in artificial intelligence features, their innovation must be carefully balanced with the risk of reputational damage. As established brands, they face more constraints than lesser-known startups, which can afford a higher level of experimental agility. This creates a unique window of opportunity for the emergence of disruptive technologies that have the potential to redefine the video conferencing experience.

Most AI-enabled tools developed recently are not standalone platforms, but integrations designed to work alongside existing services like Zoom, Google Meet, or Microsoft Teams. Notable examples include tl;dv [4], Otter.ai [5], Fathom [6], Fireflies [7], and Sembly AI [8]. These applications primarily offer meeting transcription, and some provide more advanced analytics such as sentiment analysis or participant-level speaking time metrics.

2.2 Limitations of Existing Solutions and Emerging Needs

Despite the growing number of AI integrations, fully independent platforms that natively combine video conferencing with built-in AI features remain rare. These features may include real-time transcription, intelligent meeting summarization, and contextual AI-generated recommendations. This segment remains underdeveloped, presenting a significant opportunity for innovation.

While major platforms like Zoom have started introducing their own AI assistants (e.g., Zoom AI Companion [9]), they must innovate cautiously to protect their reputation and user base. This creates space for new companies to develop more ambitious

Information Society 2025, Ljubljana, Slovenia

https://doi.org/https://doi.org/10.70314/is.2025.sikdd.14

AI-first conferencing tools, unrestricted by established brand expectations or legacy user commitments. However, innovating in markets where most users are already committed to existing platforms has notable downsides. Only about 2.5% of people are actively seeking new alternatives, with the majority being reluctant to change [10].

3 Development of WinWin Meets

3.1 Overview

As part of our research, we developed WinWin Meets, an AI-based alternative to Zoom. The application maintains familiar functionality, allowing users to start or join meetings just as they would expect. The key difference comes before entering the meeting room, where users can define their meeting goals. Once inside, they find a familiar interface with standard video conferencing features.

These core functionalities are provided through an integration with Jitsi [11], an open-source video conferencing platform. It supports screen sharing, microphone and camera toggling, chat-based communication, polls, and many other standard features.

Beyond the familiar main meeting window found in applications like Zoom, WinWin Meets adds a dedicated panel on the right side of the screen for the WinWin Agent. This panel features two main buttons: Summarize and Give Advice.

The Summarize button generates meeting summaries up to the current moment, particularly useful for late arrivals. Hovering reveals three options: Short Text, Long Text, and Mind Map. While the text options provide traditional summaries of varying length, the Mind Map offers a quicker and more accessible visual overview. The idea behind the mind map is based on the observation that modern workplace attention is highly fragmented, with a median focus duration of just 40 seconds on any screen [12].

The Give Advice button offers guidance on how to achieve the goals specified before the meeting. These goals can also be adjusted during the meeting by clicking the Manage Goals button in the top right corner. Hovering over the Give Advice button reveals three options: Short Text, Medium Text, and Long Text, which provide advice in different levels of detail.

Once the meeting concludes, a meeting report is quickly generated. The report includes all key points, action items, a meeting timeline, and the list of participants. Users can also generate a mind map from the final meeting content.

3.2 System Architecture and Implementation

The frontend of the application was developed in Cursor [13], with assistance from Claude 3.7 Sonnet [14] and GPT-4o [15]. It is built using the React 19 framework [16]. We aim for a clean and minimalistic design that intuitively guides the user through each step of the interface.

In the meeting room interface, we integrated Jitsi via its iframe API. Jitsi integration is straightforward, and the platform allows the use of its hosted servers for up to 25 active monthly users free of charge, which was sufficient for our prototype testing.

The backend is built in Python, using the FastAPI framework [17]. For transcription, we integrated Whisper [18], and for natural language processing tasks (such as summarization and advice generation), we used GPT-4o. The backend exposes several endpoints, including:

- Transcription
- Advice generation
- Meeting summarization
- Health monitoring
- Meeting notes
- File uploads

The WinWin Agent dynamically adapts to the language selected by the user. In this prototype, we supported English, German, and Slovene, allowing users to interact with the summarization and advice features in their preferred language.

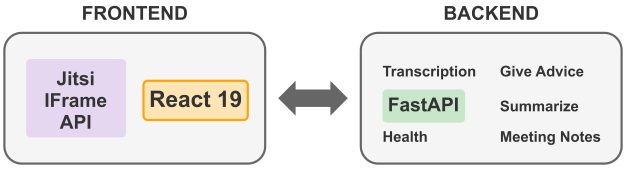


Figure 1: System architecture of the WinWin Meets application

In this prototype version, we did not use any persistent database; all data is stored locally. Additionally, user authentication is not yet implemented, as the focus was on demonstrating core functionalities.

4 Testing and User Insights

To evaluate the usefulness and usability of WinWin Meets, we conducted a structured user testing process involving 16 participants. Testing sessions were held in small groups of 2 to 4 participants, each lasting approximately 15 minutes. Participants simulated realistic discussions—including casual exchanges and role-play scenarios such as negotiations or political debates—to test all implemented functionalities. The following sections present our testing results, with key findings shown in Figure 2.

4.1 Test Coverage

Participants explored all key features, including the three variants of the Summarize function (Short Text, Long Text, and Mind Map), the three formats of the Give Advice function (Short, Medium, Long), and the Meeting Notes feature. After each session, they completed an anonymous survey with both multiple-choice and open-ended questions to assess usefulness and provide feedback.

4.2 Key Findings

General Usefulness
Most participants recognized the potential of AI-enhanced meetings. In fact, 87.5% responded *Yes* when asked whether AI could help them achieve meeting goals, while the remaining 12.5% answered *Maybe*.

Summarize Feature
The Summarize function was considered useful by 81.3% of participants. Preferences were split almost evenly: nearly half favored the Short Text, another 43.8% opted for the Mind Map, while only 12.5% selected the Long Text variant.

Give Advice Feature
When choosing advice length, participants showed a clear preference for medium-length suggestions:

- 50% selected Medium
- 25% chose Short
- 25% chose Long

Meeting Notes Feature
Participants emphasized three expectations for meeting notes:

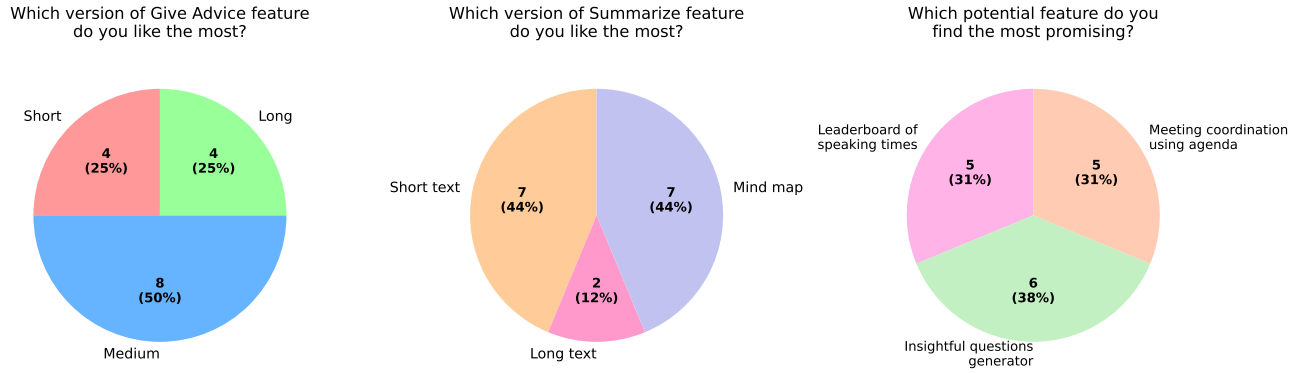


Figure 2: User survey results (n=16) comparing preferences for existing features (Give Advice and Summarize) and ranking of proposed new features for application WinWin Meets

- High reliability (timestamps, content accuracy)
- Fast post-meeting availability
- Stable performance across sessions

4.3 Ideas for Additional Features

Among the proposed additions, the insightful question generator attracted the most interest (37.5%), while the speaking time leaderboard and agenda-based coordination were equally valued (31.3% each). Participants also suggested several custom features, including personal notes, live transcription export, cloud synchronization, calendar integration, live translation with tone analysis, and domain-specific modes for law, sales, or education.

4.4 Integration Preferences

A clear majority (68.8%) preferred to use WinWin Meets as an add-on to existing platforms, while only 31.2% supported a standalone application.

4.5 Use Cases by Industry

Participants identified several promising domains for WinWin Meets, such as negotiation and sales, legal and consulting services, corporate meetings, academic events, client feedback sessions, NGO coordination, and specialized contexts like logistics, mergers and acquisitions, or trade deals.

5 Market Research and Trend Analysis

Beyond developing and testing WinWin Meets, we conducted market research to understand user needs and expectations in the video conferencing space. Our approach combined online surveys, social media engagement, search trend analysis, and reviews of blog posts and user forums. This investigation aimed to reach a wider audience than application testing alone could provide. The resulting quantitative and qualitative insights complement rather than replace our user testing results.

5.1 Survey and Social Media Feedback

Informal polls and surveys were conducted on platforms such as Facebook and Reddit. In a Facebook group focused on digital tools (GrowthHacking Slovenia), a poll asking users which feature they would most like to add to Zoom revealed that over 60% of respondents preferred having meeting notes generated at the end of a call as we can see in Figure 3. In contrast, only two respondents selected a real-time AI assistant. This suggests a clear user preference for simple and familiar enhancements over more complex and unfamiliar innovations.

Similar sentiment was observed on Reddit (r/Zoom and r/remotework), where posted polls received limited engagement. Among the few responses, a general disinterest in AI-based meeting assistance was evident, with some users explicitly selecting “None of those”.

5.2 Search Behavior and Online Interest Trends

Public search trends were analyzed using tools such as Answer the Public [19], Answer Socrates [20], AlsoAsked [21], and Ubersuggest [22]. These platforms provided insight into the types of questions users search for on Google, YouTube, and Reddit. The analysis showed minimal interest in AI-enhanced conferencing features. Instead, users were more focused on improving the efficiency and effectiveness of their meetings.

Popular search queries we found included:

- What are the 3 C’s of effective meetings?
- What is the 10-10-10 rule for meetings?
- How can I take better meeting notes?
- What are the 5 P’s of meeting productivity?
- How to extend the 40-minute limit on Zoom?
- Is Google Meet better than Zoom?
- Is Zoom free to install and use?

These patterns confirm that users are primarily concerned with meeting outcomes and platform reliability, rather than with novel AI-driven functionalities.

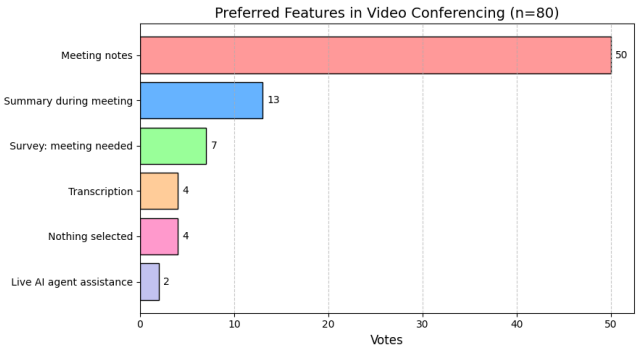


Figure 3: Distribution of 80 votes for preferred video conferencing features from our informal polling.

5.3 Forum Discussions and Deep-Search Insights

Using tools like Grok [23] and Floth [24], we conducted a deeper exploration of online discussions and feedback. The most frequently mentioned user pain points include:

- Low video quality and unstable connections
- Privacy concerns (e.g., Zoom bombing, data storage policies)
- Psychological fatigue from constant camera presence
- Lack of end-to-end encryption and transparency
- Poor UX from interface changes (e.g., Google Meet "floating bubbles", Webex chat restrictions)
- Discomfort with platform claims over recorded content

User feedback highlights a desire for reliable, simple, and secure platforms with minimal friction in setup and usage.

5.4 Conclusions from Market Research

Our market analysis reveals several key trends:

(1) Users strongly prefer practical features like note-taking and agenda management over complex AI-based tools.
(2) Popular search queries suggest a need for structured meeting frameworks and productivity strategies.
(3) Persistent dissatisfaction exists around technical reliability, interface design, and data privacy.
(4) Open-source alternatives offer control and security but are hindered by usability and cost barriers.

Overall, the market exhibits demand for video conferencing improvements that enhance meeting effectiveness and reduce user burden, rather than introducing new technical complexity.

6 Discussion

There are two primary approaches to understanding user preferences: direct inquiry and behavioral observation. Direct questioning suffers from significant limitations, including social desirability bias where respondents provide socially acceptable rather than genuine answers, and the fact that approximately 95% of human decisions occur subconsciously as discussed in [25]. Observational methods capture the unconscious preferences that drive actual user behavior, providing more accurate insights into real-world usage patterns.

These methodological considerations explain our contradictory findings. While 87.5% of WinWin Meets participants believed AI could help achieve meeting goals, market research revealed minimal organic interest in AI-enhanced conferencing. This divergence reflects the difference between conscious evaluation in controlled environments versus unconscious behavioral preferences that emerge during natural usage. Additionally, our testing participants were primarily young AI researchers, likely more receptive to AI features than typical users.

Our research uncovered widespread "Zoom fatigue", indicating that users have reached cognitive saturation with current video conferencing complexity. The strong preference for meeting notes over real-time AI assistance (60% versus minimal interest) demonstrates users' desire for post-meeting value without additional in-meeting cognitive burden. This psychological context explains why solutions that prioritize seamless integration over feature prominence tend to gain market traction [26].

Our findings suggest distinct pathways for AI-enhanced video conferencing innovation. Industry-specific applications such as negotiations, sales, and legal consultations represent focused market segments where specialized AI features deliver measurable value propositions. The 68.8% preference for add-on integration over standalone applications indicates a market opportunity in enhancing existing platforms rather than replacing them, as demonstrated by successful tools like Fathom and Otter.ai. Although there is room for breakthrough products, any new solution must be at once reliable, easy to use, and meaningfully smarter than current tools—a difficult balance as existing platforms already invest heavily in their core features.

The emphasis on reliability and customizable AI assistance reveals that AI features must meet higher performance standards than traditional features. Users consistently prioritize dependable functionality over advanced capabilities, suggesting that product development should focus on perfecting core AI functions before expanding feature sets. Future research should examine longitudinal adoption patterns and explore how user acceptance evolves as AI capabilities mature and become more familiar in workplace contexts.

7 Acknowledgements

The research described in this paper was supported by the TWON project, funded by the European Union under Horizon Europe, grant agreement No 101095095.

References

[1] Rutger Rienks, Anton Nijholt, and Paulo Barthelmess. 2009. Pro-active meeting assistants: attention please! *Ai & Society*, 23, 2, 213–231.
[2] Moira McGregor and John C Tang. 2017. More to meetings: challenges in using speech-based technology to support meetings. In *Proceedings of the 2017 ACM conference on computer supported cooperative work and social computing*, 2208–2220.
[3] T3 Technology Hub. 2024. Market share of videoconferencing software worldwide in 2024, by program. Statista. Graph. (Apr. 2024). Retrieved Jan. 13, 2025 from https://www.statista.com/statistics/1331323/videoconferencing-market-share/.
[4] tldx Solutions GmbH. 2025. Tl;dv. https://tldv.io/. Accessed: August. (2025).
[5] Otter.ai, Inc. 2025. Otter.ai. https://otter.ai/. Accessed: August. (2025).
[6] 2025. Fathom. https://fathom.video/. Accessed: August. (2025).
[7] 2025. Fireflies. https://fireflies.ai/. Accessed: August. (2025).
[8] 2025. Sembly ai. https://www.sembly.ai/. Accessed: August. (2025).
[9] Zoom Video Communications. 2025. Zoom ai companion. https://www.zoom.com/en/ai-assistant/. Accessed: August. (2025).
[10] Everett M Rogers, Arvind Singhal, and Margaret M Quinlan. 2014. Diffusion of innovations. In *An integrated approach to communication theory and research*. Routledge, 432–448.
[11] 8x8, Inc. 2025. Jitsi. https://jitsi.org/. Accessed: August. (2025).
[12] Gloria Mark, Shamsi T. Iqbal, Mary Czerwinski, Paul Johns, and Akane Sano. 2016. Neurotics can't focus: an in situ study of online multitasking in the workplace. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, 1739–1744.
[13] Anysphere Inc. 2025. Cursor. https://cursor.sh/. Accessed: August. (2025).
[14] Anthropic. 2025. Claude 3.7 sonnet. https://www.anthropic.com/news/claude-3-7-sonnet. Accessed: August. (2025).
[15] OpenAI. 2025. Gpt-4o. https://openai.com/index/hello-gpt-4o/. Accessed: August. (2025).
[16] Meta Open Source. 2025. React. https://react.dev/. Version 19. Accessed: August. (2025).
[17] Sebastián Ramírez. 2025. Fastapi. https://fastapi.tiangolo.com/. Accessed: August. (2025).
[18] OpenAI. 2025. Whisper. https://openai.com/research/whisper. Accessed: August. (2025).
[19] NP Digital. 2025. Answer the public. https://answerthepublic.com/. Accessed: August. (2025).
[20] 2025. Answer socrates. https://answersocrates.com/. Accessed: August. (2025).
[21] Candour. 2025. Alsoasked. https://alsoasked.com/. Accessed: August. (2025).
[22] Neil Patel Digital. 2025. Ubersuggest. https://neilpatel.com/ubersuggest/. Accessed: August. (2025).
[23] xAI. 2025. Grok. https://grok.x.ai/. Accessed: August. (2025).
[24] 2025. Floth. https://floth.ai/. Accessed: August. (2025).
[25] Gerald Zaltman. 2003. *How Customers Think: Essential insights into the mind of the market*. Harvard Business Press.
[26] Fred D Davis. 1989. Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS quarterly*, 319–340.

Predicting Ski Jumps Using State-Space Model

Neca Camlek*
Univerza v Ljubljani
Ljubljana, Slovenia

Živa Hegler*
Univerza v Ljubljani
Ljubljana, Slovenia

Jakob Jelenčič
Jožef Stefan Institute
Ljubljana, Slovenia
jakob.jelencic@ijs.si

Marko Grobelnik
Jožef Stefan Institute
Ljubljana, Slovenia
marko.grobelnik@ijs.si

Dunja Mladenić
Jožef Stefan Institute
Ljubljana, Slovenia
dunja.mladenic@ijs.si

Abstract

Ski jumping performance is shaped by both athlete technique and environmental conditions, with factors such as wind speed, wind direction, and ski orientation playing a critical role in determining jump trajectories. Accurate modeling of these trajectories is challenging due to dynamic and time-dependent nature of the system. In this work, we introduce a dataset of measured ski jumps and present a state-space modeling framework that captures the evolution of jumps under varying conditions. The model parameters are estimated using a ridge regression approach, enabling us to predict trajectories from initial states and wind sensor inputs. We evaluated the predictive performance of the model through leave-one-out cross-validation and analyzed its stability, showing that the approach can generate realistic trajectories with reasonable accuracy. To complement the modeling results, we developed an interactive web application that allows users to explore both recorded and simulated jumps, adjust environmental factors, and visualize their effects through animations. Together, the dataset, modeling framework, and the application offer a foundation for further research in ski jump analysis and provide an accessible tool for exploring the influence of external conditions on performance.

Keywords

datasets, state-space model, ski jumping, simulations, least squares

1 Introduction

Ski jumping is a sport strongly influenced by both athletic technique and environmental conditions. Factors such as wind speed, wind direction, and different ski angles affect the trajectory and final distance of a jump, making accurate prediction a challenging problem. While statistical models and simulations have been applied in sports research for some time, many approaches simplify the problem and do not fully capture the dynamic evolution of the jump over time [11].

Recent advances in machine learning have introduced methods capable of modeling temporal systems with greater fidelity. In particular, state-space models provide a mathematical framework for representing hidden internal states that evolve over time in response to external input. This makes them well-suited for modeling ski jumps, where environmental factors determine performance [9].

In this paper, we present a ski jump dataset together with a state-space model trained to predict jump trajectories based on changing environmental conditions. The model is estimated using a least squares approach and demonstrates how inputs such as wind and ramp adjustments influence the resulting jump. Beyond the modeling framework, we also developed an application that allows general users to interact with the data, run simulations, and visualize jump trajectories through animations.

Beyond methodological interest, accurate prediction of ski jumps can improve athlete safety by anticipating risky conditions, support planning of hill design or enlargement, and contribute to fairer competitions through a better understanding of environmental effects.

The remainder of the paper is as follows. Section 2 presents the handling of received data. Next, the proposed methodology is described in Section 3. The project results are presented in Section 4. We discuss the results in Section 5 and conclude the paper in Section 6.

2 Modeling Framework and Dataset

This section describes the handling of data, focusing on state-space models and our data processing.

2.1 State-Space Model

State-Space Models (SSMs) are a family of machine learning algorithms designed to capture and predict the behavior of dynamic systems by describing how their inner states change over time. Instead of only looking at past inputs and outputs, SSMs explicitly model the underlying dynamics, making them well-suited for sequential data. In state-space modeling, the objective is to identify the minimal set of system variables required to completely describe the system. These fundamental variables are referred to as the state variables. At any given time, the state of the system can be represented by a state vector, whose components correspond to the values of the respective state variables. SSMs are designed to predict both the manner in which inputs are reflected in the system's outputs and the evolution of a system's internal state over time and in response to specific inputs [2].

2.2 Least squares method

The least squares method is a regression technique that is used to determine the line that best fits a given set of data. It minimizes the sum of the squared differences between the observed data and the corresponding values implied by the regression function. Each data point reflects the relationship between a known independent variable and an unknown dependent variable [7].

*Both authors contributed equally to this research.

Information Society 2025, Ljubljana, Slovenia

https://doi.org/10.70314/is.2025.sikdd.30

To enhance the model, we incorporated ridge regression (L2 regularization), which helps to reduce overfitting during model training [12].

2.3 Data Processing

For our project, we used 223 CSV files, each containing the data of a jump, measured on the flying hill of Gorišek brothers in Planica, Slovenia. Each contains 17 columns (`'Position'`, `'Height above ground'`, `'Time'`, `'X'`, `'Y'`, `'Z'`, `'Opening Angle'`, `'Stalling Angle Left'`, `'Stalling Angle Right'`, `'Roll Angle Left'`, `'Roll Angle Right'`, `'Yaw Angle Left'`, `'Yaw Angle Right'`, `'Speed hor.'`, `'Speed vert.'`, `'Speed resulting'`, `'WindTime|WindName|WindSpeed|Wind...'`) and the number of rows corresponding to the length of the jump. Data are recorded for every meter of air distance from the take-off point.

The data required some pre-processing before it could be used for training the model.
The column `WindTime|WindName|WindSpeed|...` combined multiple attributes separated by '|'. Data from 12 sensors, each measuring six wind characteristics, were expanded into $12 \times 6 = 72$ columns, one per sensor–feature pair (`sensor_feature`).

Position - air distance from the take-off point in meters. Begins with a negative value, which represents the distance from the starting point to the take-off point. In ski jumping, the starting point is adjusted according to the wind conditions, so this value is not constant.
Height above ground - height above ground in meters.
Time - time of the jump in seconds from the start of the jump.
X, Y, Z - coordinates of the jumper in a 3D space in meters. The X axis is aligned with the hill direction, the Y axis is across the hill, and the Z axis is vertical. The take-off point is $(0, 0, 0)$ as shown in Figure 1
Opening Angle - angle between the skis in degrees.
Stalling Angle Left, Stalling Angle Right - angle between the chord line of the left/right ski and the horizontal plane in degrees.
Roll Angle Left, Roll Angle Right - angle of the left/right ski around its longitudinal axis relative to the horizontal plane in degrees.
Yaw Angle Left, Yaw Angle Right - angle between the left/right ski and the horizontal plane in degrees. (angles are shown in Figure 2)
Speed hor., Speed vert., Speed resulting - horizontal, vertical, and the resulting speed of the athlete in km/h [13].

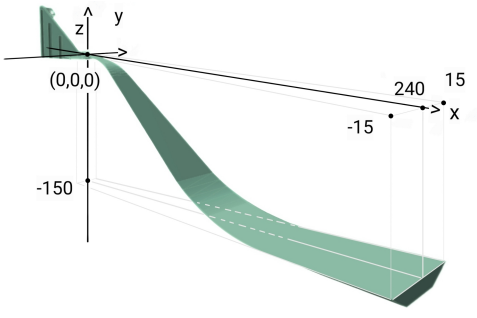


Figure 1: 3D model of Ski jump in Planica with added coordinates [10, 1]

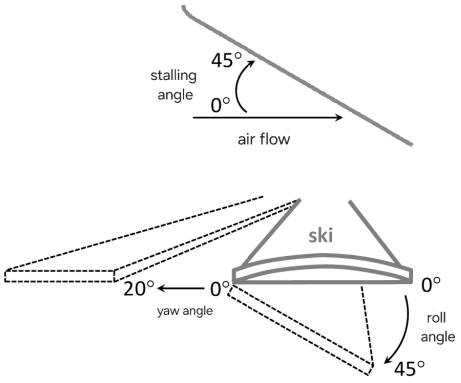


Figure 2: Different angles affecting the jump

The wind features are as follows:

WindTime - time of the wind measurement in the same format as the Time column itself. Since wind measurements are recorded less often, the wind values are applied to the most recent jump measurement and then just repeated until a new wind measurement is available. Since the wind is represented by a nonlinear function, it would be hard to capture its movements with interpolation, so we decided to drop this column.
WindName - name of the sensor (Wi for $i = 1, \ldots, 12$)
WindSpeed - resulting speed of the wind in km/h
WindSpeedTangent - speed of the wind tangent measured along the x axis (hill direction) in km/h
WindTurbulence - vertical speed of the wind turbulence in km/h
WindSpeedCleanTan - wind speed tangent with turbulence removed in km/h
WindSpeedCross - speed of the wind measurement along the y axis across the hill in km/h

There are 12 wind sensors spread across the ski jump hill. To help with the analysis, we separated the jump section of the hill into 3 zones. The first zone contains wind sensors 1 to 4, the second zone contains sensors 5 to 8, and the third zone contains sensors 9 to 12 [11].

During processing, we also removed some ski jumps that were incomplete or had corrupted data, so the final dataset contained around 200 ski jumps.

3 Methodology

This section describes our research methodology. We first present different variations of the SSM that we tested for the ski jump simulation, followed by describing our model and how it predicts the jumps. Finally, we present the description of our ski jump animation app.

3.1 Different modeling approaches

In addition to pure SSM, we considered different approaches for modeling ski jumps that included classical physics-based models, but the data are not sufficient to accurately capture all the forces acting on the jumper. We also tried a hybrid approach that combined SSM and Physics-informed Neural Networks (PINNs [14]), where the SSM would provide a baseline prediction and the PINN would learn to correct any discrepancies, taking into account physical properties of the system, such as the mass of the pilot, the properties of the wind, and gravitational force [4].

These parameters are included in the equations of motion and added to the total loss function. So, the model prefers solutions that are consistent with the laws of physics. This turned out to be less effective than a pure SSM approach, but the reason exceeds the purpose of this paper. More about errors and models' comparison is given in Section 4.1.

3.2 Ski jump prediction model

In order to fit our data to the SSM, we stored the data in each file in three vectors. The main vector contains states or state variables of the system, which in our case are the X, Y, and Z coordinates, jumper velocities, and all angles (opening, stalling, roll, and yaw) [6].

The observation vector contains the measured outputs of the system, which in our case are the X, Y, and Z coordinates and height above ground. The controls contain the external inputs to the system, which in our case are the wind measurements from all the sensors that are averaged over each zone and feature (speed, tangent, cross and turbulence).

We then used ridge regression to estimate the matrices A, B, C and D of the SSM, as shown in Figure 3, where we minimized the computed values from the current and previous values and the next time-stamped values. Thus, matrix A computes the next state from the current state, B computes the next state from the current control, C computes the next observation from the current state, and D computes the next observation from the current control. We then use recursion to predict the next state from the prediction of the previous state and the current control, to get the full simulated jump. This allows us to predict the jump trajectory based on the environmental conditions and the starting state of the jumper [9].

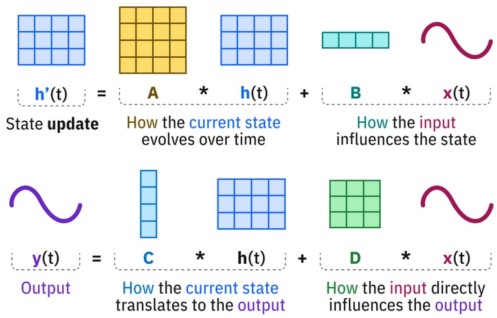


Figure 3: Schema of SSM matrices [3]

3.3 Ski jump animation app

To make our results accessible beyond the research setting, we developed an interactive web application using Shiny for Python [8]. The application serves as a front-end to the trained state-space model and allows users to explore ski jump simulations under varying environmental conditions or just to observe different measured ski jumps. Firstly, through a set of input controls, users can adjust factors such as wind speed, wind directions, or different ski angles, and the application instantly updates the predicted jump trajectory. Secondly, users can simply explore random jumps from the provided dataset or upload their own CSV file of measured jumps, as long as it includes the columns described in Section 2.3.

The application presents the results as an animated visualization of the ski jump, showing the full trajectory and the final distance. In this way, the application functions both as an analytical tool, helping to test how different conditions affect performance, and as an educational resource that makes the mechanics of ski jumping easier to understand for a wider audience. It is available online. [1]

4 Main results

In this section, we present the results of our simulations. Firstly, we present a statistical comparison of all the models, followed by a precise analysis of our predictions.

4.1 Models' error

In order to evaluate different models, we first had to define a metric to measure the prediction error. Since actual and simulated jumps are represented with x, y, and z coordinates but are measured at different time stamps and can contain a different number of measurements, we had to find a way to compare them. We first tried to project the shorter trajectory on to the other one and compute the distance between the original and the projection, but this method turned out to be computationally expensive. So we decided to compute the distance between the actual and the simulated jumps by interpolating both jumps. The new measurements contain the start and end point and all the ones, where x reaches a natural value. We then compute the error as the norm of the difference between the two jumps. And after one of the jumps ends, we just add the distance from the end of the shorter jump to the end of the longer jump to the error. In this way, we penalize the model for not being able to predict the correct length of the jump.

Since we had a limited number of jumps, we used leave-one-out cross-validation to evaluate the models. For each jump, we trained the model on all other jumps and then simulated the left-out jump. We then calculated the average error between the actual jump and the simulated jump for both the training set and the test set, as shown in Figure 4.

In the process of developing our ski jump prediction model, we evaluated several variations to determine the most effective approach. We compared the performance of a pure SSM with a hybrid model that combined SSM with PINN. The pure SSM demonstrated superior predictive accuracy, probably due to its ability to directly model the temporal dynamics of ski jumps without the added complexity of PINNs. We also experimented with different configurations of the SSM, including using all available wind sensor data versus an averaged value of the zone. When we used all sensors, the average error for each point (in the training data is 1.67 m and in the test data is 1.89 m), while when we averaged the sensors over the zones, the error (in the training data was 1.76 m and in the test data 1.82 m). This suggests that averaging the wind data helps with the simulation.

4.2 Analysis of our model

Wind is a critical factor in ski jumping, so we attempted to capture its nonlinear effects by including columns for the squared wind features. However, we found that adding these squared terms did not significantly reduce the prediction error.

Since the simulation still requires numerous inputs, we made it interactive, allowing users to adjust the wind conditions and observe their impact on the jump. In the ski jumping app, users

[1] https://camlekn.shinyapps.io/ski-jump/

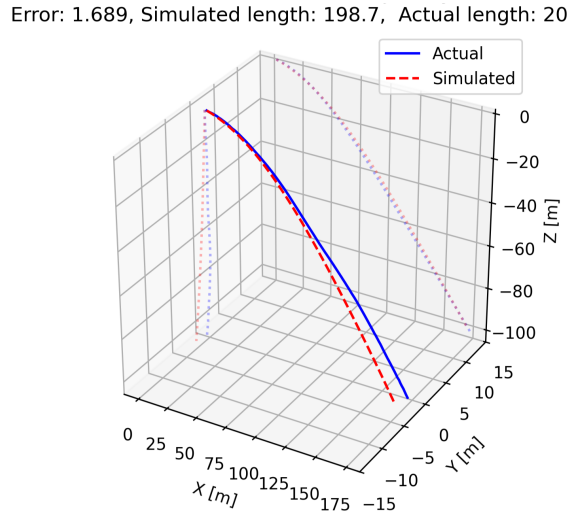


Figure 4: Actual vs. simulated ski jump trajectory

can manipulate sliders to set the wind speed, wind tangent, wind cross, and wind turbulence for each of the three zones. As a result, the wind loses its original movement function in the simulation. All other inputs are set to the average values computed from the dataset.

5 Discussion

This section examines the predictive performance of the trajectories, highlights the limitations of our current approach, and suggests directions for future improvements.

5.1 Limitations

Given the relatively small dataset of ski jumps, the main limitation of our project lies in the limited data available for training the model. After preprocessing, the dataset contained only about 200 jumps, which may limit the SSM's ability to represent the full range of trajectory variations under different circumstances. As a result, the model may struggle to accurately predict jumps under novel or extreme conditions.

Furthermore, the dataset lacks detailed information, or any information at all, about individual jumpers, such as body weight, sex, or other physiological characteristics that are known to influence jump performance. Incorporating these variables could improve model accuracy and provide more personalized predictions [5].

Lastly, due to limited computing power, only one CPU was available, restricting the use of possible better models. To address these challenges, using cloud-based resources could help run larger models and improve the prediction of trajectories.

5.2 Future work and potential improvements

Although the current approach shows promise, there are several avenues for future improvements. Some of which we are working on at the time of writing this paper.

Currently, we are working on improving the sliders' functions. Since the wind data determined by the user is static throughout the jump, this adds a lot of generalization. In reality, wind conditions can change rapidly during a jump. So we would like to add additional controls to the app that would allow the user to define how the wind changes during the jump. They could choose whether the wind would gain or lose a certain feature (such as speed or turbulence) during the jump.

Expanding the dataset to include more jumps and additional contextual information about individual jumpers could improve the accuracy of the model. We could try to generate more data by using data augmentation techniques, such as adding noise to the wind measurements or slightly modifying the angles. We could also try to find the nonlinear movements of the wind and interpolating the wind measurements by their original time stamps to better capture the wind dynamics.

6 Conclusion

This paper presents a method for predicting ski jump trajectories based on environmental conditions. By incorporating external factors into the modeling framework and applying least squares estimation, we demonstrated that the model is capable of capturing the dynamics of ski jumps and producing realistic trajectory predictions. In addition, we developed an interactive application that makes the results accessible to a broader audience through simulations and animations of predicted jumps. Although the current model is limited by the size of the dataset and the absence of certain athlete-specific variables, the results show that state-space models are a promising tool for analyzing ski jumping performance.

7 Acknowledgments

This work was supported by Smučarska Zveza Slovenije (Ski Association of Slovenia), whom we would like to thank for providing the ski jump data.

References

[1] 3DWarehouse via 3dmdb.com. 2025. "ski jumping planica" [3d model]. https://3dmdb.com/en/3d-model/ski-jumping-planica/8386000/?free=True&q=Ski+jump. Free model; accessed: 2025-08-26. (2025).

[2] Masanao Aoki. 1990. *State Space Modeling of Time Series*. (2nd, revised and enlarged ed.). *Universitext*. Springer Berlin Heidelberg, Berlin, Heidelberg. ISBN: 978-3-642-75883-6. doi:10.1007/978-3-642-75883-6.

[3] Dave Bergmann. 2025. What is a state space model? Accessed: 2025-09-24. https://www.ibm.com/think/topics/state-space-model.

[4] Shengze Cai, Zhiping Mao, Zhicheng Wang, Minglang Yin, and George E. Karniadakis. 2021. Physics-informed neural networks (pinns) for fluid mechanics: a review. *Acta Mechanica Sinica*, 37, 12, 1727–1738. doi:10.1007/s10409-021-01148-1.

[5] Wolfram Müller. 2008. Performance factors in ski jumping. In *Sport Aerodynamics*. CISM International Centre for Mechanical Sciences. Vol. 506. Helge Nørstrud, editor. Online ISBN: 978-3-211-89297-8. Springer, Vienna, 139–160. ISBN: 978-3-211-89296-1. doi:10.1007/978-3-211-89297-8_8.

[6] Wolfram Müller. 2006. The physics of ski jumping. Tech. rep. CERN report on the aerodynamics and physics of ski jumping. CERN. https://cds.cern.ch/record/1009275/files/p269.pdf.

[7] Bor Plestenjak. 2016. *Razširjen uvod v numerične metode*. Slovenian textbook on numerical methods. DMFA-založništvo.

[8] Posit Team. 2025. Shiny for python. Accessed: 2025-08-29. https://shiny.posit.co/py/.

[9] Serrano.Academy. 2025. State-space model (ssm) tutorial. https://youtu.be/g1AqUhP00Do. State-Space Model (SSM) video. (2025).

[10] Ski Jumping Hill Archive, skisprungschanzen.com. 2025. Letalnica (letalnica bratov gorišek), planica, slovenia — ski jumping hill archive. https://www.skisprungschanzen.com/EN/Ski+Jumps/SLO-Slovenia/Planica/0475-Letalnica/. Accessed: 2025-09-12. (2025).

[11] Ava Thompson, ed. 2025. *Ski Jumping*. Found via Google Books at https://www.google.si/books/edition/Ski_Jumping/G2pPEQAAQBAJ?hl=en&gbpv=0. Publifye AS.

[12] Wessel N. van Wieringen. 2015. Lecture notes on ridge regression. arXiv preprint arXiv:1509.09169. Revision v8, submitted 30 September 2015; revised 27 June 2023. (2015). doi:10.48550/arXiv.1509.09169.

[13] Mikko Virmavirta and Juha Kivekäs. 2019. Aerodynamics of an isolated ski jumping ski. *Sports Engineering*, 22, 1, 1–6. doi:10.1007/s12283-019-0298-1.

[14] StatQuest with Josh Starmer. 2025. Neural networks tutorial. https://youtu.be/CqOfi41LfDw. Neural networks introduction video. (2025).

Predicting milling overload based on sensor data: a graph-based approach

Roy Krumpak
Jožef Stefan Institute
Ljubljana, Slovenia
krumpak.roy@gmail.com

Jože M. Rožanec
Jožef Stefan Institute
Ljubljana, Slovenia
joze.rozanec@ijs.si

Dunja Mladenić
Jožef Stefan Institute
Ljubljana, Slovenia
dunja.mladenic@ijs.si

Zhenyu Guo
BGRIMM Technology Group
Beijing, China
guozhenyu@bgrimm.com

Tao Song
BGRIMM Technology Group
Beijing, China
songtao@bgrimm.com

Dumitru Roman
SINTEF Digital
Oslo, Norway
titi.roman@sintef.no

Inna Novalija
Jožef Stefan Institute
Ljubljana, Slovenia
inna.koval@ijs.si

Xiang Ma
SINTEF Industry
Oslo, Norway
xiang.ma@sintef.no

ABSTRACT

In this paper, we present an approach to predict milling overload that leverages time series-to-graph transformations, which, along with domain data encoded as a graph, are fed to predictive machine learning models. Additionally, we compared the performance of the graph-based approach with the TS2Vec foundational model, regarded as the State-Of-The-Art. Our results show that TS2Vec performed best across all time windows. While combining TS2Vec and graph embeddings resulted in reduced performance compared to TS2Vec, it enhanced the outcomes when compared to the sole use of graph embeddings. Furthermore, combining Ordinal Partition Graph and TS2Vec embeddings resulted in more stable performance across predictive time windows.

KEYWORDS

Time series, graphs, mining, milling, predictive maintenance, sensor data

1 INTRODUCTION

Milling, central to mineral processing, involves breaking down ores into smaller particles, but is prone to abnormal behavior due to material properties and upstream steps (Hodouin et al. 2001 [3]; Galán et al. 2002 [2]). While traditional control relied on operators, advances in machine learning (ML) have enabled data-driven optimization and predictive maintenance (Mobley 2002 [6]). Graph-based methods are increasingly applied to time series to capture temporal and structural relations (Silva 2021 [8]). Variants include Natural Visibility Graphs (NVG) to capture the time series topology (Lacasa et al. 2008 [4]; Stephen et al. 2015 [10]), Quantile Graphs for time series values' transitions (Silva et al. 2024 [9]), and Ordinal Partition Graphs to capture regular temporal patterns and their transitions.

The contributions of this paper include the use of multiple graph representations (not just one) to capture the structure of a time series and evaluation of the described approach on a real-world dataset.

2 USE-CASE DESCRIPTION

BGRIMM Technology Group is a Chinese leader in mining and mineral processing solutions, focusing on automation and intelligent control, with grinding optimization as a core area. Grinding is both the most energy-intensive step in mineral processing—accounting for 40% of total energy costs—and a key determinant of downstream recovery and product quality (Zhou et al. 2009 [11]; Lessard et al. 2016 [5]; Groenewald et al. 2006 [1]). At a 10,000 ton/day copper plant in Anhui Province using a SAG–ball–pebble (SABC) circuit, BGRIMM is developing intelligent control strategies to maximize throughput while preventing SAG mill overload. Central to this effort is accurate SAG power prediction, which serves as a feedforward signal to improve feed regulation and overall process efficiency.

3 DATASET

The dataset used in this article was collected and provided by BGRIMM Technology Group. The data consists of various sensor measurements from the machines used in their mine's ore processing plant, accounting for a total of 42 columns. One column stores the date and time of the measurement, while the rest contain numerical values. The sensor data was sampled every two seconds and compiled across a hundred days from January 1^{st} 2019, to April 12^{th} 2019, excluding the first two days of April, resulting in 4.32 million rows in the data. Besides the raw data, a description of an overload state was also provided. A column named `SAG_2201.power`, which represents the power of the SAG mill, is used to decide whether there is an anomaly in the data. If the column reaches a value above 4700 [kW] and has an upward trend or whenever it surpasses the value of 4800, this is considered an overload of the system, and a supervisor might take appropriate actions to stop the overloading.

Jože M. Rožanec and Roy Krumpak are co-first authors with equal contribution and importance.
Corresponding author: Jože M. Rožanec: joze.rozanec@ijs.si.

Information Society 2025, 6–10 October 2025, Ljubljana, Slovenia

https://doi.org/10.70314/is.2025.sikdd.21

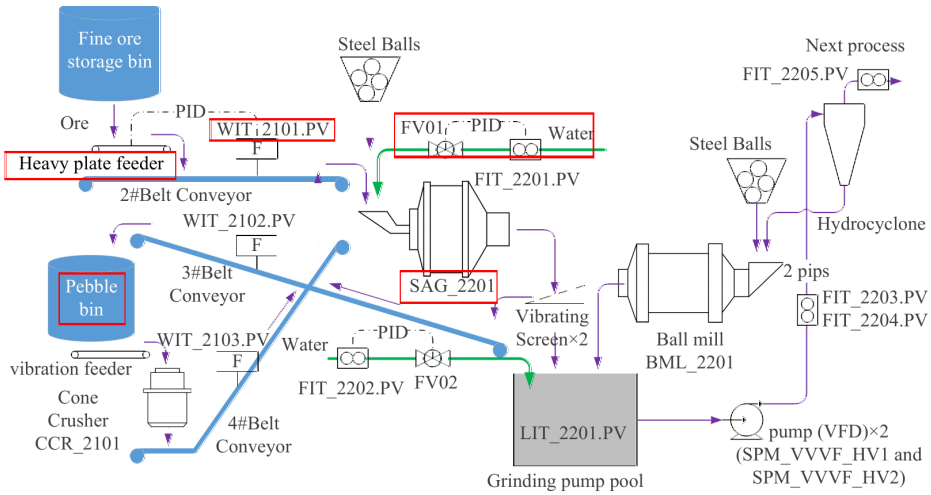


Figure 1: The diagram depicts the milling plant components and how they are connected. The components of interest are highlighted with red rectangles.

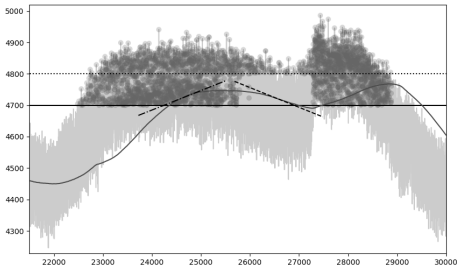


Figure 2: `SAG_2201.power` column (light gray), where anomalies (gray dots) are annotated based on the moving average (gray), the automatic anomaly label threshold (dotted black), the possible anomaly label threshold (solid black), and linear regression slope (positive - dashed and dotted, negative - dashed).

4 METHODOLOGY

4.1 Data preparation

Based on experts' input, the samples with `SAG_2201.power` < 4700 were labeled 0 (no anomalous event), others with 1 (milling overload). A 1-hour (1800-sample) moving average with linear regression checked for upward trends; if none, the label was reset to 0 (see Fig. 2). Next, we selected a subset of columns to be used in the analysis, utilizing expert knowledge to choose only those columns that are measured in the workflow before `SAG_2201.power` column. The resulting columns are `LIT_2103A.PV`, `FCV_2201.PID_SP`, `SAG_2201.Press_Ziyouduangaoya2`, `Feeder_Control.SP`, `SAG_2201.power` and `WIT_2101.PV`.

4.2 Feature engineering

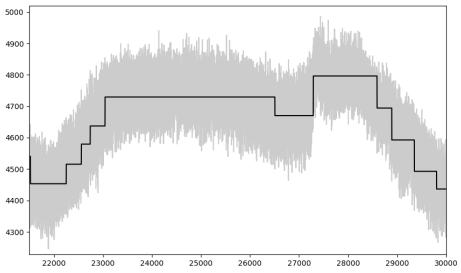
The raw data from the selected columns was first checked for any missing values, which were not present. In the next step, we detected changes in the columns and then replaced the values in the samples between two such changes with the mean value of that segment (see Fig. 3a). This data was further simplified with the help of a k-bins discretizer, which was used to encode each column with seven values based on the quantile into which each sample fell (see Fig. 3b). The column named `WIT_2101.PV` was excluded from the first step of data simplification and graph representations and was processed separately because its values did not appear to have distinct oscillating levels and did not benefit from such processing. After discretization, every column had an integer value between zero and six, and with each row being then interpreted as a state. The average state duration is 42 seconds. Repeated states (duplicate rows) were dropped, decreasing the size of the dataset (see Fig. 3c). For a visual representation of these steps, see Fig. 3, where the data from one picture is used, and, where important, also noted in the next one. The data here include raw data in Fig. 3a, the 'means' data in Fig. 3a and Fig. 3b, simplified data in Fig. 3b, and unique sample data in Fig. 3c. The annotated plot in Fig. 3c is used as the base data for an example NVG generation in Fig. 4. The numbers represent the same data point, one in the plot and one in the graph representation.

4.3 Modeling the data as graphs

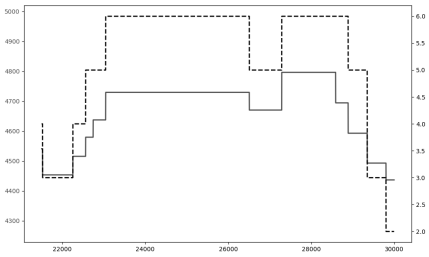
We employ three strategies for converting time series into graphs: Natural Visibility Graphs (NVG), Ordinal Partition Graphs (OPG), and Quantile Graphs (QG). We used the time series to graph and back library[1] to achieve this.

For each sample in the data, we built a graph representation of it by looking at the samples within a selected window w_s preceding it and applying the described time series to graph strategies on each column, apart from `WIT_2101.PV`, separately. Such graphs, called subgraphs, were bound to a default graph structure that presents which columns are neighboring in the plant process (see Fig. 1) by connecting a node which represents the `SAG_2201.power` column to every other column. The result of this step was a larger type of graph called a state graph (see Fig. 5). The black nodes represent nodes for a particular column, while gray nodes represent the subgraphs created from the time series. The subgraphs are connected to the column nodes via the node that corresponds to the first instance from the timeseries. Depending on the experiment, we made an additional step of joining w_0 many of the state graphs into a larger graph, which was used to generate embeddings.

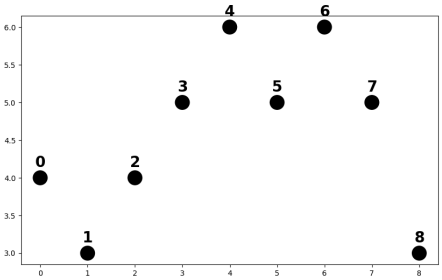
[1] https://timeseriestographs.com/



(a) `SAG_2201.power` column (light gray), where a threshold change detection was used to detect changes and to replace in-between values with the mean value (black).



(b) Result (dashed black) of applying a k-bins discretizer model on the previously simplified data (solid black) from Fig. 3a. Note the different y-axis scales of the overlaid graphs.



(c) A representation of the simplified column data from Fig. 3b, considering only the unique consecutive values.

Figure 3: Pipeline of transformations on the `SAG_2201.power` column.

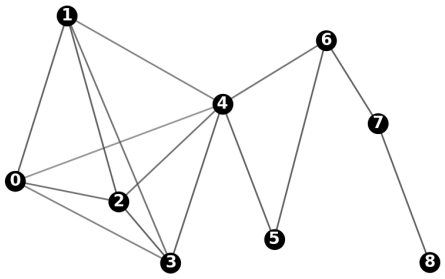


Figure 4: The Natural Visibility Graph representation of the data in Fig. 3c.

A Graph2Vec model from the karateclub library [7] was used to generate graph embeddings, with an embedding size of 250.

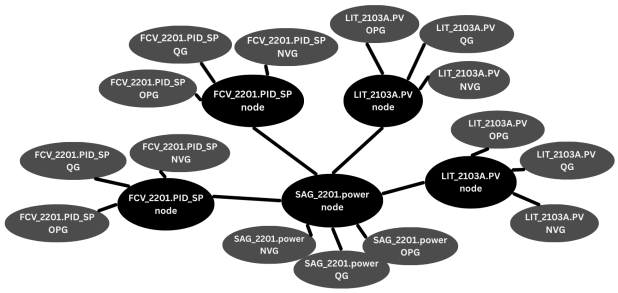


Figure 5: Example of a state graph.

We chose this model for its ease of use and performance reasons. Column `WIT_2101.PV` was also transformed into an embedding form by using a TS2Vec model[2]. The embedding output size was set to 40, as this is approximately the size of features proportional to the number of columns in the graph embeddings.

4.4 Model training and evaluation

An initial subset of the data, which included the data from the first available day, was used to test the performance of different graph embeddings. This was done to reduce the time and memory consumption for the first assessment. A CatBoost model was used, where it was trained for 800 iterations, with a learning rate equal to 0.03 and the Cross Entropy loss function, as well as the leaf regularization parameter set to 0.3. To assess our model's ability to predict anomalous states, we also tried to fit the model on the same data, but with the target column shifted accordingly. This was done for up to 90 shifts, which is equivalent to predicting 63 minutes in advance. When we selected the best graph embeddings, we built and tested the model on the entire data set.

5 EXPERIMENTS

We conducted three experiments, all of which follow the same template, where we tested how the structure of a graph affects the end model's ability to predict anomalies. This includes first creating subgraphs as NVG, OPG and QG representations of the columns with window size w_s and joining them into the state graph representation (see Fig. 5). Finally, w_0 many of these state graphs are joined sequentially according to the order given by the time at which the represented states appear in the data. The experiments differ in the window sizes w_s and w_0. Experiment A used $w_s = 50, w_0 = 1$, Experiment B used $w_s = 15, w_0 = 20$, lastly Experiment C used $w_s = 15, w_0 = 40$. If we take the average state duration of 42 seconds into account, we see that in Experiment A, data from the last 35 minutes is used, in Experiment B, 15 minutes, and finally in Experiment C, 28 minutes.

We carried out experiments similar to Experiment B, where the state graphs were structured based only on one specific type of subgraph. Furthermore, the impact of the separately processed `WIT_2101.PV` was also tested, by repeating the same experiments, with the difference being that this column's embeddings were excluded when training the final model. These experiments do not have a mark in the 'WIT' column of the resultst Table 3.

[2]https://github.com/zhihanyue/ts2vec

Time to predict[min]				
7	21	35	49	63
0.9905	0.9528	0.8929	0.8235	0.7623

Table 1: ROC AUC results of the experiment where all data was embedded with TS2Vec models.

	Time to predict[min]				
Experiment	7	21	35	49	63
A	0.6083	0.5763	0.5356	0.5333	0.4945
B	**0.6943**	**0.6698**	**0.6364**	0.6184	**0.6128**
C	0.5897	0.5688	0.6109	**0.6417**	0.5910

Table 2: ROC AUC results of the three experiments with respect to how far ahead the model is predicting. The best results are marked in bold text.

Lastly, a separate experiment was carried out, in which all raw data were processed using the TS2Vec model. Each column had its own TS2Vec model, which was used to embed the data associated with that column. Then, a CatBoost model with the same configuration as in the previous experiments was used in combination with TS2Vec joined embeddings to predict the anomalies. These results are gathered in Table 1.

6 RESULTS

The results of the three experiments, which tested the informativeness of the graph structure, as well as the experiments designed to determine which type of data is the most predictable, are summarized in the following tables.

As can be seen in Table 2, Experiments A and C have lower scores than Experiment B. However, Experiment C approaches the performance of Experiment B at the maximum predicting shift. For this reason, and because the types of graphs in Experiment B are smaller compared to those in Experiment C, the experiments that tested the impact of different types of data used Experiment B-type graphs. The best results for the final model were obtained from the data, where all columns were embedded using TS2Vec models, as shown in Table 1. Similarly, the results in table 3 show that when we predict anomalies from only the TS2Vec embeddings of the column `WIT_2101.PV`, the performance is the best.

Additionally, if we compare the experiments with `WIT_2101.PV` embeddings to the ones without them, we can see that the latter perform worse. This suggests that the TS2Vec embeddings are more informative than the graph embeddings. Nevertheless, when comparing different types of graphs used in the final graph, we can see that OPGs alone yield the best performance.

A few possible explanations for the difference in performance between the graph-based and time series-based approaches are possible. First, when working with graphs, there are more parameters that need to be optimized, such as window sizes and parameters for constructing graphs from time series. Another reason might be that NVGs have approximately thirty times more edges and eight times more nodes compared to OPGs and QGs, which makes them disproportionately large. Additionally, the construction of state graphs has repeated structures, which is inefficient. Lastly, the TS2Vec embeddings do not have these limitations, and embeddings can be made from the entirety of the data, as opposed to the simplified ones when not using TS2Vec.

type of data used				Time to predict[min]				
NVG	OPG	QG	WIT	7	21	35	49	63
✓	✓	✓	✓	0.6558	0.6418	0.6251	0.6402	0.6184
✓	✓	✓		0.5938	0.6257	0.5831	0.5882	0.5725
	✓		✓	0.7427	<u>0.7146</u>	<u>0.6930</u>	<u>0.6853</u>	<u>0.6719</u>
		✓	✓	0.7265	0.6959	0.6586	0.6502	0.6365
	✓			<u>0.7452</u>	0.6978	0.6838	0.6734	0.6578
		✓		0.7219	0.6866	0.6643	0.6416	0.6096
			✓	**0.9292**	**0.9025**	**0.8893**	**0.8004**	**0.7042**

Table 3: ROC AUC results of the models trained on different types of graphs and data for Experiment B across all days. The best results are written in bold text, while the second best are underlined.

7 CONCLUSIONS

In this paper, we discuss the use of graph-based time series representations for training machine learning models. Our experiments suggest that while this approach has potential, it did not outperform the TS2Vec foundational model and was unable to yield superior results when combined with it. Future work will explore alternative graph representations and utilize GNNs to integrate topological, semantic, and time series information directly into a single machine learning model, aiming to achieve superior results.

ACKNOWLEDGEMENTS

The Slovenian Research Agency supported this work. It was also developed as part of the Graph-Massivizer project (grant agreement No. 101093202), the enRichMyData project (grant agreement No. 101070284), and the DataPACT project (grant agreement No. 101189771), all funded by the Horizon Europe research and innovation program of the European Union.

REFERENCES

[1] J.W. de V. Groenewald, L.P. Coetzer, and C. Aldrich. 2006. Statistical monitoring of a grinding circuit: an industrial case study. *Minerals Engineering*, 19, 11, 1138–1148. DOI: 10.1016/j.mineng.2006.05.009.

[2] O. Galán, G.W. Barton, and J.A. Romagnoli. 2002. Robust control of a sag mill. *Powder Technology*, 124, 3, 264–271. DOI: 10.1016/S0032-5910(02)00021-9.

[3] D. Hodouin, S.-L Jämsä-Jounela, M.T. Carvalho, and L. Bergh. 2001. State of the art and challenges in mineral processing control. *Control Engineering Practice*, 9, 9, 995–1005. DOI: 10.1016/S0967-0661(01)00088-0.

[4] L. Lacasa, B. Luque, F. Ballesteros, J Luque, and J.C. Nuño. 2008. From time series to complex networks: the visibility graph. *Proceedings of the National Academy of Sciences*, 105, 13, 4972–4975. DOI: 10.1073/pnas.0709247105.

[5] J. Lessard, W. Sweetser, K. Bartram, J. Figueroa, and L. McHugh. 2016. Bridging the gap: understanding the economic impact of ore sorting on a mineral processing circuit. *Minerals Engineering*, 91, 5, 92–99. DOI: 10.1016/j.mineng.2015.08.019.

[6] R. Keith Mobley. 2002. 4 - benefits of predictive maintenance. In *An Introduction to Predictive Maintenance (Second Edition)*. Plant Engineering. (Second Edition ed.). R. Keith Mobley, editor. Butterworth-Heinemann, Burlington, 60–73. ISBN: 978-0-7506-7531-4. DOI: 10.1016/B978-075067531-4/50004-X.

[7] B. Rozemberczki, O. Kiss, and R. Sarkar. 2020. Karate Club: An API Oriented Open-source Python Framework for Unsupervised Learning on Graphs. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM '20)*. ACM, 3125–3132. DOI: 10.1145/3340531.3412757.

[8] V.F. Silva, M.E. Silva, P. Ribeiro, and F. Silva. 2021. Time series analysis via network science: concepts and algorithms. *WIREs Data Mining and Knowledge Discovery*, 11, 3, 1–39. DOI: 10.1002/widm.1404.

[9] V.F. Silva, M.E. Silva, and P. Ribeiroand F. Silva. 2024. Multilayer quantile graph for multivariate time series analysis and dimensionality reduction. *International Journal of Data Science and Analytics*, 1–13. DOI: 10.1007/s41060-024-00561-6.

[10] M. Stephen, C. Gu, and H. Yang. 2015. Visibility graph based time series analysis. *PloS one*, 10, 11, e0143015. DOI: 10.1371/journal.pone.0143015.

[11] P. Zhou, T. Chai, and H. Wang. 2009. Intelligent optimal-setting control for grinding circuits of mineral processing process. *IEEE Transactions on Automation Science and Engineering*, 6, 4, 730–743. DOI: 10.1109/TASE.2008.2011562.

Short and Long Term Bike Rental Forecasting

Oskar Kocjančič*
oskar.kocjancic@gmail.com
Jožef Stefan Institute
Ljubljana, Slovenia

Martin Žnidaršič
martin.znidarsic@ijs.si
Jožef Stefan Institute
Ljubljana, Slovenia

Abstract

This paper describes the challenges and outcomes of forecasting bike rentals in a Slovenian urban bike-sharing system, focusing on the impact of data sparsity and the inclusion of external variables. We address two distinct forecasting tasks: short-horizon, one-day-ahead predictions for individual rental stations, and long-horizon, 90-day forecasts for the total rental volume. Various machine learning models were employed and evaluated in this context. We also analyzed the trade-off between using longer historical data versus shorter, weather-enriched data to improve predictive accuracy. The findings indicate a clear correlation between data sparsity at the station level and predictive performance. While the inclusion of weather data provides a modest improvement for both short-horizon and long-horizon forecasts, the overall quality of the sparse and noisy data appears to limit the potential gains from more complex modeling approaches.

Keywords

bike-sharing, forecasting, time series, data sparsity, machine learning, deep learning, weather data

1 Introduction

Predicting rental patterns of urban bike-sharing systems is challenging due to complex dynamics, including strong seasonality and trends, as well as dependence on external variables such as weather and calendar effects. Furthermore, data sparsity, particularly at the individual station level, presents a significant obstacle to building reliable predictive models. By accurately predicting bike demand, operators can improve redistribution and station availability, fostering a more reliable and sustainable urban mobility system.

This paper addresses these challenges by investigating two distinct forecasting tasks using a real-world dataset from a Slovenian city. First, we examine short-horizon, one-day-ahead predictions for individual stations to quantify the impact of data sparsity on forecastability. Second, we evaluate the accuracy of 90-day long-horizon forecasts for the total rental volume aggregated across all stations. We compare a suite of models, including classical machine learning approaches and LSTM neural networks [5], and explicitly analyze the trade-off between using longer historical data versus shorter, weather-enriched data to improve predictive accuracy. This work aims to help the bike-sharing systems to improve operational efficiency, reduce bike shortages, and inform city planning initiatives related to sustainable transportation.

*Both authors contributed equally to this research.

Information Society 2025, Ljubljana, Slovenia

https://doi.org/10.70314/is.2025.sikdd.7

Prior studies on bicycle rental forecasting often use the Washington, D.C. dataset [4]. Du et al. [2] addressed long-horizon prediction, while Karunanithi et al. [6] focused on short-horizon forecasting, both achieving results comparable to ours. In contrast, our dataset differs substantially by including station-level information, which enables per-station forecasting. We tackle both short- and long-horizon tasks, as well as the analysis of the impact of exogenous weather variables.

2 Data

The dataset we used originates from a public bicycle rental service in a Slovenian city. It contains daily rental counts for individual stations within the municipality, covering the period from January 1, 2021, to May 15, 2025. Although the dataset also records bike return counts, our work focuses exclusively on rentals.

2.1 Features

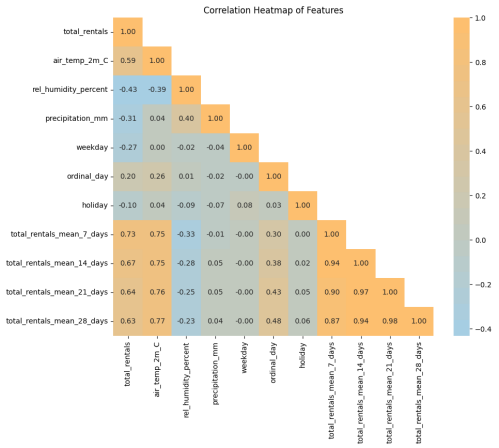


Figure 1: Pearson correlation coefficients of our features

Dependent Variable: The target feature we are forecasting.

- **total_rentals:** The total daily number of bike rentals. Based on the task, this is either the total count across all stations or per-station bike rental count.

Independent Variables: The features used for prediction.

- **Temporal Features:**
  - **date:** The specific date.
  - **ordinal_day:** The day number within the year.
  - **weekday:** A category for the day of the week.
  - **holiday:** Indicator (0 or 1) if the day is a holiday.
- **Weather-Related Features:** *Note: Our weather data only spans the date range of 2024-01-01 to 2025-05-14*
  - **air_temp_2m_C:** Air temperature.
  - **rel_humidity_percent:** The relative humidity.
  - **precipitation_mm:** The precipitation per square meter.

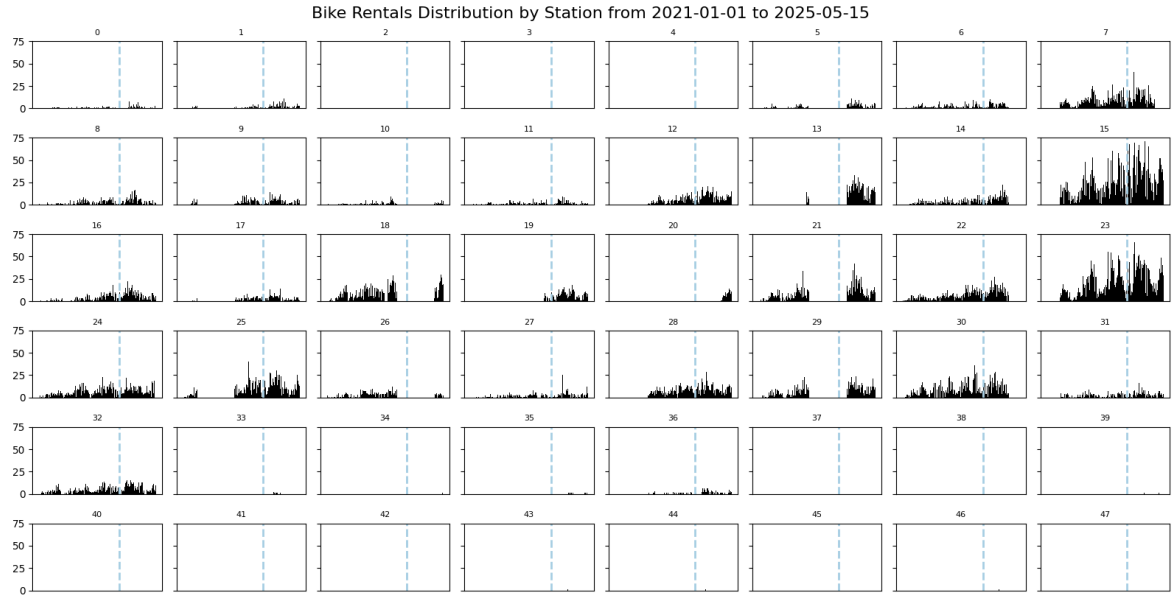


Figure 2: Distribution of bike rentals across all stations. The vertical blue line indicates the start of the year 2024.

2.2 Data Preprocessing

The dataset structure prevented distinguishing missing values from true zeros (i.e., days when no rentals occurred), so all empty or null entries were treated as zeros. This resulted in sparsity for some stations, in which many entries had little information on rental activity. To prevent this impacting our analysis, we excluded those with more than 33% zero entries, retaining 25 stations out of the original 48. For the machine learning methods described later, we also implemented a set of **lagged features:**

- **total_rentals_mean_7_days:** Average rental count over the 7 days preceding the current data point.
- **total_rentals_mean_14_days:** Average rental count over the 14 days preceding the current data point.
- **total_rentals_mean_21_days:** Average rental count over the 21 days preceding the current data point.
- **total_rentals_mean_28_days:** Average rental count over the 28 days preceding the current data point.

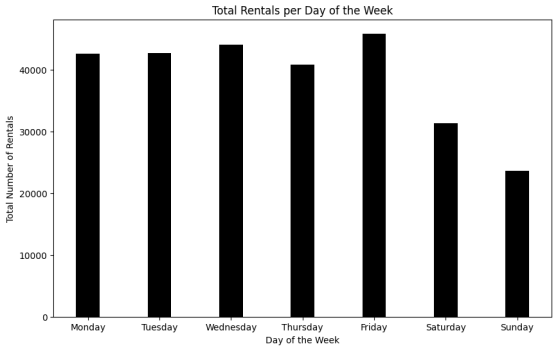


Figure 3: Rentals per day of the week

2.3 Exploratory Data Analysis

The data exhibits pronounced weekly and monthly seasonalities, as well as non-stationarity, as illustrated in Figures 3 and 4. Annual patterns show rental activity declining in winter, rising in spring, peaking in summer, and gradually decreasing in autumn, with weekends consistently exhibiting lower rental counts. Anomalous behavior was observed in the winter of 2024, when rental counts were markedly higher than typical seasonal levels.

The Pearson correlation coefficients (Figure 1) between features related to bicycle rentals indicate that the number of daily rentals (`total_rentals`) is strongly and positively associated with recent rental trends, as reflected by correlations of 0.73, 0.67, 0.64, and 0.63 with the 7-, 14-, 21-, and 28-day moving averages, respectively. A strong positive correlation is also observed with air temperature (0.59), whereas moderate negative correlations are found with relative humidity (-0.43) and precipitation (-0.31), suggesting that rentals are more frequent on warm, dry days. Weaker associations are present with the day of the week (-0.27) and holiday status (-0.10). As expected, the moving average features exhibit high intercorrelation (e.g., 0.94 between the 7- and 14-day means) due to their overlapping calculation windows.

3 Experiments

This study pursued two primary objectives. First, we examined the feasibility of forecasting bicycle rentals one day in advance and evaluated how forecastability varies across stations with different data sparsity. Second, we investigated long-horizon forecasting over a 90-day period, focusing exclusively on predicting the total number of rentals. In this task, standard machine learning models were trained on historical data and then used recursively to generate forecasts for the entire period. Due to this setup, the results for **DS_W** suffer from data leakage. Specifically, a single model is trained using past rental counts and future weather information, so, for example, predicting rentals in July involves access to the actual recorded weather conditions for that month, which artificially improves performance.

3.1 Training and Test Data Split

Because the available weather data was limited to the years 2024 and 2025, while the rental dataset spanned from 2021 onward,

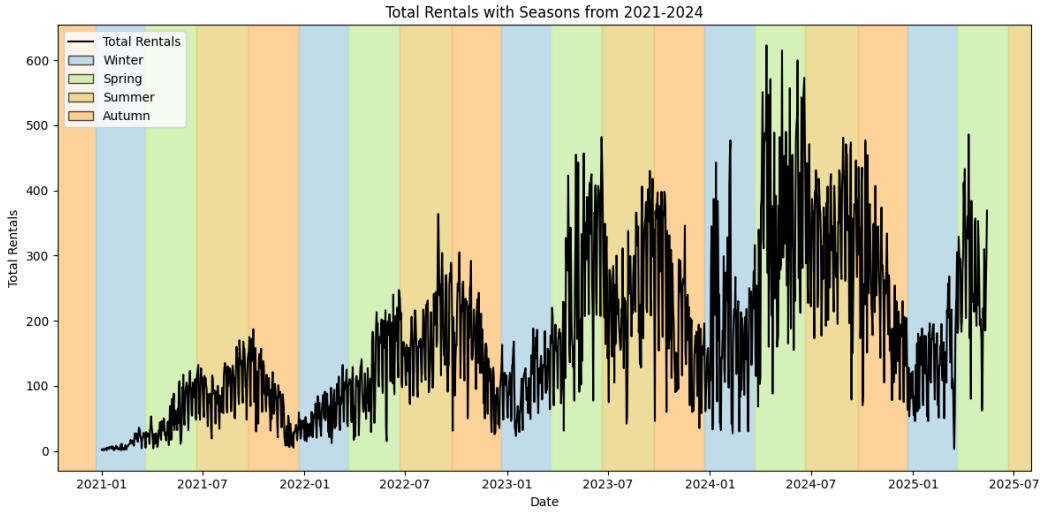


Figure 4: Bike rental data with temperate seasons

we constructed three distinct datasets. Here, each *entry* corresponds to a single day and includes rental data for all stations. The first dataset, **DS_W**, combined rental and weather data (498 entries). The second, **DS_NO_W**, included only rental data for the same period (498 entries). The third, **DS_FULL**, comprised the complete rental dataset without weather data (1,593 entries).

The data splitting strategy differed in the two tasks. For the station-level one-day-ahead forecasting task, each dataset was divided into 25 subsets, corresponding to individual stations. Within each subset, random sampling was used to split the data into training and testing sets with an 80:20 ratio. The target variable in each subset is the specific station's rental count.

For the long-horizon task, no station-level subdivision was performed, as only total rental counts were modeled. The final 90 days were used as the test set—roughly corresponding to a temperate season—allowing us to assess whether the models capture seasonal patterns in a new period while maintaining realistic temporal separation between training and testing data.

3.2 Models and Algorithms Used

For the long-horizon forecasting task, the **AutoARIMA** model served as the baseline, while for the one-day-ahead forecasting task, the baseline was the **Mean Regressor**, which predicts using the 7-day lag mean.

We evaluated several machine learning models, including **Random Forest** (500 trees, max_features=0.9), **Gradient Boosting** (500 estimators), **Linear Regression**, and **SVM** ($C = 10$, degree=2, $\gamma = 0.1$, linear kernel). The hyperparameters for the Random Forest and SVM models were selected using a grid search optimization procedure; the rest of the models used default parameters. For the Random Forest model, only the `max_features` parameter was tuned.

We additionally tested deep learning approaches: **LSTM** (input size = 96, RMSE loss, 10,000 epochs) and **N-BEATSx** (input size = 96, RMSE loss, 500 epochs).

Training was performed on a laptop equipped with an RTX 3050 GPU (4 GB VRAM), which constrained the range of hyperparameter configurations that could be explored, particularly for the neural network-based approaches.

3.3 Performance evaluation

Model performance was assessed using Root Mean Squared Error (RMSE) and Mean Absolute Percentage Error (MAPE). Additionally, the Relative Root Mean Squared Error (RRMSE)[1] was used to enable inter-station performance comparisons in the one-day-ahead forecasting task. RRMSE is defined as follows:

$$\text{RRMSE} = \frac{\text{RMSE}}{\overline{y}} \tag{1}$$

where $\overline{y}$ is the mean of the target values.

3.4 Results

The results for the one-day-ahead task are presented in Table 1, with station forecastability visualized in Figure 5. The long-horizon task outcomes are presented in Table 2.

4 Discussion and conclusion

For the one-day-ahead forecasting task, a clear correlation exists between station data sparsity (Figure 2) and forecastability (Table 1). Stations with fewer rentals or gaps in data are easier to predict accurately. Interestingly, using the **DS_FULL** dataset—which includes data prior to 2024—can reduce modeling accuracy for certain stations. Including weather features in **DS_W** leads to little or no improvement compared to **DS_NO_W**. For the long-horizon task, including weather data proves beneficial, as both classical machine learning models and neural networks show improved performance (Table 2). However, as described in the Experiments section, the machine learning results on **DS_W** are overly optimistic due to data leakage: the models are trained on historical rental counts while also accessing future weather information during recursive forecasting (e.g., predicting rentals in July uses the actual recorded weather for that month). This is reflected in the comparison with **DS_NO_W**, where classical machine learning methods achieve a 33% mean reduction in MAPE, while neural network approaches show only a 17% mean decrease, suggesting that the apparent benefit of weather data is amplified for classical methods because of this setup. Our results echo [3] where Gradient Boosting models matched or outperformed neural networks on several datasets, demonstrating the effectiveness of simpler models. While neural networks

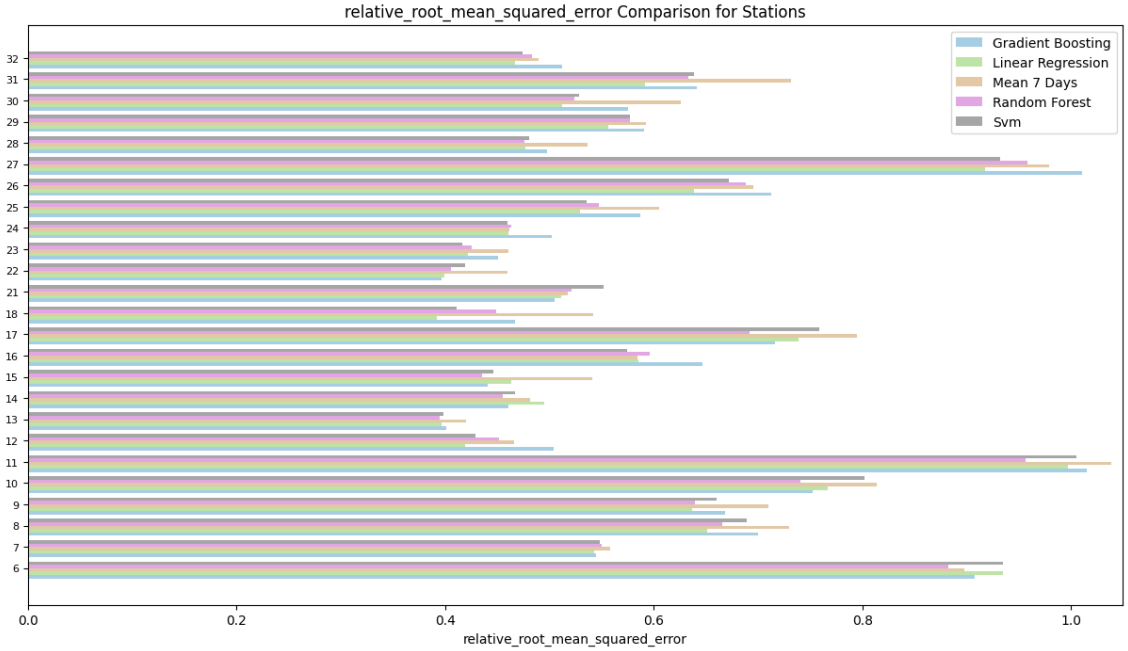


Figure 5: Model performance of one-day-head forecasting for different stations for DS_W

Table 1: Average RRMSE of all models of one-day-ahead forecasting across datasets (RRMSE) and stations.

Station	DS_FULL	DS_NO_W	DS_W
6	0.9210	**0.9097**	0.9116
7	0.5849	**0.5439**	0.5488
8	0.7948	**0.6821**	0.6872
9	**0.6532**	0.6646	0.6631
10	0.9550	**0.7747**	0.7753
11	1.0110	1.0034	**1.0027**
12	0.6028	0.4649	**0.4540**
13	0.6601	**0.4000**	0.4022
14	0.6902	0.4840	**0.4720**
15	0.5218	0.4780	**0.4652**
16	0.7185	0.5984	**0.5975**
17	0.8336	**0.7337**	0.7402
18	0.5274	0.4670	**0.4522**
21	0.5476	0.5218	**0.5215**
22	0.5198	0.4171	**0.4160**
23	0.4783	0.4363	**0.4349**
24	0.4896	0.4760	**0.4696**
25	0.6834	**0.5570**	0.5608
26	**0.6506**	0.6897	0.6812
27	**0.9463**	0.9898	0.9595
28	0.5580	**0.4898**	0.4936
29	0.6008	**0.5761**	0.5788
30	0.5941	**0.5496**	0.5531
31	0.8952	**0.6452**	0.6474
32	0.5453	0.4873	**0.4851**
Average	**0.6793**	**0.6016**	**0.5989**

could potentially benefit from hyperparameter optimization, the same applies to other methods as well. A detailed comparison of different approaches was beyond the scope of this preliminary study but could be explored in future work.

Table 2: Model performance of 90-day forecasting across datasets (RMSE / MAPE)

Model	DS_FULL	DS_NO_W	DS_W
AutoARIMA	120.09 / 0.9525	118.50 / 0.9954	118.50 / 0.9954
Random Forest	108.29 / 0.7153	100.94 / **0.7431**	76.36 / 0.7014
Gradient Boosting	95.17 / 0.7451	94.96 / 0.9584	74.69 / **0.5513**
Linear Regression	**90.29** / 0.9372	84.78 / 1.0816	71.71 / 0.8872
SVR	94.86 / 0.8893	**87.12** / 0.9507	**67.95** / 0.8036
LSTM	112.05 / **0.7133**	125.13 / 0.8494	130.00 / 0.8070
NBEATSx	106.49 / 1.0329	128.90 / 0.9972	117.45 / 0.7246
Average	**103.89 / 0.8551**	**105.76 / 0.9394**	**93.81 / 0.7815**

Acknowledgements

This work was supported in part by the Slovenian Research Agency through core funding for the programme Knowledge Technologies (No. P2-0103) and by the project *KReATIVE*, funded through NetZeroCities under the European Union's Grant Agreement No. HORIZON-RIA-SGA-NZC 101121530. We also thank Tea Tušar for her suggestions regarding data visualization.

References

[1] Shikun Chen and Nguyen Manh Luc. 2022. Rrmse voting regressor: a weighting function based improvement to ensemble regression. *arXiv preprint arXiv:2207.04837*.

[2] Jimmy Du, Rolland He, and Zhivko Zhechev. 2014. Forecasting bike rental demand. *Gebhard, K., & Noland*.

[3] Shereen Elsayed, Daniela Thyssens, Ahmed Rashed, Lars Schmidt-Thieme, and Hadi Samer Jomaa. 2021. Do we really need deep learning models for time series forecasting? *CoRR*, abs/2101.02118. https://arxiv.org/abs/2101.02118 arXiv: 2101.02118.

[4] Hadi Fanaee-Tork. 2012. Bike sharing dataset. Dataset. (2012). https://www.kaggle.com/datasets/marklvl/bike-sharing-dataset.

[5] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9, 8, 1735–1780.

[6] Meerah Karunanithi, Parin Chatasawapreeda, and Talha Ali Khan. 2024. A predictive analytics approach for forecasting bike rental demand. *Decision Analytics Journal*, 11, 100482. DOI: https://doi.org/10.1016/j.dajour.2024.100482.

Predicting Traffic Intensity on Motorway Sections

Matic Kladnik†
Jozef Stefan International Postgraduate School
Ljubljana, Slovenia
matic.kladnik@gmail.com

Dunja Mladenić
Department of Artificial Intelligence
Jozef Stefan Institute
Ljubljana, Slovenia
dunja.mladenic@ijs.si

Abstract

This paper addresses predictions of traffic intensity on sections of motorways. Predictions are computed for timespans from 24 hours up to 52 weeks. With our adaptive system, we update predictions with newer ones, once additional features can be computed from available data. We use historic context of past traffic intensities on specific sections at specific periods of time, as well as semantic context about the target period. We have evaluated our methodology with multiple machine learning models and compared performances for various timespans on a specific motorway section. The evaluation results show that our methodology improves predictions for specific periods over time.

Keywords

Motorway, traffic intensity, prediction, regression, system, semantic context, evaluation, machine learning

1 INTRODUCTION

A prediction system for predicting traffic intensity on motorway sections can support a wide range of decision making, strategic, and operative processes at the motorway management organization. It can also support end users, such as daily commuters, tourists, and other drivers with their planning of a trip.

The focus of this paper is on architecture of the motorway traffic intensity prediction system as well as on the evaluation of the machine learning models that were trained to produce the predictions for various timespans.

2 PROBLEM SETTING AND DATA

The objective of the proposed methodology is to make long term and medium-term predictions of traffic intensity or frequency (vehicle count) on various sections of motorway based on historic data of traffic counters, semantic context of motorway stations, and semantic context of time periods. Predictions serve the motorway management company for better planning of construction projects and to find the least intrusive time slots for road maintenance work. It also serves the motorway drivers when planning a trip.

2.1 Traffic Counters

There are close to one hundred traffic counters that we consider for predictions. Each counter is supported by a pair of inductive loops that are laid into the asphalt of the road. Signals are processed, sent through an IoT communication device and stored into the database.

In the data, there are counts or frequencies of total vehicles, and counts by vehicle types (passenger car, transport truck, bus) for each hour-long time period. E.g. number of vehicles from 8:00 to 9:00 for each of the lanes of a specific motorway section separately.

2.2 Semantic Context

For each of the examples in the dataset we produce semantic context features. For each day and time of day period, we produce semantic context features to inform the model whether a certain time period is on a workday or a weekend, whether the specific time period falls into the morning rush hours or the afternoon rush hours. These semantic features give additional information to improve the performance of machine learning models.

2.3 Data Processing

After downloading the data from the motorway counters via an API of the data provider, we additionally process it to increase consistency and reliability of predictions.

During data processing, we merge data from all lanes of a specific motorway section, which is usually denoted with neighboring towns and the direction of the motorway section.

3 METHODOLOGY DESCRIPTION

We propose a prediction system that includes incorporation of multiple machine learning models to deliver the most reliable predictions based on available data and the timespan for which the system is making predictions of traffic intensity.

To improve prediction accuracy, we make medium-term and long-term predictions. In our case, long-term predictions are made from 1 week to 52 weeks in advance for a specific 1-hour

Information Society 2025, 6–10 October 2025, Ljubljana, Slovenia

http://doi.org/10.70314/is.2025.sikdd.25

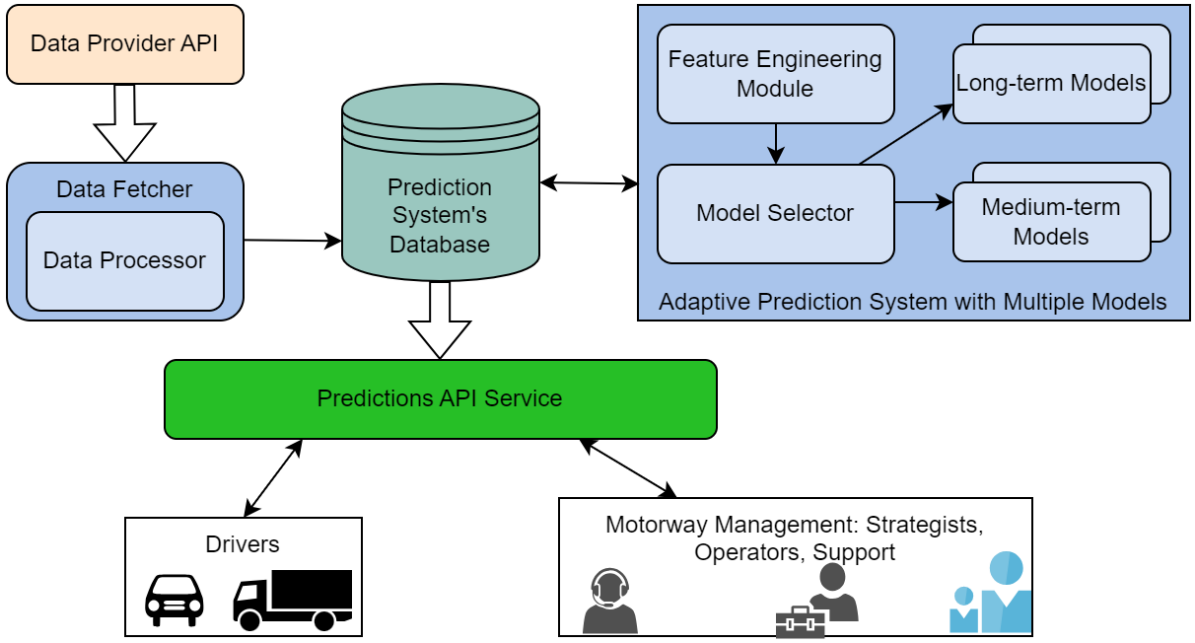


Figure 1: Diagram of the system for producing and distributing predictions of traffic intensity on motorway sections

time period for a specific day of week. Which means that we can make up to 52 predictions when conducting long-term predictions after receiving a new data example, e.g. traffic frequency for a specific 1-hour time period (e.g. 14:00-15:00) for a specific day in time (e.g. Monday).

Whereas medium-term predictions are those that predict from less than 24 hours up to 1 week in advance. For medium-term predictions, we take more features for recent traffic frequency into account for improved accuracy.

Long-term predictions are useful when making decisions for actions that are several weeks or months in the future, while medium-term predictions are more useful when making decisions for actions that will take place from 1 to 7 days in the future.

We have a separate machine learning model for each of the included counters on the motorway to better adjust to specifics of the traffic dynamic of a specific counter when making predictions of traffic frequency. We have also trained several general-purpose models that are trained on a group of counters or all counters. These are present to support counters with short data history.

Predictions are exposed through a REST API service and are available upon request. They are computed and updated regularly, e.g. daily or hourly. More approaches in [1][6].

3.1 Machine Learning Models

To compute predictions of traffic intensity in the future, we use regression machine learning models. We have trained and evaluated several models with the usage of different machine learning algorithms. These are: linear regression, SVM (SVR – Support Vector Machine for Regression), and XGBoost, which is an ensemble model of decision trees.

Features for training models and making predictions are engineered in such a way that each one of the models can use the whole set of features. E.g. we use a one-hot encoding approach when a feature would otherwise have multiple categorical values. We focus on training a specific model for each of the motorway sections that were part of the research. Note that a more general model, trained on data from multiple motorway sections could be more appropriate for motorway sections that have been newly added and do not have enough historical data to support training of a reliable machine learning model with sufficient evaluation period. We use up to 7 features that are based on historic data, 7 time period features, and 6 semantic context features for a specific time period and location.

Model training processes use MAPE (Mean Absolute Percentage Error, used interchangeably with MARE – Mean Absolute Relative Error). More on relevant machine learning models and metrics in references ([2][3][4][5]).

3.2 Prediction System Description

We continue with the description of our proposed prediction system. The system consists of two main subsystems. One for periodically computing and storing traffic intensity predictions for various time spans. And another for delivering predicted traffic intensity via a REST API service.

As we can see on Figure 1, the system fetches data from the data provider's REST API service. Data is processed after retrieval and sent into a table of prediction system's database. This data is read periodically by the adaptive prediction system.

Once a new value is processed by the system, it checks if there are any additional models with a shorter timespan available, compared to the model used for the currently available prediction. The system prioritizes predictions from models with a shorter timespan in order to update the database with the most reliable predictions available at the time. E.g., prediction with a 1-month timespan succeeds and replaces the prediction with a 3-month timespan.

Different long-term and medium-term models can be trained using different machine learning algorithms, depending on the

algorithm that performed the best during the evaluation of the models.

Once updated the predictions are stored in the database, they are available to users, such as strategists, operators and support specialists within the motorway management organization. Or end users of the motorway, such as drivers of cars, trucks, buses, etc. A key advantage of this approach is that drivers and motorway operators and specialists get insights that are based on the same predictions for traffic intensity, which supports greater transparency of information and stronger compatibility of different applications for end users and motorway professionals.

E.g. the system can support long-term planning for larger maintenance or reconstruction projects for up to 1 year ahead, as well as long-term planning of road users. For instance, drivers can plan their holidays and the time of their commute ahead. And highway maintenance operators can find the most optimal schedule for short maintenance work.

4 EVALUATION

We continue with the evaluation of the machine learning models. To compare models, trained with different algorithms, we use the evaluation results for the same motorway section on the Slovenian motorways. We use the period from 1 May 2024 until 5 May 2025 for evaluation.

We use Scikit-learn library[7] to train the linear regression (using ordinary least squares approach) and SVM (SVR) models and the XGBoost library[8] to train the XGBoost models. SVM model is trained using the RBF kernel, and with scaled gamma hyperparameter. In majority of motorway sections, XGBoost models with a maximum depth of 6 performed the best which is why we used models with the same hyperparameter value for the following analyses. We use gbtree as the booster, while the learning rate is 0.3.

Table 1: Model Performance Comparison

timespan	algorithm	MAE	RMSE	MAPE
24 hours	XGB	39.43	62.75	10.5%
24 hours	SVM	42.38	65.86	11.5%
24 hours	lin. reg.	43.14	66.93	11.6%
7 days	XGB	45.66	70.69	11.6%
7 days	SVM	43.70	68.91	12.1%
7 days	lin. reg.	43.51	69.04	12.1%
4 weeks	XGB	57.30	88.56	13.9%
4 weeks	SVM	50.20	77.86	14.1%
4 weeks	lin. reg.	51.33	78.63	14.7%
52 weeks	XGB	88.33	121.93	20.9%
52 weeks	SVM	53.54	84.49	14.9%
52 weeks	lin. reg.	70.46	96.98	21.3%

We evaluated the models on a little over 1 year of test data, which was not included in the training or validation part of the process.

We continue with the analysis of the model performances as seen in Table 1. If the timespan attribute's value is '7 days', it means that the model predicts 7 days into the future. We use several metrics to describe the performance of the models. These are: MAE (Mean Absolute Value), RMSE (Root Mean Square Error), and MAPE (Mean Absolute Percentage Error). MAPE is a crucial metric as it shows relative errors in percentages which is key when evaluating the models as traffic frequency varies significantly throughout different parts of the day.

We can see some interesting performance dynamics of the models. The XGBoost model performs the best for 24-hour timespan, with a significant performance uplift of at least 1 percentage point in MAPE, compared to the other two models. It is also better in the other two metrics: MAE and RMSE.

We continue with the performance analysis of the long-term predictions. For the 7-day timespan, the XGBoost model is still noticeably better than the other two models with a 0.5 percentage point uplift in performance. For the 4-week timespan, XGBoost still holds a small lead in the key metric (MAPE), whereas the SVM model has significantly better results when considering just MAE and RMSE metrics. For the 52-week timespan, we can see an interesting dynamic as the SVM model takes a significant lead in performance as it is the only one with the MAPE value of less than 15%, whereas the MAPE values of the other two models surpass 20%.

The dynamic is likely caused by a reduced set of features as there are significantly less historic traffic count features that are included when making predictions with a 52- week timespan. It seems this has a significantly negative impact on training the XGBoost model, which is a tree ensemble model, while having additional features available gave the XGBoost model an edge for predictions with a timespan up to 4 weeks, especially up to 7 days.

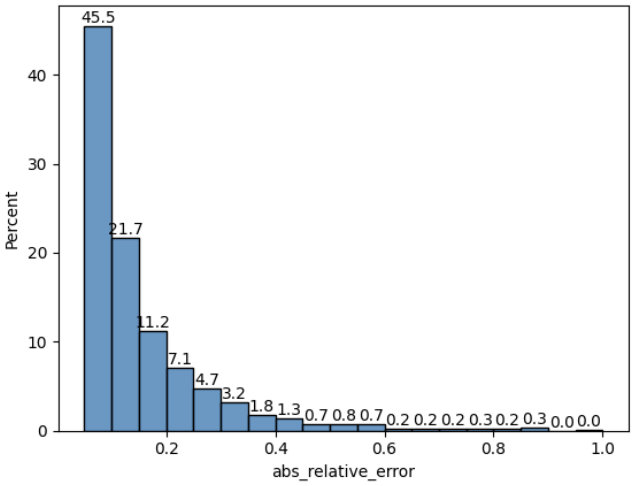


Figure 2: Distribution of absolute relative errors by 5% buckets for XGBoost 7-day timespan model

On Figure 2 we can see how absolute relative errors are distributed if they are split into 5% absolute relative error buckets. We can see that in 45.5% of the cases, the absolute relative (or percentage) error of the predicted traffic frequency is less than 5% of the actually measured traffic frequency. 21.7% of predictions have a relative error between at least 5 and

(excluding) 10 percent, and 11.2% of predictions have a relative error between 10 and 15 percent.

This means that in 78.4% of predictions, the relative error was less than 15%, which can be considered as a sufficiently good performance for the models to support a sufficiently reliable traffic intensity prediction system.

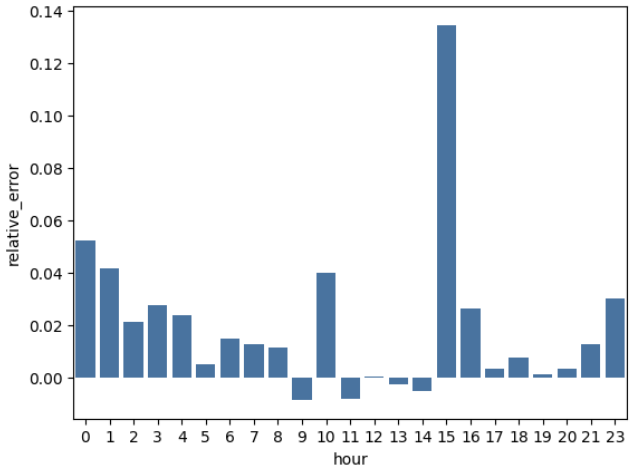


Figure 3: Mean relative errors by each hour of the day for XGBoost 7-day timespan model

We continue by analyzing the distribution of mean relative errors by each hour of the day as seen on Figure 3. We can see that the model generally tends to slightly overestimate or overshoot with its predictions. Especially during the night-time periods, when there are fewer vehicles on the motorway.

In the mean aggregate, there is less than a 2% mean relative error during the morning rush hours (at 6:00-7:00, 7:00-8:00, and 8:00-9:00). It is the highest during the 15:00-16:00 period, with more than 13% of mean relative error. However, the error is substantially smaller during other afternoon rush-hour periods, 14:00-15:00, 16:00-17:00, and 17:00-18:00, where it remains under 4%. Apart from the 15:00-16:00 period, the mean relative errors are consistently under 6%. When the model does undershoot or underestimate with its prediction, the mean relative error is less than 2%, close to 1%.

We can see a spike of mean relative error at the 15:00-16:00 period. Upon investigation, it turns out only around 20 vehicles were counted in the data for a specific period, which is unusual for this period and likely a consequence of a traffic accident or some issue with data collection.

We have also conducted an aggregated evaluation of models on 10 various motorway sections, where mean MAPE values were 14%, 15%, 18%, and 20% for 24-hour, 7-day, 4-week and 52-week timespans respectively. Predictions for sections near the capital city were generally less reliable than others.

4.1 Evaluation Insights

When considering the results of the evaluation of trained machine learning models for specific motorway sections, we have gathered several key insights.

In some examples, we could not compute all features due to missing values in data, meaning that certain features had NaN values after computing historic time-series features with Pandas' shift function. In this case there is a strong advantage of having a decision tree ensemble model (e.g. XGBoost) as a backup, even if it is not the best performing model for a certain timespan. This is due to the ability of the tree ensemble models to apply only those trees that are covered by features with available values. In this case the predictions are generally less accurate but possible.

Another key insight is that the evaluation supports our proposed methodology with multiple models to improve the performance of the predictions for each included timespan.

Another useful insight is that different algorithms can produce the best models for different timespans on the same motorway section. As was the case with the SVM model in our evaluation.

5 CONCLUSION

We have overviewed the methodology that we use as the foundation for our proposed system for predicting traffic intensities on motorway sections. Including the adaptive prediction system and the supporting machine learning models that support making predictions for various timespans to, in time, improve already available predictions for specific time periods in the future. We have also overviewed the evaluation of the trained machine learning models and found some useful insights that support our proposed prediction system.

Compared to related work, the key contributions in our methodology are significantly longer prediction timespans, inclusion of semantic context, and higher adaptability to data.

Based on the presented current evaluation results, our methodology produces predictions with sufficient reliability to support long-term decision making of various roles.

For further improvements to the system, we could train and evaluate some deep learning models and models that are based on the transformer architecture, as well as some other time-series forecasting procedures, such as Facebook Prophet. We could also engineer additional semantic context features for further improvements to the performance of the existing models. For additional improvements for shorter timespans, we could also include weather forecast data.

References

[1] Bernardo Gomes, Jose Coelho, Helena Aidos. 2023. A survey on traffic flow prediction and classification. In *Intelligent Systems with Applications*, vol. 20. DOI: https://doi.org/10.1016/j.iswa.2023.200268

[2] Jithin Raj, Hareesh Bahuleyan, Lelitha Devi Vanajakshi. 2016. Application of Data Mining Techniques for Traffic Density Estimation and Prediction. *Transportation Research Procedia, vol 17*. DOI: https://doi.org/10.1016/j.trpro.2016.11.102

[3] Yuyu Zhu, QingE Wu, Na Xiao. 2022. Research on highway traffic flow prediction model and decision-making method. *Scientific Reports*, vol. 12. DOI: https://doi.org/10.1038/s41598-022-24469-y

[4] Carl Goves, Robin North, Ryan Johnston, Graham Fletcher. 2016. Short Term Traffic Prediction on the UK Motorway Network Using Neural Networks. *Transportation Research Procedia, vol. 13, 184-195.* DOI: https://doi.org/10.1016/j.trpro.2016.05.019

[5] Adriana-Simona Mihaita; Zac Papachatgis; Marian-Andrei Rizoiu. 2020. Graph modelling approaches for motorway traffic flow prediction. 2020. *IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*. DOI: https://doi.org/10.1109/ITSC45102.2020.9294744

[6] Sayed A. Sayed, Yasser Abdel-Hamid, and Hesham A. Hefny, 2022. Artificial Intelligence-Based Traffic Flow Prediction: A Comprehensive Review. *Pre-review*. DOI: http://dx.doi.org/10.21203/rs.3.rs-1885747/v1.

[7] Scikit-learn: https://scikit-learn.org

[8] XGBoost: https://xgboost.ai/

Empowering Youth on Smart Cities with AI Solutions to Community and Urban Challenges Towards SDG 11

Mustafa Zaouini†, Lee Chana, Ruben Frank, Kim August
AI in Africa
Johannesburg, South Africa
mus@fliptin.io

Joao Pita Costa, Davor Orlic, Mihajela Črnko
IRCAI, Quintelligence
Ljubljana, Slovenia
joao.pitacosta@ircai.org

Yousef Rahmani
ToumAI
Rabat, Morocco
odin@toum.ai

Rayan Kassis, Swethal Kumar
EnergyAED
London, UK
rayan@aed.energy

Luka Stopar
Solvesall
Ljubljana, Slovenia
luka.stopar@solvesall.com

Sohaib Souss, Wahid Laleeg, Yassine Bounouader
SLTVERSE
Casablanca, Morocco
sohaibsoussi@gmail.com

Asmae Lamgari, Maroja Zoubir, Hajar Doukhou
University Mohammed V (UM5)
Rabat, Morocco
asmaelamgarim@gmail.com

Ouidad Mochariq, Zahira Elmelsse, Chaimae Fadil
ENSA National School of Applied Sciences (ENSA-M)
Marrakesh, Morocco
o.mochariq3846@uca.ac.ma

Abstract / Povzetek

Achieving Sustainable Development Goal 11 − ensuring cities are inclusive, safe, resilient, and sustainable − remains a pressing global priority. In this pursuit, Artificial Intelligence (AI) has emerged as a transformative driver of urban innovation, enabling policymakers, academic institutions, and industry stakeholders to make data-driven decisions for complex urban systems such as housing, transportation, energy, and infrastructure. Despite its potential, the vast scale, variety, and fragmentation of urban data, coupled with the rapid evolution of AI technologies, create significant challenges in converting SDG 11-related information into practical solutions. This paper reports on the results of the AI4SDG11 programme, which combined expert community building, knowledge exchange, and competitive challenges. The programme brought together 50 students and 30 startups I 15 locations worldwide, to develop AI-driven solutions targeting key aspects of urban sustainability. Using diverse machine learning techniques, participants addressed challenges including intelligent mobility systems, efficient waste management, smart and efficient urbanism, and climate-resilient urban planning. Conducted in 2025, this initiative formed part of a youth-focused innovation challenge co-organized by AI in Africa, the International Research Centre on Artificial Intelligence (IRCAI), and GITEX, with the goal of promoting interdisciplinary innovation and strengthening regional AI capacity for sustainable urban development.

Keywords / Ključne besede

Machine learning, text mining, large language models, community engagement, urbanism, mobility, AI competition, AI Community

†Corresponding author

Information Society 2025, 6–10 October 2025, Ljubljana, Slovenia

http://doi.org/10.70314/is.2025.sikdd.18

1 Introduction

Established by the United Nations as an essential goal for the forthcoming 2030, the Sustainable Development Goal 11 (SDG 11) — *"Make cities and human settlements inclusive, safe, resilient and sustainable"* — reflects a critical global commitment to improving urban living conditions amid increasing urbanization, population growth, and environmental stress. With more than half of the world's population now residing in cities—and projections estimating two-thirds by 2050—the urgency of building sustainable urban environments has never been better fit. In this context, AI has emerged as a transformative tool capable of reshaping how cities are planned, managed, and experienced. AI technologies offer powerful capabilities to harness vast amounts of urban data, generate predictive insights, and support evidence-based decision-making. From optimizing public transportation systems to monitoring air quality, improving waste management, and enabling climate-resilient infrastructure, AI is at the forefront of innovative urban solutions worldwide. However, the deployment of AI in support of SDG 11 varies significantly across regions, influenced by differences in digital infrastructure, data availability, institutional capacity, and local priorities [1].

In Africa, AI is increasingly being applied to address urban informality, mobility challenges, and infrastructure gaps. For instance, AI-powered geospatial mapping tools are being used to identify informal settlements in rapidly growing cities such as Nairobi and Lagos, helping governments to improve service delivery and urban planning [2]. In North African cities, machine learning models have been developed to optimize water distribution in drought-prone areas and to improve traffic flow in congested urban corridors. AI is also being tested for predictive waste collection and smart energy use in off-grid communities. These solutions are particularly valuable in regions where resources are limited, and where rapid urban growth creates pressure for low-cost, scalable interventions [2].

On the other hand, in Europe, AI applications in cities often focus on enhancing sustainability, efficiency, and citizen engagement. Examples include real-time public transport optimization in cities like Helsinki and Barcelona [3], AI-based air pollution forecasting in Paris [4], and intelligent energy management

systems in smart buildings across the Netherlands and Germany [5]. Many European municipalities are also investing in AI-driven participatory governance platforms, enabling data-informed urban policymaking that incorporates citizen feedback [5]. Furthermore, [6] highlights how AI can extract and analyze news media information to enhance knowledge and understanding of water-related extreme events, supporting improved disaster risk reduction..

This paper presents the outcomes of a collaborative youth AI innovation programme, including AI mentorship and challenges aimed at exploring the impact of AI on SDGs. It builds on the related initiative initiating the programme in 2026 under the focus of Water Sustainability to progress SDG 6 (see [7] and [8]), and refocuses the approach addressing SDG 11-related problems through applied machine learning solutions. The initiative brought together 50 students and 20 professors across 10 research institutions in North Africa, as well as 30 AI startups and domain experts worldwide culminating in 30 projects and initiatives tackling real-world urban challenges. By leveraging AI and data science, these teams addressed issues ranging from urbanism and mobility to waste management and climate resilience—drawing on lessons and methods from both African and European contexts. The competition, co-organized by AI Africa and IRCAI in a collaboration with GITEX, short for Gulf Information Technology Exhibition, being one of the world's largest technology and innovation events, held annually in Dubai, United Arab Emirates. The event held in May 2024 [7], served as a model for interdisciplinary, cross-regional collaboration in the pursuit of sustainable urban futures.

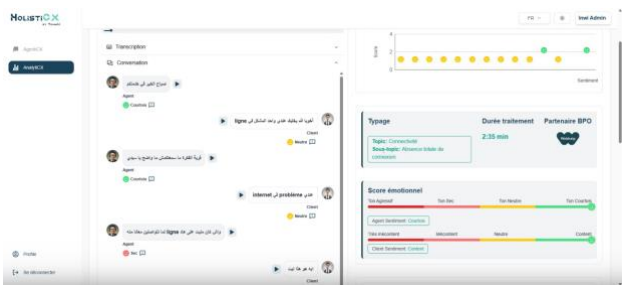


Figure 1: Screenshot of the AI engine ToumAI, winner of the AI4SDG11 startup competition at the inaugurating edition of GITEX Europe, Berlin, as a prime example of the relevance of languages in the resilience of cities and communities

2 AI4SDG Programme Methodology

The AI4SDG Programme, spearheaded by IRCAI under the auspices of UNESCO, in collaboration with AI in Africa and GITEX, is a transformative initiative designed to harness artificial intelligence to address the United Nations Sustainable Development Goals (SDGs). With a focus on capacity building, entrepreneurship, and ethical AI deployment, the programme connects technological innovation with global sustainability challenges, particularly in the Global South.

At the core of AI4SDG is a multi-pronged approach integrating certified training, competitive innovation events, and startup acceleration. Launched through global showcases and pitch competitions at major GITEX events across Africa, Asia, Europe, and the Middle East, the initiative provides a dynamic platform for students, researchers, and entrepreneurs to ideate, prototype, and scale AI solutions aligned with specific SDGs. Previous editions have focused on Water Sustainability (SDG 6) and Sustainable Cities and Communities (SDG 11), while the 2026 programme will extend to all 17 SDGs. The key components include:

- **Research2Startup Competition**: A 4–6 week programme blending AI education, design thinking, and acceleration tracks for startups and university spinouts, culminating in regional and global pitch events.
- **Certified AI for SDG Training**: Professional certification tracks for corporate teams, startup founders, and SMEs, focusing on topics like large language models, AI governance, ethical data practices, and generative AI applications.
- **AI4SDG Lab Accelerator**: A 3–6 month cohort-based programme supporting university-originated AI startups through mentorship, technical workshops, and investor networking, culminating in a high-profile Demo Day at *GITEX Global*.

The programme not only equips participants with practical AI competencies but also facilitates access to global networks, funding opportunities, and collaboration through GITEX's innovation ecosystem. It champions responsible AI development by emphasizing ethics, transparency, and inclusivity, while offering tangible incentives such as certifications, cash prizes, MVP co-development and impactful international exposure through IRCAI and GITEX channels. In doing so, AI4SDG acts as a catalyst for fostering the next generation of AI-driven changemakers committed to creating impactful, scalable solutions for a sustainable future.

3 AI-enabled Innovation Advancing SDG11

The joint IRCAI, AI in Africa and GITEX competition served as a global platform for surfacing innovative AI-driven solutions to SDG 11 challenges, bridging the ideas of PhD researchers in North Africa with the entrepreneurial agility of startups worldwide. Among the standout innovations emerging from the competition were AI-powered geospatial mapping systems for monitoring informal settlements, predictive analytics for optimizing urban transport routes in congestion-prone cities, and machine learning models for forecasting waste generation to improve collection efficiency. Several projects addressed climate resilience, including early-warning systems for urban flooding and AI-assisted tools for assessing heat island effects and guiding green space planning. From energy-efficient building design algorithms to citizen engagement platforms that use natural language processing for policy feedback, the competition highlighted the breadth of AI's potential to make cities more sustainable and inclusive. By uniting academic depth with market-ready solutions, the initiative not only identified promising prototypes but also laid the groundwork for scalable interventions adaptable to diverse urban contexts..

ToumAI. A holistic multilingual AI platform designed to bridge the digital divide in Africa by enabling voice-driven customer experiences in low-resource languages, advancing SDG 11. Built on a compound AI structure that saves computing

power compared to foundational LLMs, the system supports speech-to-text, text-to-speech, emotion analysis, churn detection, and predictive insights across African dialects such as Swahili, Amharic, Yoruba, and Darija. By integrating AI-powered voice agents, IVR optimization, and multilingual analytics, ToumAI delivers inclusive, real-time, and cost-effective communication for telco, banking, and transport sectors (see Figure 1). Its innovation lies in industrializing underrepresented African languages for AI applications, ensuring accessibility for populations historically excluded from the AI revolution.

AED EnergyAED. An AI-enabled renewable energy storage system that converts electricity into high-temperature heat (up to 800° C) using salt-based thermal bricks, providing 24/7 clean power and heat without combustion. Unlike batteries or diesel, the system delivers up to 24 hours of dispatchable energy at lower cost, using safe, stable, and modular 10MWh units. Applications include microgrids, telecoms, industrial heat, and desalination, making it particularly suited for regions with unreliable energy supply. By enabling baseload renewable energy, AED Energy strengthens critical infrastructure and advances SDG 11 while reducing dependence on diesel.

SolvesALL Mobility. Delivery district planning and optimization machine learning tools that support smarter urban logistics impacting the sustainable of cities and communities. Its Postal POI system uses algorithms to automatically design delivery districts, balancing workload, reducing overlap, and minimizing travel time. Leveraging GPS trace analysis, stay-point detection, regression models, and crowdsourced field data, the system learns delivery micro-locations, service times, and accessibility factors (e.g., stairs, obstacles). By integrating these AI-driven insights, SolvesAll enables cost savings, operational efficiency, and improved registry accuracy−demonstrated by expected multimillion-euro annual savings for postal operators− while offering scalability to sectors such as waste management and ATM/vending machine logistics.

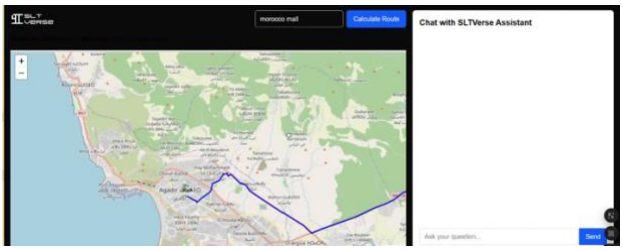


Figure 2: Screenshot of the SLTverse engine, winner at the AI stage of GITEX Africa 2025

SLTverse. This smart city solution introduces an AI-powered travel app that supports SDG 11 by enhancing safety, sustainability, and cultural engagement in tourism. At its core is an AI Route Advisor that leverages structured mobility data—spanning cost, CO_2 emissions, safety, time, and distance—to recommend optimal transport options. This is strengthened by a Retrieval-Augmented Generation (RAG) framework, which combines vector search, large language models, and workflow orchestration to deliver fast, contextual, and multilingual guidance (see screenshot at Figure 2). The system's AI assistant adapts to real-time inputs such as weather, safety alerts, and user preferences, ensuring tailored and secure travel recommendations. Beyond mobility, the platform enriches tourism through VR-based storytelling with avatars narrating site histories, and employs metadata-driven personalization supported by visual analytics (route maps, CO_2 vs. cost comparisons, safety heatmaps). Collectively, these AI innovations position the app as a smart city enabler that aligns sustainability, cultural engagement, and traveler well-being.

SOBEK. A federated AI system for flood resilience that addresses the lack of early-warning systems in rapidly urbanizing African cities. Unlike centralized models, it applies federated learning to collaboratively improve predictions while preserving data privacy and sovereignty. Local nodes train specialized models−LSTMs for weather series, GNNs for hydrological networks, and U-Nets for satellite imagery−using geospatial, meteorological, and historical flood data. Model updates are aggregated with FedAvg and refined through station similarity graphs to capture regional hydrological patterns. Despite challenges of data heterogeneity and low connectivity, Sobek delivers more accurate flood seasonality, year, and magnitude predictions, enabling timely early warnings, urban planning, and disaster resilience across Africa.

Ecoguardians. This initiative introduces an AI-powered system to optimize water-saving advertisements in Morocco, advancing SDG 11 (Sustainable Cities and Communities). By analyzing diverse campaign content (videos, images, text, social media engagement, and survey data), the system identifies what makes ads effective and generates improved variations. It integrates computer vision (CNNs) for visual features, language models (BERT/GPT) for text and sentiment, predictive models (XGBoost/Random Forest) for engagement forecasting, and GANs for generating impactful ad variations. Ethical and data-driven personalization ensures campaigns remain responsible, transparent, and locally relevant. Early prototypes show measurable engagement gains, empowering cities to run evidence-based, AI-enhanced awareness campaigns that strengthen sustainable water use.

4 Conclusions and further work

The integration of AI with the SDGs represents a critical frontier in global innovation, particularly as we confront complex challenges in health, education, climate, and urbanization. The AI4SDG programme, as implemented through the collaboration of IRCAI, AI in Africa, and GITEX, demonstrates a strategic and scalable model for aligning technological advancement with sustainable impact. By combining certified training, research-to-startup pathways, and accelerator programs, AI4SDG empowers diverse stakeholders—from students and researchers to entrepreneurs and SMEs—to develop responsible, ethical and context-sensitive AI solutions across the 17 SDGs.

One of the programme's most significant contributions lies in its ability to bridge the gap between academic research and real-world application, particularly in the Global South. Through its global reach and multi-region engagements, AI4SDG not only promotes responsible AI development but also facilitates access to funding, mentorship, and global markets, thereby amplifying the reach and effectiveness of AI for social good. However, while the AI4SDG11 programme has laid a robust foundation, several

avenues remain open for further development, now open to all SDGs. Future work should focus on:

- **Longitudinal impact assessments** to evaluate the sustainability and real-world outcomes of AI solutions emerging from the programme.
- **Expanded participation** across underrepresented regions and communities, ensuring equitable access to AI training and opportunities.
- **Integration of emerging technologies**, such as neurosymbolic AI, edge AI, and federated learning, into training tracks and solution design.
- **Stronger policy linkages** to influence national and international AI governance frameworks through insights derived from grassroots innovation.
- **Enhanced data infrastructure**, including open datasets aligned with the SDGs, to support more accurate, inclusive, and transparent AI development.

The AI4SDG programme highlights the transformative potential of AI when it is purposefully directed toward sustainable development. As the initiative expands and evolves, it will be crucial to maintain a balance between innovation, ethics, and inclusivity—ensuring that AI becomes not just a tool for growth, but a vehicle for equitable and sustainable global progress. At the same time, it is also important to acknowledge the programme's inherent challenges and limitations. Sustaining long-term participation from diverse stakeholders requires consistent resources, local capacity-building, and incentives that extend beyond initial pilot enthusiasm. Scaling successful pilots into broader, systemic solutions often encounters barriers such as fragmented policy environments, limited infrastructure in low-resource settings, and uneven access to funding. Moreover, as AI solutions transition from competitive innovation contexts to real-world deployment, questions of ethical oversight, data governance, and accountability become increasingly complex—particularly in cross-border collaborations. Addressing these challenges will be essential to ensure that the AI4SDG initiative not only inspires innovation but also establishes durable, ethically grounded impact at scale.

Acknowledgments / Zahvala

This research was partially funded by the European Commission's Horizon research and innovation program under grant agreement 820985 (NAIADES) and 101120237 (ELIAS).

References / Literatura

[1] Gupta, S. and Degbelo, A., (2023) An empirical analysis of AI contributions to sustainable cities (SDG 11). In *The ethics of artificial intelligence for the sustainable development goals* (pp. 461-484). Cham: Springer International Publishing.

[2] Mhlanga, David, and Deo Shao (2025). AI-optimized urban resource management for sustainable smart cities. In *Financial inclusion and sustainable development in sub-saharan Africa*, pp. 96-116. Routledge.

[3] Mohsen, B. M. (2024). *AI-driven optimization of urban logistics in smart cities: Integrating autonomous vehicles and iot for efficient delivery systems*. Sustainability, 16(24), 11265.

[4] Petry, Lisanne, et al. (2021) *Design and results of an AI-based forecasting of air pollutants for smart cities*. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Inform. Sciences 8: 89-96.

[5] Aguilar, J., et al. (2021) *A systematic literature review on the use of artificial intelligence in energy self-management in smart buildings*. Renewable and Sustainable Energy Reviews 151: 111530.

[6] Pita Costa J., Rei L., Bezak N., Mikoš M., Massri M.B., Novalija I. and Leban, G. (2024) Towards improved knowledge about water-related extremes based on news media information captured using artificial intelligence. International Journal of Disaster Risk Reduction, 100, p.104172.

[7] Mustafa Zaouini, Joao Pita Costa, Manal Cherkaoui, Hanaa Hachimi, M. Wahib Abkari, Kamal Gourari, Hatim Lachheb and Jad Tounsi El Azzoiani (2024) Addressing Water Sustainability Challenges in North Africa with Artificial Intelligence In *Proceedings of SIKDD /24*.

[8] IRCAI (2024*) IRCAI Partners with AI in Africa for the AI 4 Water Sustainability Challenge*. Available at: https://ircai.org/inircai-partners-with-ai-in-africa/

Automated First-Reply Generation for IT Support Tickets Using Retrieval-Augmented Generation and Multi-Modal Response Synthesis

Domen Jeršek
domenjersek@gmail.com
Jožef Stefan Institute
Slovenia

Klemen Kenda
klemen.kenda@ijs.si
Jožef Stefan Institute
Slovenia

Rok Klančič
rok.klancic@gmail.com
Jožef Stefan Institute
Slovenia

Matteo Frattini
Matteo.Frattini@gft.com
GFT Italia
Italy

Abstract

IT support organizations require timely and consistent first responses to incoming support tickets. This paper presents a Retrieval Augmented Generation system for automatic generation of contextually appropriate first replies. The approach combines semantic similarity search with multi-modal response synthesis, retrieving similar resolved tickets using sentence embeddings and FAISS indexing. Response-type detection determines whether structured templates or personalized conversational replies are most suitable for each request. The system incorporates temporal context detection for status updates and employs few-shot prompting with selected examples to maintain organizational communication standards. Evaluation using semantic similarity metrics demonstrates the system's ability to generate replies that closely match human-written responses across various ticket types, providing a practical solution for reducing response times while maintaining quality and consistency.

Keywords

IT support, retrieval-augmented generation, automated response generation, natural language processing, semantic similarity

1 Introduction

IT support organizations face increasing volumes of support tickets that require timely and consistent issue resolution, starting from the first response. Manual processing creates bottlenecks that delay user support and increases operational costs, while the quality and consistency of first replies varies significantly between support agents, leading to inconsistent user experiences.

The primary challenge lies in generating contextually appropriate first replies that match organizational communication standards while addressing the specific nature of each support request. Support tickets exhibit diverse characteristics: some require structured template responses with specific form fields, while others benefit from personalized conversational replies that acknowledge the user's specific situation.

Traditional automated response systems relied on template-based approaches and rule-based classification [2], which provided consistent but inflexible responses that failed to capture nuanced requirements. Recent advances in natural language processing have enabled more sophisticated approaches using transformer architectures [11] and pre-trained models like BERT [1]. Retrieval-based systems identify similar historical cases and adapt previous responses [5], while retrieval-augmented generation (RAG) [6] combines parametric knowledge in language models with retrieval from external knowledge bases for knowledge-intensive tasks.

However, retrieval systems may struggle with novel scenarios, and purely generative approaches face challenges in maintaining organizational consistency. Hybrid approaches attempt to balance flexibility with reliability [3], while response classification has evolved from traditional feature engineering to transformer-based models [9].

Our research addresses these limitations by developing an automated first-reply generation system that combines retrieval-augmented generation with multi-modal response synthesis. The system distinguishes between different response types, maintains organizational communication standards, and generates contextually relevant replies through response-type detection, temporal context awareness, and few-shot prompting with carefully selected examples.

2 Data

Our dataset consists of 1,847 IT support tickets containing ticket titles, descriptions, and complete communication logs. Each ticket includes the full conversation history between users and support agents, from initial submission through resolution.

The dataset exhibits significant diversity in ticket types, including software installation requests, access rights management, hardware support, VPN configuration, employee onboarding and offboarding, and system outage reports. Communication logs contain multiple exchanges, requiring careful extraction of first replies from the complete conversation history.

We developed a specialized extraction algorithm to isolate the initial support agent response from the multi-turn conversation logs. The extraction process identifies timestamp patterns and user information markers to separate individual responses. The cleaning heuristics systematically remove formatting artifacts including: (1) leading and trailing dash sequences, (2) formal greeting patterns like "Dear Name,", (3) separator lines containing five or more consecutive dashes, (4) user identification lines

Information Society 2025, Ljubljana, Slovenia

https://doi.org/https://doi.org/10.70314/is.2025.sikdd.19

with parenthetical ID patterns, and (5) responses shorter than 50 characters to filter noise. The algorithm ensures only substantial first replies are retained by validating minimum content length.

After preprocessing, 1,466 tickets contained valid first replies suitable for training and evaluation. The first replies range from 50 to 2,000 characters in length, with an average length of 387 characters. Response types include structured template responses (42%) containing form fields and specific requirements, personalized conversational responses (38%) addressing individual user situations, and status update communications (20%) providing incident or outage information. Response types were automatically classified using keyword-based heuristics and regular expression patterns, as described in Section "3.3 Response Type Detection".

The dataset was split using stratified random sampling with a fixed seed (random_state=42) to ensure reproducibility. Eighty tickets were randomly selected for the test set, representing approximately 5.5% of the processed dataset, with the remaining 1,386 tickets forming the knowledge base for retrieval. The test set maintains proportional representation across all response types: 34 template responses (42.5%), 30 personalized responses (37.5%), and 16 status updates (20%), closely matching the overall dataset distribution. This stratified approach ensures evaluation coverage across diverse ticket categories while preventing data leakage between training and test sets. This was repeated several times to ensure the selected test sets are representative of the entire dataset.

3 Methodology

Our system implements a multi-stage pipeline for automated first-reply generation, combining semantic retrieval, response-type detection, and few-shot prompting. Figure 1 illustrates the complete system architecture, showing the flow from input ticket processing through knowledge base retrieval to final response generation.

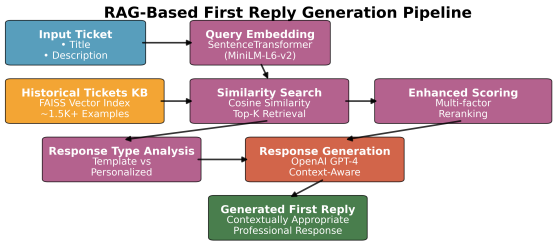


Figure 1: System Architecture: The complete RAG pipeline for automated first-reply generation, showing the eight-stage process from ticket input through embedding generation, knowledge base search, enhanced scoring, response type detection, and final reply generation using GPT-4.

3.1 Knowledge Base Construction

We construct a knowledge base from historical tickets using sentence embeddings [8]. Each ticket is represented by title and description embeddings computed using the all-MiniLM-L6-v2 sentence transformer model [12], which provides a compact 384-dimensional representation optimized for semantic similarity tasks. We build separate embeddings for titles and descriptions, plus combined embeddings for comprehensive similarity search, enabling multi-granular matching across different text components.

The embeddings are indexed using FAISS (Facebook AI Similarity Search) [4] for efficient retrieval with approximate nearest neighbor search. We normalize embeddings using L2 normalization and employ inner product similarity for fast retrieval, achieving sub-linear search complexity through hierarchical clustering and inverted file structures. Figure 2 provides a conceptual visualization of how tickets are positioned in the semantic embedding space based on their content similarity.

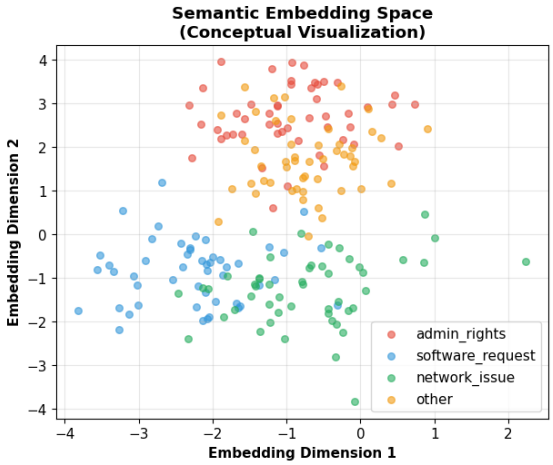


Figure 2: Semantic Embedding Space: Conceptual visualization of how support tickets are distributed in the high-dimensional embedding space, where semantically similar tickets cluster together, enabling effective retrieval of relevant historical examples.

3.2 Retrieval System

For each incoming ticket, we retrieve similar historical cases using a multi-factor scoring approach that combines semantic similarity with categorical and structural matching. The enhanced retrieval score combines:

- Base semantic similarity (50%) from FAISS cosine similarity using normalized embeddings
- Category match bonus (20%) when ticket types align, using exact string matching
- Title similarity (15%) using dedicated title embeddings with cosine similarity
- Description similarity (10%) using dedicated description embeddings with cosine similarity
- Response quality bonus (5%) based on response structure analysis and content completeness metrics

These weights reflect the relative importance of semantic similarity, categorical alignment, and structural relevance in ensuring that retrieved examples are both contextually appropriate and organizationally consistent. We retrieve a larger candidate set (4× the target number) from the FAISS index and apply this multi-factor re-ranking to select the most relevant examples, ensuring both semantic relevance and categorical appropriateness.

3.3 Response Type Detection

We implement response-type detection using keyword-based heuristics with regular expression patterns to classify responses as template-based, personalized, or status updates. Template responses are identified by structured formatting indicators such as

form field markers (e.g., "Field:", "Value:"), bullet point patterns, numbered lists, and specific organizational phrases like "Below you will find the additional form information."

Personalized responses are characterized by conversational elements including direct questions, user-specific acknowledgments (e.g., "Thank you for contacting us"), empathy expressions, and conditional statements. Status updates contain temporal references using datetime patterns, incident identification numbers, system status keywords, and global communication patterns following organizational incident response protocols.

3.4 Few-Shot Prompting

Response generation employs few-shot prompting with GPT-4 [7], using retrieved examples to guide generation through in-context learning. We construct structured prompts that include:

- Current ticket information (title, description, detected response type).
- 4-5 most relevant historical examples with their corresponding responses.
- Response type-specific instructions (template vs. personalized formatting).
- Organizational communication guidelines and tone specifications.

Template responses receive strict formatting instructions with explicit field markers and structural constraints to maintain exact organizational formatting, while personalized responses are guided toward conversational but professional tone with specific phrase patterns and acknowledgment structures.

3.5 Temporal Context Detection

We implement temporal context detection using compiled regular expressions to identify tickets related to system outages, status updates, or global communications. The detection system uses pattern matching for temporal indicators (e.g., "since", "until", "during"), incident terminology ("outage", "maintenance", "downtime"), and organizational communication markers ("all users", "system-wide", "scheduled maintenance"). Detected temporal contexts trigger specialized status update generation that mirrors organizational incident communication patterns, including severity levels, expected resolution times, and escalation procedures.

4 Results

We evaluate our system using semantic similarity metrics and response quality assessments across 80 test tickets representing diverse support scenarios.

4.1 Similarity Metrics

We employ two sentence transformer models for comprehensive evaluation [8]:

- all-MiniLM-L6-v2 [12]: Lightweight 384-dimensional model optimized for general semantic similarity with 22.7M parameters
- all-mpnet-base-v2 [10]: Higher-capacity 768-dimensional model with 109M parameters for nuanced similarity assessment using masked and permuted pre-training

The selection of these two models provides complementary evaluation perspectives. all-MiniLM-L6-v2 serves as the primary embedding model in our RAG system due to its computational efficiency and proven effectiveness in semantic similarity tasks, making it suitable for real-time ticket processing. all-mpnet-base-v2 offers higher representational capacity through its bidirectional encoder architecture and serves as a more sophisticated evaluation metric, providing additional validation of semantic coherence through its enhanced understanding of contextual relationships and nuanced text representations.

Our system achieves an average MiniLM similarity of 0.7841 and MPNet similarity of 0.8048 between generated and expected responses. These scores indicate strong semantic alignment with human-written replies, confirmed through cross-validation analysis showing confidence intervals within a 3% range (±2.9% for MiniLM similarity). Figure 3 shows the performance variation across different test tickets, demonstrating consistent quality across diverse support scenarios.

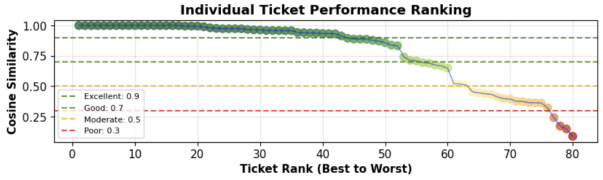


Figure 3: Individual Ticket Performance: Semantic similarity scores (MiniLM) for each test ticket, showing consistent performance across diverse support scenarios with most tickets achieving similarity scores above 0.7.

4.2 Response Quality Analysis

Quality assessment reveals that 55 out of 80 generated responses (68.8%) achieve similarity scores above 0.7, indicating high semantic alignment. The system successfully maintains organizational communication standards while addressing specific user requirements. Figure 4 illustrates the distribution of response quality scores across the evaluation dataset.

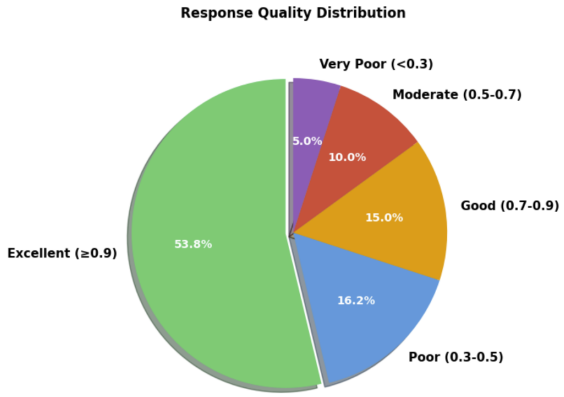


Figure 4: Response Quality Distribution: Distribution of semantic similarity scores showing that 68.8% of generated responses achieve scores above 0.7, indicating strong semantic alignment with expected human-written replies.

Template responses demonstrate particularly strong performance, with exact structural matching and appropriate placeholder handling. Personalized responses achieve good contextual relevance while maintaining professional tone.

4.3 Response Type Distribution

The system correctly identifies response types in 87% of cases, routing requests to appropriate generation strategies. Template detection achieves 90% accuracy, while personalized response detection reaches 85% accuracy.

Temporal context detection successfully identifies 100% of status update scenarios on the tested examples, enabling appropriate global communication style responses.

The plot of the length of the generated responses against the expected responses further supports these results. Figure 5 demonstrates that generated responses maintain appropriate length characteristics compared to human-written replies, with strong correlation between generated and expected response lengths.

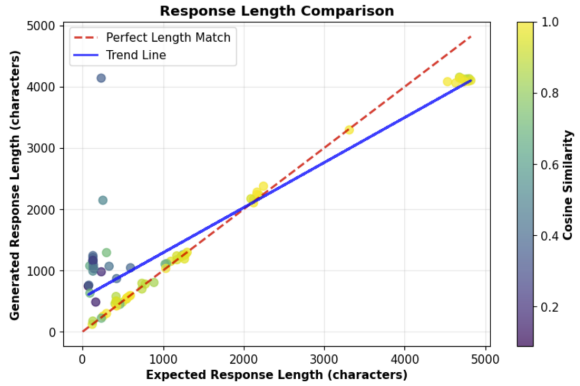


Figure 5: Response Length Comparison: Scatter plot comparing the length of generated responses versus expected responses, showing strong correlation and indicating that the system generates appropriately sized replies consistent with human writing patterns.

4.4 Error Analysis

Remaining challenges include handling of highly specialized technical scenarios and tickets requiring complex multi-step procedures. Some responses exhibit placeholder artifacts when exact matching fails, and very short or very long responses occasionally deviate from expected patterns.

The system shows consistent performance across different ticket categories, with minor variations in quality for edge cases involving complex technical requirements or unusual organizational procedures.

5 Conclusion

This paper presents a comprehensive approach to automated first-reply generation for IT support tickets using retrieval-augmented generation and multi-modal response synthesis. Our system successfully combines semantic similarity search, response-type detection, and few-shot prompting to generate contextually appropriate replies that closely match human-written responses.

The evaluation demonstrates strong performance across diverse ticket types, achieving semantic similarity scores of 0.78-0.80 and maintaining organizational communication standards. Cross-validation analysis confirms the stability of these results, with performance metrics varying within a ±3% range, indicating robust and reliable performance across different evaluation scenarios. The system provides a practical solution for reducing response times while ensuring quality and consistency in IT support communications.

Future work will explore improving template handling using instruction-tuned large language models and developing fine-tuned classifiers for more accurate response type detection, enabling more structured and context aware reply generation.

References

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: pre-training of deep bidirectional transformers for language understanding, 4171–4186. doi:10.18653/v1/N19-1423.

[2] Yixin Diao, Hani Jamjoom, and Zhen-Yu Shae. 2009. Rule-based problem classification in it service management. In *2009 IEEE International Conference on Services Computing*. IEEE, 221–228. doi:10.1109/SCC.2009.31.

[3] Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. Retrieval augmented language model pre-training. *arXiv preprint arXiv:2002.08909*. https://arxiv.org/abs/2002.08909.

[4] Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7, 3, 535–547. doi:10.1109/TBDATA.2019.2921572.

[5] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering, 6769–6781. doi:10.18653/v1/2020.emnlp-main.550.

[6] Patrick Lewis et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33, 9459–9474. https://arxiv.org/abs/2005.11401.

[7] OpenAI. 2023. Gpt-4 technical report. (2023). https://arxiv.org/abs/2303.08774 arXiv: 2303.08774 [cs.CL].

[8] Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, 3982–3992. doi:10.18653/v1/D19-1410.

[9] Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2018. A primer on neural network models for natural language processing. *Journal of Artificial Intelligence Research*, 61, 65–95. https://arxiv.org/abs/1510.00726.

[10] Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: masked and permuted pre-training for language understanding. In *Advances in Neural Information Processing Systems*. Vol. 33. Curran Associates, Inc., 16857–16867. https://proceedings.neurips.cc/paper/2020/hash/c3a690be93aa602ee2dc0ccab5b7b67e-Abstract.html.

[11] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30. https://arxiv.org/abs/1706.03762.

[12] Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: deep self-attention distillation for task-agnostic compression of pre-trained transformers. In *Advances in Neural Information Processing Systems*. Vol. 33. Curran Associates, Inc., 5776–5788. https://proceedings.neurips.cc/paper/2020/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

A Machine-Learning Approach to Predicting the Pronunciation of Pre-Consonant *l* in Standard Slovene

Jaka Čibej
jaka.cibej@ff.uni-lj.si
Centre for Language Resources and Technologies & Faculty of Arts, University of Ljubljana
Jožef Stefan Institute
Ljubljana, Slovenia

Abstract

The pronunciation of pre-consonant *l* in Slovene words (e.g. *alge*, *polž*, *gledalka*) is not easily predictable (/l/, /u̯/, or both) and poses a problem for the otherwise effective rule-based grapheme-to-phoneme conversion. We present a method to discriminate between the various pronunciations of pre-consonant *l* using machine-learning models trained on vectors of character-level *n*-gram features from approximately 153,500 manually annotated Slovene words with pre-consonant *l* from the *ILS 1.0* dataset. We achieve an accuracy of 86% (over a majority baseline of 76.53%) and conclude the paper with potential steps for future work.

Keywords

pronunciation, grapheme-to-phoneme conversion, pre-consonant *l*, pronunciation ambiguity, Slovene

1 Introduction

In languages that are characterized by greater orthographic depth (i.e., a greater discrepancy between the written form and its pronunciation), such as English or French, grapheme-to-phoneme (G2P) conversion requires more sophisticated methods such as neural networks (see e.g. [10] for French and [14] for English). Slovene, on the other hand, features a much more transparent orthography ([15]; [17]). Phonetic transcriptions of Slovene words – with some exceptions, such as acronyms, symbols, numerals, and certain words of foreign origin (e.g. *sommelier*), including proper nouns (e.g., *Johnson*; more on this in [3]) – can be very reliably generated using a rule-based approach, especially if taking the accentuated form (e.g., *drevó* instead of the unaccentuated *drevo*) as the starting point, as the diacritic disambiguates the position of the accent and the manner of pronunciation of the accentuated vowel grapheme. The *Slovene IPA/X-SAMPA G2P Converter*[1] achieves an accuracy of approximately 98% (based on an evaluation on a stratified sample of words; see [2]).

However, there are several exceptions (in addition to the ones already mentioned) in which the pronunciation of certain graphemes is much more difficult to predict with rules. We focus on one such problem in this paper: the pronunciation of pre-consonant *l* in Slovene words. The grapheme *l*, when preceding a consonant grapheme, can be pronounced as either /l/ or /u̯/. In some cases, both variants are acceptable. Examples include words such as *alge* ('algae', IPA: /ˈaːlgɛ/, but never */ˈaːu̯gɛ/), *polž* ('snail', IPA: /ˈpɔːu̯ʃ/, but never */ˈpɔːlʃ/), *gledalka* ('spectator (female)', IPA: /glɛˈdaːu̯ka/ or /glɛˈdaːlka/), and *decimalka* ('decimal number', IPA: /dɛciˈmaːlka/, but never */dɛciˈmaːu̯ka/). The reasons for these different pronunciations are historic and etymological in some cases, while in others, the difference cannot be easily explained and has more to do with conventions in language use. The issue of pre-consonant *l* has been tackled by Slovene linguistics for more than a century (see [4] for a brief overview). Perception tests and small-scale surveys ([16]; [11]) have recently been conducted to collect data for lexicographic resources (such as the *Slovenian Normative Guide 8.0*),[2] but empirical data remains scarce: relevant language resources are not machine-readable or openly accessible (as is the case of the *Dictionary of Slovenian Literary Language*[3]) or contain inconsistent data (e.g., *OptiLeX* [19]). In this paper, we use the recently published *ILS 1.0* dataset ([1]; described in Section 2).

Because the *Slovene IPA/X-SAMPA G2P Converter* is currently entirely rule-based, all pre-consonant *l* graphemes are transcribed as /l/, resulting in errors that need manual corrections when compiling language resources. Our goal is to implement a machine-learning approach[4] to disambiguate between different pronunciations. Increasing the accuracy of the converter is important in the context of the automatic compilation of modern lexicographic resources that can also be used as machine-readable databases for training models (including large language models) and improving speech recognition and speech synthesis for Slovene. We describe the dataset (Section 2), the statistical analysis used for feature selection (Section 3), the results (Section 4), and several steps for future work (Section 5).

2 Dataset

ILS 1.0 ([1]; described in more detail in [4]) is a dataset of approx. 173,400 inflected Slovene word forms (of approx. 6,000 Slovene lexemes) containing a single pre-consonant *l* grapheme. Each occurrence of pre-consonant *l* was annotated for its pronunciation by 5 linguists (2 annotations per occurrence). The word forms were extracted from the manually validated lexemes of *Sloleks 3.0* [5], the largest open-access dataset with machine-readable morphosyntactic information on Slovene words. Table 1 shows the distribution of word forms by agreement: in 89% of word

[1] The *Slovene IPA/X-SAMPA G2P Converter* is part of *Pregibalnik*, a custom tool that was developed for the expansion of the *Sloleks Morphological Lexicon of Slovene* [5], which is the morphological basis for the *Digital Dictionary Database of Slovene* [8]. *Pregibalnik* is available as open-access code at https://github.com/clarinsi/SloInflector and as an API service at https://orodja.cjvt.si/pregibalnik/docs; the *Slovene IPA/X-SAMPA G2P Converter* is also available as an API at https://orodja.cjvt.si/pregibalnik/g2p/docs.

Information Society 2025, Ljubljana, Slovenia

https://doi.org/10.70314/is.2025.sikdd.1

[2] *Pravopis 8.0* (*Slovenian Normative Guide 8.0*): https://pravopis8.fran.si/
[3] The *Dictionary of Slovenian Literary Language* (SSKJ) is available at https://fran.si/.
[4] An attempt at using machine learning for Slovene phonetic transcriptions was made by [9]; however, the method was evaluated on the *Sloleks Morphological Lexicon of Slovene 3.0* [5], where the issue of pre-consonant *l* is still unresolved.

Table 1: Word forms in *ILS 1.0* by agreement.

Pronunciation	Number of Forms	%
/l/	117,459	67.73
/u̯/	23,884	13.77
Both	12,160	7.01
Both | /l/	11,205	6.46
Both | /u̯/	7,051	4.07
/l/ | /u̯	1,660	0.96
Total	**173,419**	**100.00**

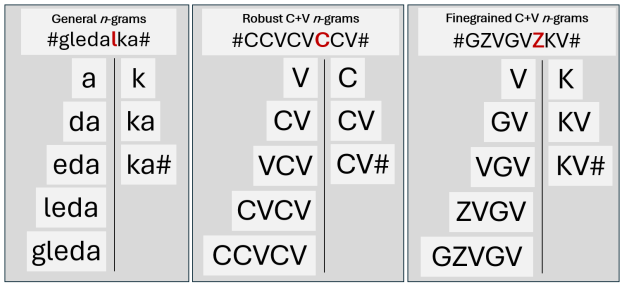


Figure 1: Extraction of character-level *n*-gram features for the pre-consonant *l* in the word *gledalka*.

forms (highlighted in gray), the annotators agree on the pronunciation of pre-consonant *l*. They disagree in 11% of the examples, with one annotator allowing for both pronunciation variants and the other allowing for only one pronunciation. Complete disagreement is present only in less than 1% of the examples.

We use the 153,503 forms with complete agreement as training data for machine-learning models as described in the following sections. It should be noted, however, that while *ILS 1.0* is the largest open-access dataset on pre-consonant *l* pronunciations, it is not completely representative of language use in general (with annotations by only 5 linguists with a background in translation and Slovene studies; these can be biased towards linguistic rules that might not reflect real language use). Despite this, the dataset is robust enough to help disambiguate the more obvious examples (such as *alge*, IPA: /ˈaːlgɛ/, and *polž*, IPA: /ˈpɔːu̯ʃ/).

3 Feature Selection

To some extent, the pronunciation of pre-consonant *l* depends on the preceding and subsequent graphemes,[5] so we use character-level *n*-grams as features for prediction. For each pre-consonant *l* in each word form, we identify the *n*-grams ($1 \leq n \leq 5$) in its direct left/right surroundings as shown in Figure 1 (see footnote 6). We include word boundary markers (#) to discriminate between word-initial and word-final *n*-grams. We also perform the same extraction on robust and finegrained C+V representations of each word form.[6]

[5]The *Slovenian Normative Guide 8.0* (*Pravopis 8.0*, see https://pravopis8.fran.si), for instance, states that a pre-consonant *l* preceded by the grapheme *o* is often characterized by the /u̯/ pronunciation; this is true of words that historically used the syllabic *l* (e.g. *polh* IPA: /ˈpɔːu̯x/ 'dormouse'; *volk* IPA: /ˈvɔːu̯k/ 'wolf'). However, there are exceptions as not all *ol* *n*-grams originate from the syllabic *l* (e.g., *polkovnik* IPA: /pɔlˈkɔːu̯nik/ 'colonel'; *voltaža* IPA: /vɔlˈtaːʒa/ 'voltage').

[6]In the robust C+V form, all consonant graphemes are substituted with *C* and all vowel graphemes with V. In the finegrained C+V form, consonant graphemes were generalized into more finegrained categories, e.g. graphemes denoting Slovene sonorants (M), voiced (G) and voiceless obstruents (K), foreign consonants (X), etc.

Table 2: Contingency table for the general *n*-gram *c* when following a pre-consonant *l*.

Pronunciation → ↓ Presence	/l/	/u̯/	/l/+/u̯/
Yes	2,653	1,847	5,980
No	114,898	22,045	6,180

Table 3: A sample of statistically significant general character-level *n*-grams.

n-Gram	χ^2	p	V	$r_{\|max\|}$	Category
c	38,199.59	****	0.499	178.81, /l/, No	post-*l*
n	29,081.52	****	0.435	79.27, /l/, No	post-*l*
ce	16,003.46	****	0.323	118.30, /l/, No	post-*l*
o	77,025.17	****	0.708	227.83, /l/, No	pre-*l*
po	48,241.29	****	0.560	193.98, /l/, No	pre-*l*
a	16,592.50	****	0.329	-79.85, /l/, No	pre-*l*

We extract a total of 8,082 different general *n*-grams (consisting of actual graphemes; 3,041 in pre-*l* position, 5,541 in post-*l* position), 116 different robust C+V *n*-grams (65 pre- and 51 post-*l*), and 603 different finegrained C+V *n*-grams (262 pre- and 341 post-*l*). For each *n*-gram, we compile a contingency table. For instance, Table 2 shows the occurrences of the general *n*-gram *c* in the position directly following a pre-consonant *l* (e.g., *morilca*, 'murderer', masculine common noun, genitive singular form) depending on the pronunciation of the pre-consonant *l*.

In order to determine statistically significant features that help discriminate between different pronunciations, we performed a series of Pearson's χ^2 tests [12] and corrected for family-wise error rate with the Holm-Bonferroni method [7]. We calculated Cramér's V [6] as the measure of effect size.[7] This resulted in a total of 4,263 statistically significant features (1,856 pre-*l* general and 1,794 post-*l* general *n*-grams; 60 pre-*l* and 40 post-*l* robust C+V *n*-grams; 242 pre-*l* and 271 post-*l* finegrained C+V *n*-grams). Several statistically significant pre-*l* general *n*-grams are shown in Table 3.[8] The table shows the values of the χ^2 statistic and Cramér's V, the p-value representations, the maximum absolute value of Pearson's residuals (and its position in the contingency table), and the category of the *n*-gram (post-*l* or pre-*l*). With the exception of the *a* *n*-gram, which is more indicative of the /l/ pronunciation, the others indicate one of the other two options (/u̯/; or /l/+/u̯/). The results also confirm the statement found in the *Slovenian Normative Guide 8.0* that the *o* grapheme in pre-*l* position is strongly indicative of the /u̯/ pronunciation.

4 Prediction and Evaluation

We compiled a custom vectorizer based on the identified features. The vectorizer scans each input word form (along with its Multext-East v6 morphosyntactic tag[9]) for all occurrences of

[7]We calculate Cramér's V as $\sqrt{\frac{\chi^2}{N * d_{min}}}$, where χ^2 is the Pearson's χ^2 statistic, N is the total sample size, and d_{min} is the minimum dimension of the contingency table.

[8]For all tests, the degrees of freedom (df) were equal to 2 and the total sample size (N) was equal to 153,603. The p-values should be interpreted in the following manner: **** → $p \leq 0.0001$; *** → $p \leq 0.001$; ** → $p \leq 0.01$; * → $p < 0.05$

[9]Multext-East v6 Morphosyntactic specifications: https://nl.ijs.si/ME/V6/msd/html/msd-sl.html

Table 4: Model performance based on 10-fold cross-validation.

Model	A	BA	P	R	F1
LinearSVC	**86.08**	72.39	**69.26**	55.39	61.54
Multin. NB	77.29	69.54	33.33	**81.84**	47.36
kNN (k=5)	85.91	**73.30**	64.11	62.98	**63.53**
Majority	76.53	-	-	-	-

pre-consonant *l*, extracts the surrounding *n*-grams, converts the morphosyntactic tag into 146 morphosyntactic features, and represents the occurrence as a 4,409-dimensional vector of {0,1} values (with 0 and 1 indicating the absence or presence, respectively, of the *n*-gram in the direct surroundings of the pre-consonant /l/). We compile a total of 153,503 vectors in this way and use the *scikit-learn* Python library [13] to train several models for a classification task with three classes: the goal is to correctly predict whether a pre-consonant *l* is pronounced as /l/, /u̯/, or both.

4.1 Automatic Evaluation

We trained three different models: a Linear Support Vector Classifier (LinearSVC), a Multinomial Naïve Bayes Classifier (Multin. NB), and a *k* Nearest Neighbors Classifier (kNN) and evaluate their performance with a 10-fold cross-validation (with a stratified random test set of word forms). The results are shown in Table 4.[10] The worst performing model is the Multin. NB classifier, which barely achieves an above-baseline accuracy and a very low F1-score compared to the other two classifiers, although its recall is much higher. In terms of balanced accuracy and F1-score, the best model is the kNN classifier. However, it seems that the algorithm is not the most suited for this type of data. It performs similarly to the LinearSVC classifier, but if we compare the sizes of the resulting models, it becomes apparent that the LinearSVC model is much more efficient (with a size of approximately 100 kB) compared to the kNN model, which is overly inflated (with a size of more than 2 GB), possibly indicating overfitting.[11]

Because the LinearSVC model is the most viable, we analyze its performance in more detail. Table 5 shows the confusion matrix for the classifications of the LinearSVC model on a stratified test set (20% of the total 153,503 dataset instances). The model seems to lean more towards the most frequent category (/l/) in its predictions, with approximately 30% of /u̯/ and /l/+/u̯/ instances being misclassified as /l/, whereas 94% of the /l/ instances are classified correctly. It seems that instances allowing both pronunciations are very rarely misclassified as /u̯/ (only 1%). It should also be noted that the instances of /l/+/u̯/ misclassified as either /u̯/ or /l/ are not entirely incorrect, just incomplete. Compared to the rule-based approach (which classifies everything as /l/), the model performs quite well in terms of /l/+/u̯/ and /u̯/ instances and sacrifices only 6% of its accuracy for /l/ instances. In order to determine any future improvements to the model, we analyze some of the misclassified examples in more detail in Section 4.2.

[10] A, BA, P, R, and F1 refer to accuracy, balanced accuracy, macro-precision, macro-recall and macro-F1, respectively.

[11] We also ran a 10-fold cross-validation using only *n*-gram features (no morphosyntax). The performance of the models was slightly worse, e.g. for LinearSVC: A = 85.05, BA = 69.14, P = 68.94, R = 46.85, F1 = 55.76.

Table 5: Confusion matrix for the Linear Support Vector classifier.

True → ↓ Predicted	/l/	/u̯/	/l/+/u̯/	∑
/l/	**22,006**	1,495	729	24,230
/u̯/	1,071	**2,764**	31	3,866
/l/+/u̯/	434	519	**1,672**	2,625
∑	23,511	4,778	2,432	-

4.2 Manual Evaluation

We performed a manual analysis of the misclassified examples to determine whether there are any patterns to the errors that could be help further improve the model with additional features. Due to space limitations, we only focus on the most obvious problems in this paper.

In the examples where the /l/ pronunciation was misclassified as /u̯/, many words contain a pre-consonant *l* followed by the grapheme *d* (*kaldera* 'caldera', *buldožerski* 'pertaining to a bulldozer', *heraldičen* 'heraldic', *bodibilder* 'bodybuilder'). The majority of these examples are pronounced with /l/, with the exception of words like *dopoldne* 'late morning', *popoldanski* 'pertaining to the afternoon', where the pre-consonant *l* is preceded by an *o* grapheme. This could indicate that an additional *n*-gram feature should be added (the *l* along with its preceding and subsequent graphemes: *old*, *ald*, etc.). This could resolve some other misclassifications, such as *impulziven* 'impulsive' and *pulzirajoč* 'pulsating', where words with the *ulz* combination are never pronounced as /u̯/, but words with *olz* are (e.g., *polzeti* 'to slip'). The emergence of such patterns in the misclassifications is a good sign that the classifiers might benefit from a joint pre-*l*/post-*l* feature. This will be explored in future versions.

Many of the instances in which the /u̯/ was misclassified as /l/ contain compound words with the element *pol* 'semi, half': *polnag* 'half-naked', *polfinale* 'semi-final', *polpuščava* 'semi-desert'. Because the element *pol* is always pronounced with /u̯/, this is also true of derived compound words. However, the *n*-gram features used offer no indication of morpheme boundaries, so these misclassifications can be expected.

Additional *n*-gram features could be extracted from the accentuated forms of words. In some examples, the accentuation diacritic can disambiguate the pronunciation of the subsequent pre-consonant *l*. For instance, *dólnji* 'pertaining to something that is downwards or downstream' and *prestólničen* are pronounced with /l/, whereas *tôlšča* 'blubber' and *pôlhográjski* 'pertaining to the town of Polhov Gradec' are pronounced with /u̯/. However, accentuation is rarely written in Slovene and is much more difficult to assign automatically compared to morphosyntactic features. Relying on too many features that are not easily extractable would make the model less robust (more on this in Section 5).

5 Conclusion

We presented a machine-learning approach to improve the accuracy of phonetic transcriptions of Slovene words that contain the ambiguous pre-consonant *l*. While the method does improve accuracy (86% over a majority baseline of cca. 76%) by using very simple character-level *n*-gram and morphosyntactic features, it does not resolve the problem entirely. Aside from several exceptions in language use which are difficult to predict (e.g. *gasilci*,

čistilka; both pronounced with /l/ even though the majority of words ending with *-ilec* and *-ilka* in the dataset can be pronounced with either /l/ or /u̯/), the analysis of misclassified examples has shown several potential future steps that can be implemented to further improve the performance of the models. First, several additional features should be tested. Some of the features are simple, such as word length or number of syllables in word (which could potentially help to correctly classify words such as *volk* and *polh*; short words where the pre-consonant *l* is pronounced as /u̯/). The relative position of the pre-consonant *l* in the word could also potentially be helpful. Several more complex features could also be added, such as word formation relations and morpheme boundaries to help disambiguate, for instance, *decimal-ka* 'decimal number', which is derived from the adjective *decimalen* 'pertaining to decimal numbers' and is pronounced with /*l*/; and *mor-ilka* 'murderer (feminine)', which is derived from the verb *moriti* 'to murder' and can be pronounced as either /l/ or /u̯/). Taking into account the accentuated form of the word could also help: for instance, the *ôl* accentuation – *vôlk* 'wolf', *pôlh* 'dormouse' – indicates the /u̯/ pronunciation, while the *ól* accentuation is indicative of the /l/ pronunciation, e.g. *pólka* 'polka'). However, more complex features cannot be extracted from the word form itself, so making the model too heavily reliant on external linguistic knowledge would sacrifice its robustness and usefulness for unseen words. We will explore these options in our future work but we will first focus on the simplest features to determine the upper boundary of accuracy that can be achieved based solely on the word form and its morphosyntactic features. We will perform additional statistical analyses on *n*-grams containing the pre-consonant *l* as well, and once the optimal model is achieved, it will also be evaluated on previously unseen words containing the pre-consonant *l* that have not been included in the *ILS 1.0* dataset. The results will hopefully also provide more interesting material for further linguistic analyses (such as exceptions to the rules).

As already mentioned, the *ILS 1.0* dataset does not necessarily accurately reflect the linguistic landscape of pre-consonant *l* pronunciation in Slovene words, and more annotations along with perceptive tests and surveys are required. The pronunciations will be manually validated as part of the work on the *Digital Dictionary Database of Slovene* [8], the largest machine-readable open-access database of Slovene linguistic and lexicographic data. The pronunciations will also be cross-referenced with the recordings from the *GOS Corpus of Spoken Slovene* [18], which contains real recordings of Slovene speech and can contribute towards a more accurate distribution of different pronunciations for individual lexemes (e.g., how many occurrences of /glɛˈdaːu̯ka/ or /glɛˈdaːlka/), along with any potential relevant metadata (for instance, whether the pronunciation depends on the region the speaker originates from). The models can then be re-trained on new data and further improved to better reflect real language use.

The models will be implemented into the *Slovene IPA/X-SAMPA Grapheme-to-Phoneme Converter* as part of the *Pregibalnik* tool for automatic Slovene lexicon expansion, which is available under a Creative Commons BY-SA 4.0 license.[12]

[12]The best-performing LinearSVC model (and the accompanying code) for the prediction of pre-consonant *l* pronunciation is available on Github: https://github.com/jakacibej/sikdd2025_predicting_preconsonant_l

Acknowledgements

The research presented in this paper was carried out within the research project titled *Basic Research for the Development of Spoken Language Resources and Speech Technologies for the Slovenian Language* (J7-4642), the research programme *Language Resources and Technologies for Slovene* (P6-0411), and the *CLARIN.SI Research Infrastructure* (I0-E004), all funded by the Slovenian Research and Innovation Agency (ARIS). The author also thanks the anonymous reviewers for their constructive comments.

References

[1] Jaka Čibej. 2024. Dataset of annotated slovene words with pre-consonant l ILS 1.0. Slovenian language resource repository CLARIN.SI. (2024). http://hdl.handle.net/11356/2025.

[2] Jaka Čibej. 2023. Leksikon besednih oblik sloleks. poročilo projekta razvoj slovenščine v digitalnem okolju aktivnost ds1.3. Development of Slovene in a Digital Environment. (2023). https://www.cjvt.si/rsdo/wp-content/uploads/sites/18/2023/06/RSDO_Kazalnik_Sloleks_v2.pdf.

[3] Jaka Čibej. 2024. Predicting pronunciation types in the sloleks morphological lexicon of slovene. In *Data mining and data warehouses (SiKDD): Information Society (IS) 2024 - proceedings of the 27th International Multiconference: volume C*. Institut „Jožef Stefan", 23–26. https://is.ijs.si/wp-content/uploads/2024/11/IS2024_Volume-C.pdf.

[4] Jaka Čibej. 2025. Statistična analiza izgovora črke l v slovenskem oblikoslovnem leksikonu sloleks. *Jezikoslovni zapiski*, 31, 1, (maj 2025), 37–54. doi:10.3986/JZ.31.1.03.

[5] Jaka Čibej et al. 2022. Morphological lexicon sloleks 3.0. Slovenian language resource repository CLARIN.SI. (2022). http://hdl.handle.net/11356/1745.

[6] Harald Cramér. 1946. *Mathematical Methods of Statistics. Princeton Mathematical Series*. Vol. 9. Princeton University Press.

[7] Sture Holm. 1979. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6, 2, 65–70.

[8] Iztok Kosem, Simon Krek, and Polona Gantar. 2021. Semantic data should no longer exist in isolation: the digital dictionary database of slovenian. In *9th EURALEX International Congress "Lexicography for Inclusion"*, 81–83. https://elex.is/wp-content/uploads/2021/09/Semantic-Data-should-no-longer-exist-in-isolation-the-Digital-Dictionary-Database-of-Slovenian_Kosem-Krek-Gantar_EURALEX2020.pdf.

[9] Janez Križaj, Simon Dobrišek, Aleš Mihelič, and Jerneja Žganec Gros. 2022. Uporaba postopkov strojnega učenja pri samodejni slovenski grafemsko-fonemski pretvorbi. In *Jezikovne tehnologije in digitalna humanistika: zbornik konference 2022*. Inštitut za novejšo zgodovino, 248–251. https://nl.ijs.si/jtdh22/pdf/JTDH2022_Proceedings.pdf.

[10] Xavier Marjou. 2021. Gipfa: generating ipa pronunciation from audio. In *eLex 2021 Conference Proceedings*, 588–597. https://elex.link/elex2021/wp-content/uploads/2021/08/eLex_2021_38_pp588-597.pdf.

[11] Tanja Mirtič. 2019. Glasoslovne raziskave pri pripravi splošnega razlagalnega slovarja. In *Slovenski javni govor in jezikovno-kulturna (samo)zavest*. Znanstvena založba Filozofske fakultete, 81–90. https://centerslo.si/wp-content/uploads/2019/10/Obdobja-38_Mirtic.pdf.

[12] Karl Pearson. 1900. X. on the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 50, 302, 157–175. eprint: https://doi.org/10.1080/14786440009463897. doi:10.1080/14786440009463897.

[13] F. Pedregosa et al. 2011. Scikit-learn: machine learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.

[14] Uwe Reichel, Hartmut R. Pfitzinger, and Horst-Udo Hain. 2008. English grapheme-to-phoneme conversion and evaluation. In *Speech and Language Technology 11*, 159–166. https://www.phonetik.uni-muenchen.de/~reichelu/publications/ReichelPfitzingerHainSASR2008.pdf.

[15] Anja Schüppert, Wilbert Heeringa, Jelena Golubovic, and Charlotte Gooskens. 2017. Write as you speak? a cross-linguistic investigation of orthographic transparency in 16 germanic, romance and slavic languages. English. *From semantics to dialectometry*, 32, 303–313. ISBN: 9781848902305.

[16] Hotimir Tivadar. 2004. Priprava, izvedba in pomen perceptivnih testov za fonetično-fonološke raziskave (na primeru analize fonoloških parov). *Jezik in slovstvo*, 49.2, 2, 17–36. https://ojs.zrc-sazu.si/jz/article/view/14222.

[17] Antal van den Bosch, Alain Content, Walter Daelemans, and Beatrice de Gelder. 1994. Analysing orthographic depth of different languages using data-oriented algorithms. In *Proceedings of the 2nd International Conference on Quantitative Linguistics*.

[18] Darinka Verdonik et al. 2023. Spoken corpus gos 2.1 (transcriptions). Slovenian language resource repository CLARIN.SI. (2023). http://hdl.handle.net/11356/1863.

[19] Jerneja Žganec Gros, Tanja Mirtič, Miroslav Romih, and Kozma Ahačič. 2022. *Slovar izgovarjav OptiLEX*. (1. e-izd. ed.). Založba ZRC. ISBN: 978-961-05-0672-0. https://doi.org/10.3986/9789610506720.

Sequencing News Articles with Large Language Models within Enterprise Risk Management Context

Žiga Debeljak†
Jožef Stefan International Postgraduate School
Ljubljana, Slovenia
ziga.debeljak@mps.si

Dunja Mladenić
Department for Artificial Intelligence,
Jožef Stefan Institute
Ljubljana, Slovenia
dunja.mladenic@ijs.si

Klemen Kenda
Department for Artificial Intelligence,
Jožef Stefan Institute
Ljubljana, Slovenia
klemen.kenda@ijs.si

Abstract

This paper evaluates the capability of Large Language Models (LLMs) to reconstruct event timelines from unstructured news data. This capability is highly relevant for Enterprise Risk Management (ERM) applications, where the reconstruction and forecasting of coherent event trajectories are crucial for identifying, assessing, and predicting emerging risks and analyzing risk scenarios. In this study, we tasked twenty LLMs with chronologically ordering randomly shuffled business news articles for three distinct real-world event chains. To prevent simple date sorting, all explicit date markers were removed from the articles. The experiments were conducted under one unassisted and three assisted scenarios that provided the models with hints for the first, the last, or both the first and the last articles in the sequence. The results reveal a systematic variation in difficulty across the three tasks in addition to significant performance disparities among the models, with Grok 4 (xAI), GPT-5, o3 and o3-pro (all three OpenAI), and Gemini 2.5 Pro (Google) consistently outperforming other models practically across all tasks and prompting scenarios. As expected, prompting assistance with additional information systematically improved accuracy, especially for the models that performed poorly in the unassisted scenario. The high level of accuracy achieved by the top-performing models indicates a practical utility for real-world ERM applications.

Keywords

Large Language Models, News-Stream Sequencing, Temporal Reasoning

1 INTRODUCTION

Within Enterprise Risk Management (ERM) practice, organizations monitor external developments also by analyzing streams of publicly available news. Each news article captures a momentary state of the political-economic environment, and by accurately structuring unordered information into a chronological narrative, organizations can better understand the evolution of events and the relationships that connect them. The reconstruction and forecasting of these event trajectories are important for identifying, assessing, and predicting emerging risks, especially within risk scenario analysis [10, 11]. The capability to build structured timelines from unstructured textual information is therefore of high relevance to ERM.

LLMs are increasingly utilized in ERM for their ability to process and analyze unstructured textual data, including news articles, to identify and assess risks [1, 2, 3, 4, 5]. In the financial sector, applications include extracting sentiment from news to gauge market perception or identify reputational risks [3, 6, 7, 8], and identifying specific risk factors or events discussed in news and corporate disclosures [2, 4, 5, 9]. Existing literature mainly demonstrates LLMs' utility in analyzing individual or aggregated news items for tasks such as sentiment analysis, risk factor identification, or event detection, but the capabilities of the models to recover the temporal order and causal links among a sequence of discrete news items that describe an unfolding narrative are less directly explored. This paper aims to address this gap by investigating LLM performance in temporal-causal reasoning within news streams, a crucial aspect for understanding the dynamics of unfolding risk narratives.

By investigating whether state-of-the-art commercial or open-source LLMs can reconstruct the chronological narrative of business-event chains from unordered news articles, this paper contributes to the field by: (a) systematically evaluating the performance of multiple LLMs on a challenging temporal-reasoning task; (b) analysing the efficacy of diverse prompting strategies — both unassisted and assisted — in improving model accuracy; (c) providing insights into model-and-task dynamics, revealing substantial performance disparities, task-specific difficulty patterns, and the outsized gains weaker models receive from contextual hints; and (d) demonstrating the practical readiness of these technologies for ERM deployment.

2 RESEARCH METHOD

Task Definition

To evaluate the capabilities of LLMs, three event chains were constructed, focusing on: (1) Trump's Tariffs and EU [“Task_1”], (2) Gold Prices [“Task_2”], and (3) the Ukraine-Russia War [“Task_3”]. These topics were selected due to their significant relevance to the business environment. For each topic, ten articles were manually selected from the online editions of two reputable sources of financial and business information, published between March 1st and May 2nd, 2025. For the purpose of LLM processing, the raw text from the selected articles was extracted. To prevent temporal bias, explicit date indicators—such as full dates—were removed, and no two

Information Society 2025, 6–10 October 2025, Ljubljana, Slovenia

http://doi.org/ 10.70314/is.2025.sikdd.4

articles shared the same publication date. Subsequently, the articles within each event chain were randomly shuffled, and this fixed random order was then applied to all models within the experiment.

The primary task for the selected LLMs was to reconstruct the chronological sequence of news articles within three distinct event chains. This task was evaluated across four experimental scenarios: (1) an unassisted scenario ["Assist_No"], and three assisted scenarios providing the (2) first ["Assist_First"], (3) last ["Assist_Last"], or (4) both first and last ["Assist_FirstLast"] articles in the sequence.

In the unassisted scenario, the LLMs were required to determine the correct chronological order of the articles without any external information regarding their placement. In the assisted scenarios, the models were provided with hints within the user prompt. Specifically, for the Assist_First and Assist_Last scenarios, the prompt identified the article occupying the initial or final position, respectively. In the Assist_FirstLast scenario, the LLMs were given the identifiers for the articles that correspond to the beginning and end of the chronological sequence.

The required output from the LLMs was a reconstructed timeline of the news articles. For each position in the timeline, the following information was mandated: (i) the article's identification number, (ii) the article's title, (iii) a brief justification for its placement relative to the preceding article, and (iv) a brief justification for its placement relative to the subsequent article. The models were required to provide a structured output in JSON format.

Prompt Engineering

Prompt engineering included manual drafting, testing on different models, and optimization both with LLMs (GPT o3 and Gemini 2.5 Pro) as well as manually, in several iterations. In the end, an effective user prompt was developed which worked reasonably well for all selected models. The main challenges with regard to the design of prompts were: (a) stimulating a systematic approach to causal reasoning, which was considered to be mainly important for the non-reasoning models; (b) ensuring the output consisted of exactly ten distinct articles, with no repetitions or omissions; (c) enforcing the required output JSON schema; and (d) providing concise reasoning for the positioning of the observed articles.

Within the user prompt, the models were explicitly instructed to use the following reasoning principles: (a) inferring sequences of events (how events described in different articles relate to each other over time), (b) causal reasoning (identifying cause-and-effect relationships between the content of different articles), (c) logical story progression (understanding how a narrative or situation typically develops or unfolds), (d) utilizing any implicit time references if available within the articles, and (e) using models' general knowledge about events. Prompts with clear instructions about the guidelines for the reasoning process worked better than prompts without such instructions, even with models with strong reasoning capabilities. System prompts were not utilized, as the one-shot user prompt contained all necessary instructions for the models. The full user prompt is available from the authors.

Selected LLMs and Experiment Execution

Twenty different models by eight different providers were selected for this research, based on their expected capabilities with regard to the tasks, and their availability. Overview of selected models is shown in Table 1.

Table 1: Selected LLMs

#	Model Provider: Model Name	Context Window (tokens)	Date Created
1	OpenAI: GPT-4.1	1.047k	14.04.2025
2	OpenAI: o3	200k	16.04.2025
3	OpenAI: o3-pro	200k	10.06.2025
4	OpenAI: gpt-oss-120b	131k	5.08.2025
5	OpenAI: GPT-5	400k	7.08.2025
6	Google: Gemini 2.5 Pro Preview	1.048k	7.05.2025
7	Google: Gemini 2.5 Flash Preview	1.048k	20.05.2025
8	xAI: Grok 3 Beta	131k	9.04.2025
9	xAI: Grok 4	256k	9.07.2025
10	Anthropic: Claude Sonnet 4	200k	22.05.2025
11	Anthropic: Claude Opus 4	200k	22.05.2025
12	Anthropic: Claude Opus 4.1	200k	5.08.2025
13	Meta: Llama 4 Maverick	1.048k	5.04.2025
14	Meta: Llama 4 Scout	1.048k	5.04.2025
15	Mistral AI: Mistral Medium 3	131k	7.05.2025
16	Mistral AI: Mistral Medium 3.1	262k	13.08.2025
17	Qwen: QwQ 32B	131k	5.03.2025
18	Qwen: Qwen 2.5 VL 32B Instruct	128k	24.03.2025
19	DeepSeek: DeepSeek V3	163k	24.03.2025
20	DeepSeek: R1	128k	28.05.2025

All models were accessed using the OpenRouter platform via the APIs. For models supporting this parameter, the temperature was set to 0.0 to ensure the most reliable and reproducible experimental results; otherwise, default parameters were used.

There were 12 experiments executed: 3 different event topic chains (tasks) in 4 experimental scenarios (prompts) each, by using all 20 LLMs as shown in Table 1, thus resulting in 240 results (outputs). Experiments were executed on June 1st, 2025 with the models available on that date, and on August 19th, 2025 with the newer models.

3 EVALUATION AND DISCUSSION

General Evaluation

In terms of the **output content**, all models demonstrated strong performance in response to a standardized user prompt, successfully producing the requested ordered lists of news articles with all accompanying metadata. From a logical standpoint, the outputs from all models were accurate, presenting ordered lists that included all required supplementary information. Substantial variations in output quality were observed across the different models. This variation was also influenced by the three distinct tasks, which seemed to be of substantially different difficulty, with the first task being the most straightforward and the last presenting the most significant challenge. As anticipated, the implementation of assisted prompting strategies consistently enhanced the accuracy of the outputs for all models across all evaluated tasks.

Regarding the **output formatting**, the majority of the models adhered to the specified JSON schema. Notable exceptions to

this were Claude models (models #10, #11 and #12), which occasionally deviated from the requested format by including a short introductory text. In these instances, the textual outputs were programmatically reformatted to conform to the required JSON structure. It is relevant to note that these three models are the only ones in the evaluation that do not natively support the Structured Output functionality, a factor that likely contributed to their formatting inconsistencies.

Performance Metric

To quantify the models' performance with the given tasks, a robust evaluation metric was required. For this purpose, **Kendall's rank correlation coefficient** ("Kendall's τ", "τ") was selected as the most appropriate measure. Kendall's τ is a non-parametric statistic that measures the ordinal association between two ranked lists. Its methodology is centered on comparing the concordance of all possible pairs of items within the sequences, yielding a score in the interval from -1 (perfect reversal) to +1 (perfect match). The focus on relative, pairwise ordering makes Kendall's τ exceptionally well-suited for a chronological sorting task, as the core challenge lies in correctly establishing which event occurred before another, which is precisely what the metric evaluates.

An alternative metric, the sum of absolute Manhattan distances, was also considered but ultimately deemed less suitable. Its primary drawback is its sensitivity to the magnitude of displacement, which can produce misleading evaluations by heavily penalizing single items that are wildly out of place, while potentially under-penalizing a sequence with numerous smaller, local errors that might represent a poorer overall sort.

Performance by Tasks and Scenarios

The performance of each model, quantified by the Kendall's τ, is detailed in Tables 2 and 3. Table 2 presents the coefficients organized by task (event chain), averaged across all experimental scenarios (prompts). Table 3, in turn, presents the coefficients organized by experimental scenario, averaged across all the tasks. The ranks in both tables were determined by averaging the performance rankings of all the models across individual tasks and scenarios. They largely correspond to the rankings based on average τ, but discrepancies may arise from variation in the scale and distribution of τ values across experiments.

To contextualize these performance metrics, their relationship to pairwise accuracy is critical: within a 10-item sequence, a Kendall's τ of 0.90, 0.80 or 0.50 indicates that approximately 95%, 90% or 75% of the 45 possible pairs are concordantly ordered, respectively.

The aggregated results in Table 2 underscore two principal findings. First, a significant and systematic variation in task difficulty was evident, with Task_1 representing the simplest case and Task_3 the most demanding. This pattern held true for practically all the evaluated models and experimental scenarios. The performance differences indicating different task difficulty were substantial. For Task_1 and the unassisted scenario, the Kendall's τ values for the average, best model, and worst model performance were 0.78, 0.91 and 0.02, respectively. For Task_2, the values were 0.63, 1.00 and 0.16, and for Task_3, they were 0.02, 0.38 and 0.33. These findings clearly establish Task_3 as the most difficult of the three tasks evaluated. Note that a negative Kendall's τ value indicates an inverse correlation between the predicted and true rankings, and a value around zero represents a random ordering. Second, the results show that the more recent versions and models with strong reasoning capabilities (models Grok 4, GPT-5, o3 and o3-pro, and Gemini 2.5 Pro) consistently outperform other models practically across all tasks.

Table 2: Average Performance by Tasks (Kendall's τ)

Rank	Model #	Task_1	Task_2	Task_3	Avg. τ
1	9	0.96	0.98	0.70	0.88
2	2	0.94	0.94	0.56	0.81
3	5	0.96	0.99	0.49	0.81
4	3	0.94	0.93	0.52	0.80
5	6	0.94	0.96	0.52	0.81
6	8	0.93	0.79	0.43	0.72
7	12	0.94	0.70	0.41	0.69
8	20	0.83	0.82	0.50	0.72
9	7	0.84	0.89	0.48	0.74
10	11	0.93	0.67	0.36	0.65
Avg. top 5:		0.95	0.96	0.56	0.82
Avg. all 20:		0.85	0.71	0.36	0.64

The aggregated results in Table 3 underscore three principal findings. First, assisted prompting systematically improved the performance across all models and tasks, which is logical and expected since additional relevant information is provided to the models. Anchoring with known positions in the majority of cases helped the models to better position the remaining articles as well.

Table 3: Average Performance by Scenarios (Kendall's τ)

Rank	Model #	Assist_ No	Assist_ First	Assist_ Last	Assist_ FirstLast	Avg. τ
1	9	0.75	0.88	0.90	0.99	0.88
2	2	0.69	0.88	0.76	0.93	0.81
3	5	0.73	0.84	0.81	0.87	0.81
4	3	0.72	0.87	0.76	0.85	0.80
5	6	0.57	0.93	0.84	0.90	0.81
6	8	0.48	0.81	0.78	0.81	0.72
7	12	0.48	0.66	0.73	0.87	0.69
8	20	0.66	0.75	0.64	0.82	0.72
9	7	0.54	0.73	0.81	0.87	0.74
10	11	0.48	0.64	0.66	0.82	0.65
Avg. top 5:		0.69	0.88	0.81	0.91	0.82
Avg. all 20:		0.47	0.67	0.63	0.79	0.64

Second, the provision of additional information proved more beneficial for the most demanding task (Task_3) than for the less demanding tasks (Task_1 and Task_2). For example, in the Assist_FirstLast scenario, the increase in average τ relative to the unassisted scenario was 0.13 for Task_1, 0.17 for Task_2, and 0.65 for Task_3. This finding follows logically from the models' greater ability to identify the first and/or last article in simpler tasks by themselves: in Task_1, 15 of 20 models correctly identified the first position, while none identified the last position, in Task_2 9 models identified the first position and 4 identified the last position, and in Task_3 no model identified either position correctly.

Third, the provision of additional information disproportionately benefited models that performed poorly in the unassisted scenario. For instance, on Task_3 — the most difficult task with

an average Kendall's τ of only 0.02 in the unassisted scenario — the Assist_First scenario yielded average and maximum performance improvements of 0.46 and 1.07, respectively. For the Assist_Last scenario, the corresponding improvements were 0.27 and 0.80, while for the Assist_FirstLast scenario they were 0.65 and 1.02. The results demonstrate that supplementing less capable models with limited key information can yield significant performance gains at these tasks.

A qualitative examination of the models' reasoning justifications failed to yield systematic insights into their capacity to reconstruct accurate chronological sequences of articles. Although the generated rationales were generally logical and relevant, they frequently omitted crucial contextual information essential for correct chronological reasoning. This observation underscores the challenge that certain timelines may not be uniquely re-constructible due to insufficient contextual information. Furthermore, in some instances, the provided justification could plausibly support an alternative, yet equally valid, timeline. Moreover, this is compounded by the inherent challenge of discerning whether the provided reasoning justifications represent the model's actual inferential process or are merely a result of the post-hoc rationalization.

4 CONCLUSIONS AND FURTHER RESEARCH IDEAS

This research provides insight into the practical application and inherent challenges of utilizing LLMs to sequence news streams in the context of ERM. The selected use cases are based on real-world, business-relevant event chains.

A comparative analysis reveals significant performance disparities among the evaluated models across all tasks and experimental scenarios. Models with superior reasoning capabilities surpassed those with less developed abilities. The varying complexity of the presented tasks further accentuated these performance differences. Also, providing additional anchoring information disproportionately benefited models that performed poorly in the unassisted scenario. Five models, **Grok 4** (xAI), **GPT-5**, **o3** and **o3-pro** (all three OpenAI), and **Gemini 2.5 Pro** (Google), consistently outperformed all other models in practically every task and experiment scenario. The performance level achieved by these models demonstrates their practical utility for real-world ERM applications.

This research has opened several promising areas for **further research**:

(1) Benchmarking LLMs against human experts: A rigorous comparative study should be undertaken in which large LLMs and domain specialists (human experts) perform identical tasks under strictly matched contextual conditions.

(2) Systematically varying model settings to probe "creativity" and reliability: Experiments that modulate the temperature and other model settings can clarify how stochasticity affects task performance and reliability.

(3) Enabling models to request task-critical information: Instead of supplying predefined contextual information—such as the first and/or last article in a sequence—future studies might allow the model to query for the minimal supplementary data it deems most informative. This strategy would approximate an active-learning workflow and might even illuminate new modes for human-LLM collaboration.

(4) Diagnosing mis-ordering errors through reasoning audits: To understand why models fail to reconstruct the correct temporal ordering of news articles, one could extract each model's stated reasoning features for every placement decision, then have human experts or adjudicating LLMs rate their accuracy and relevance. Such audits would expose specific deficits in reasoning and could even inform targeted retraining regimes.

(5) Experimenting with extended or interleaved event chains: Evaluating models on substantially longer sequences—or on mixtures of events drawn from multiple chains—would markedly raise task complexity and furnish a stringent benchmark of temporal-reasoning competence for business use cases.

ACKNOWLEDGMENTS

The authors acknowledge the use of LLMs during various stages of this research. These models provided support in tasks such as idea generation, text processing, prompt engineering, methodological exploration, and language optimization. While the LLMs contributed to enhancing efficiency and refining the presentation of this work, all conceptual frameworks, analyses, and interpretations remain the sole responsibility of the authors.

REFERENCES

[1] Y. Cao et al., 'RiskLabs: Predicting Financial Risk Using Large Language Model Based on Multi-Sources Data', Apr. 11, 2024, arXiv: arXiv:2404.07452. doi: 10.48550/arXiv.2404.07452.

[2] A. Kim, M. Muhn, and V. V. Nikolaev, 'From Transcripts to Insights: Uncovering Corporate Risks Using Generative AI', Jul. 11, 2024, Rochester, NY: 4593660. doi: 10.2139/ssrn.4593660.

[3] T. Li and X. Dai, 'Financial Risk Prediction and Management using Machine Learning and Natural Language Processing', ijacsa, vol. 15, no. 6, 2024, doi: 10.14569/IJACSA.2024.0150623.

[4] Y. Wang, 'Generative AI in Operational Risk Management: Harnessing the Future of Finance', May 17, 2023, Rochester, NY: 4452504. doi: 10.2139/ssrn.4452504.

[5] X. Zhu, H. Jin, J. Li, and Y. Wang, 'Topic-Gpt: A Novel Risk Identification Method Based on Large Language Model', Jul. 04, 2024, Social Science Research Network, Rochester, NY: 4885365. doi: 10.2139/ssrn.4885365.

[6] M. Katamaneni, P. Agrawal, S. Veera, A. K. Sahoo, K. Singh Sidhu, and M. F. Hasan, 'AI-Based Risk Management in Financial Services', in 2024 Second International Conference Computational and Characterization Techniques in Engineering & Sciences (IC3TES), Nov. 2024, pp. 1–5. doi: 10.1109/IC3TES62412.2024.10877497.

[7] X. V. Li and F. S. Passino, 'FinDKG: Dynamic Knowledge Graphs with Large Language Models for Detecting Global Trends in Financial Markets', in Proceedings of the 5th ACM International Conference on AI in Finance, Nov. 2024, pp. 573–581. doi: 10.1145/3677052.3698603.

[8] A. Nygaard et al., 'News Risk Alerting System (NRAS): A Data-Driven LLM Approach to Proactive Credit Risk Monitoring', in Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track, F. Dernoncourt, D. Preoțiuc-Pietro, and A. Shimorina, Eds., Miami, Florida, US: Association for Computational Linguistics, Nov. 2024, pp. 429–439. doi: 10.18653/v1/2024.emnlp-industry.32.

[9] Z. Xiao, Z. Mai, Z. Xu, Y. Cui, and J. Li, 'Corporate Event Predictions Using Large Language Models', in 2023 10th International Conference on Soft Computing & Machine Intelligence (ISCMI), Nov. 2023, pp. 193–197. doi: 10.1109/ISCMI59957.2023.10458651.

[10] Committee of Sponsoring Organizations of the Treadway Commission (COSO), Enterprise Risk Management—Integrating with Strategy and Performance. Durham, NC: COSO, 2017.

[11] International Organization for Standardization, ISO 31000:2018 – Risk management — Guidelines. Geneva, Switzerland: ISO, 2018.

Graph-Based Feature Engineering for DeFi Security Incident Severity Prediction

Daria Pavlova*
daria.pavlova@mps.si
Jožef Stefan International Postgraduate School
Ljubljana, Slovenia

Inna Novalija
inna.koval@ijs.si
Jožef Stefan Institute
Ljubljana, Slovenia

Dunja Mladenić
dunja.mladenic@ijs.si
Jožef Stefan Institute
Ljubljana, Slovenia

ABSTRACT

Decentralized Finance (DeFi) has emerged as a rapidly growing sector, but it has been plagued by numerous security incidents resulting in billions of USD in losses. An important challenge is predicting which security incidents will lead to *severe* financial losses, as this can inform risk management and mitigation strategies. In this paper, we present a novel approach that integrates a semantic knowledge graph of the DeFi ecosystem into the machine learning pipeline for incident severity prediction. We construct a knowledge graph capturing rich relationships between DeFi protocols (including protocol fork lineage, multi-chain deployments, and historical incidents), and we engineer graph-based features from this graph to augment traditional incident features. Using these features in a gradient boosting trees classifier, we predict whether an incident will cause above-threshold (severe) losses. Our results show that incorporating graph-based features yields a substantial improvement in predictive performance: the model with semantic graph features achieves an Area Under ROC Curve (AUC) of 0.787, a 31.6% relative increase over the baseline model using only non-graph features. We observe particularly large gains in precision (from 0.341 to 0.490), indicating a significantly reduced false alarm rate. While these absolute performance values remain moderate, they represent substantial improvements for this challenging prediction task. The findings demonstrate the practical value of graph-enriched feature engineering for security analytics in DeFi. This work provides new insights into how protocol interconnections and characteristics contribute to incident severity, opening avenues for more robust DeFi risk assessment tools.

KEYWORDS

Decentralized Finance, DeFi, Security, Knowledge Graph, Feature Engineering, Incident Severity Prediction

1 INTRODUCTION

Decentralized Finance (DeFi) platforms have experienced rapid growth, alongside a surge in security breaches such as hacks and exploits. In 2022 alone, crypto attacks led to over $3.8 billion in stolen assets, with the majority coming from DeFi protocol exploits [1]. These incidents vary widely in impact: while many attacks result in limited losses, a significant fraction escalate into catastrophic failures causing losses in the tens or hundreds of millions of dollars. Predicting which security incidents will become *severe* (high-loss) events is crucial for proactive risk management, insurance underwriting, and developing early warning systems for the DeFi ecosystem.

Prior research has analyzed DeFi vulnerabilities and attack taxonomy [6], and industry reports highlight the growing scale of DeFi hacks. However, there is a gap in predictive approaches: existing studies focus on identifying vulnerabilities or classifying attack types, rather than forecasting the *severity level* of an incident before it fully unfolds. To our knowledge, this is the first work to apply semantic knowledge graph features specifically for DeFi incident severity prediction, establishing a new baseline for this important problem.

In traditional cybersecurity contexts, incorporating relational context via knowledge graphs and network models has been shown to improve threat detection [3]. For example, graph-based severity triage using attack graphs has been studied in traditional cybersecurity [5].

In this work, we propose a novel graph-based feature engineering approach to address this challenge. We construct a semantic **knowledge graph** of the DeFi ecosystem that encodes domain knowledge: nodes represent entities such as protocols and incidents, and edges capture relationships like *"forked-from"* (denoting protocol lineage) and *"deployed-on"* (connecting protocols to blockchain platforms), among others. From this knowledge graph, we derive a set of **graph-based features** for each security incident. These features quantify properties such as a protocol's structural position in the ecosystem (e.g., number of fork "children," cross-chain deployments, past incident count), which we posit are predictive of how severe an incident could be.

We integrate these semantic graph features with conventional features (e.g., time of incident, incident type categories) in a machine learning classifier to predict whether an incident's loss will exceed a severity threshold. The contributions of our work are as follows:

- We introduce a methodology to incorporate a DeFi-specific knowledge graph into security incident severity prediction.
- We demonstrate significant performance gains over a baseline model lacking graph features (improving AUC by 31.6% and F1-score by 25%).
- We provide a comprehensive analysis including case studies, illustrating how related protocol dependencies can influence risk.
- We discuss practical implications of our findings for improving DeFi risk assessment.

All code and the publicly available dataset for this work are available in an open-source repository [4].

*First author and presenter.

Information Society 2025, 6–10 October 2025, Ljubljana, Slovenia

https://doi.org/10.70314/is.2025.sikdd.6

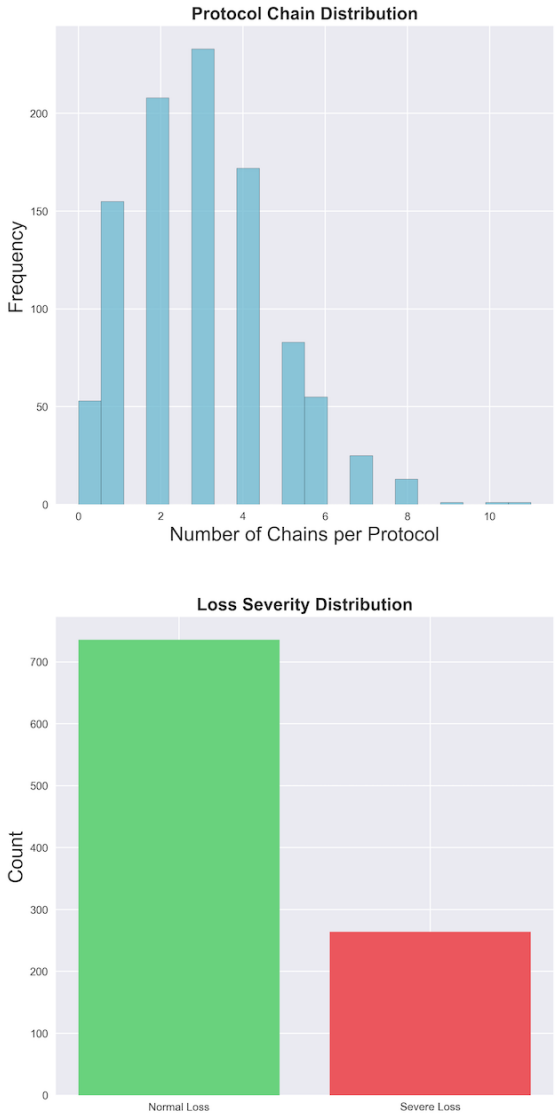


Figure 1: DeFi knowledge graph overview: protocols, blockchains, and incidents with relations (forked-from, deployed-on, involves).

Figure 2: Convex-centric subgraph. Dependency on Curve highlights potential severity propagation via upstream vulnerabilities.

2 METHODOLOGY

2.1 Knowledge Graph Construction

We built a knowledge graph representing the DeFi ecosystem to serve as a basis for feature engineering. The construction process was semi-automated, combining API data extraction with manual curation to ensure semantic consistency.

Data Sources: We integrated data from three primary sources: (1) the Rekt database (https://rekt.news) containing detailed DeFi security incident reports, (2) DeFiLlama's API providing protocol metadata including deployment chains and fork relationships, and (3) SlowMist Hacked for additional incident verification. All data sources are publicly available.

Semi-Automated Process: Protocol and incident data were automatically extracted using APIs and web scraping. Fork relationships were identified through a combination of automated code similarity analysis (for protocols with public repositories) and manual verification based on project documentation. The resulting knowledge graph contains 892 protocol nodes, 1,608 incident nodes, and 42 blockchain nodes, connected by over 3,500 edges representing various relationships. We use Neo4j to store and query this graph efficiently through asynchronous operations.

The graph's schema defines several entity types and relations relevant to DeFi security:

- **Protocol nodes:** Each DeFi protocol (e.g., lending platform, DEX, yield aggregator) is a node. Attributes include protocol name and launch date.
- **Incident nodes:** Major recorded security incidents (hacks, exploits) are represented as nodes with attributes such as date, loss amount, and qualitative classification (e.g., flash loan, smart contract bug).
- **Blockchain nodes:** Blockchain platforms (Ethereum, Binance Smart Chain, etc.) are included to capture deployment contexts.

Key relationships are encoded as directed edges:

- *Fork-of:* Connects a protocol to the protocol it was forked from (if applicable), capturing lineage (e.g., SushiSwap → Uniswap).
- *Deployed-on:* Links a protocol to a blockchain platform on which it is deployed.
- *Incident-involves:* Links an incident node to the protocol(s) affected by that incident.

The resulting graph captures a rich hierarchical structure of protocol relationships (including parent–child fork trees and cross-chain deployment links), as well as the association of past incidents with protocols.

An overview of the graph structure is shown in Figure 1, and an illustrative Convex-centric subgraph is given in Figure 2.

2.2 Feature Engineering with Graph-Based Features

From the knowledge graph, we derived several quantitative features that characterize the structural and historical context of the protocol involved in a given incident:

- **Protocol multi-chain count**: the number of distinct blockchains on which the protocol is deployed (degree of *deployed-on* edges). A higher count indicates a widely deployed protocol, potentially implying larger user bases or attack surfaces.
- **Fork lineage indicators**: whether the protocol is a fork of another (*has parent*) and the number of forks derived from it. These capture if a protocol inherits code (and possibly vulnerabilities) from a parent and how prevalent its code is in offspring projects.
- **Past incident count**: the total number of past security incidents involving the protocol (count of *incident-involves* edges to prior incidents). A history of frequent past incidents might signal underlying security weaknesses or attractive target value.

In addition to these graph-derived features, we include conventional features for each incident:

- **Temporal features**: the year and month of the incident, and day-of-week if relevant, to capture any time-related patterns or trends in attack occurrence.
- **Categorical features**: the general type of attack or vulnerability exploited (e.g., reentrancy, price oracle manipulation), and the asset or protocol category targeted, which provide contextual information on the incident.

All features are computed or retrieved at the time just before the incident (to avoid using any post-incident information). The combination of graph-based features with traditional features forms the feature vector used for prediction.

The end-to-end feature extraction and modeling pipeline is summarized in Figure 3.

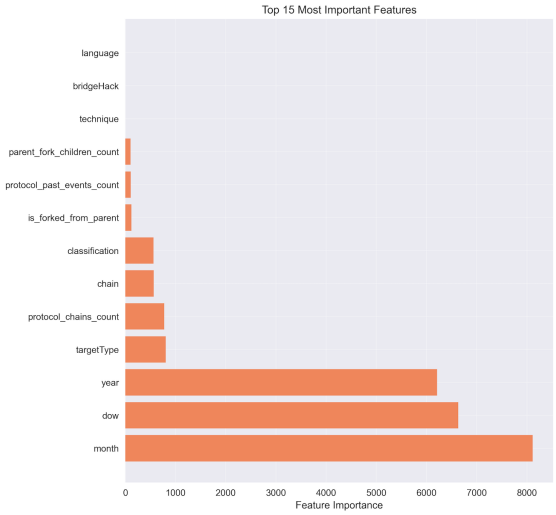


Figure 3: Workflow: derive graph-based features from the DeFi knowledge graph and combine with conventional incident features for classification.

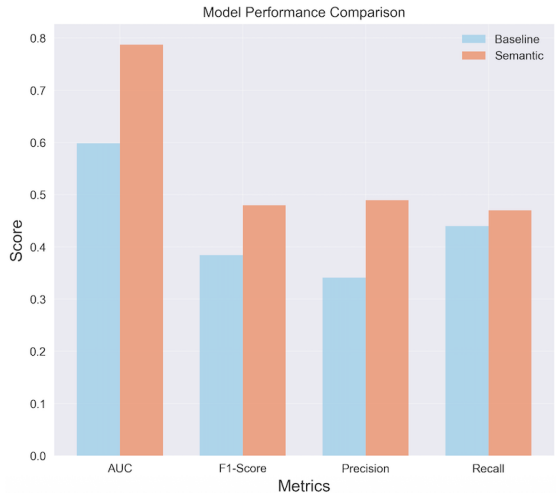


Figure 4: Performance comparison. Bar chart for AUC, F1, precision, recall.

2.3 Classification Model and Training

We frame incident severity prediction as a binary classification task: *severe* vs. *non-severe* loss outcome. Following prior work in financial risk modeling, we define a severe incident as one with loss exceeding a high quantile threshold of the loss distribution. In our dataset, we tested multiple thresholds (70th, 75th, and 80th percentiles), with the 75th percentile ($2.21 million) serving as the primary cutoff, yielding 402 severe incidents out of 1,608. The model showed consistent improvements across all thresholds, confirming the robustness of our approach.

Our primary model is a gradient boosting decision trees ensemble (LightGBM [2]), selected for its efficiency, ability to handle heterogeneous feature types, and proven performance in tabular financial risk modeling. We enabled LightGBM's built-in class imbalance option (`is_unbalance=True`), as severe cases represent 25% of the data.

Train/Test Split: Data were split chronologically into 75% training and 25% testing. Early stopping was not applied due to dataset size; hyperparameters were fixed after preliminary tuning.

We compare two feature sets: a **Baseline** model using only non-graph features (temporal and categorical), and a **Semantic Graph** model combining these with graph-based features. Performance is evaluated with Area Under the ROC Curve (AUC) and supported by Precision, Recall, and F1-score.

3 EXPERIMENTS AND RESULTS

3.1 Dataset and Experimental Setup

We compiled a publicly available dataset of 1,608 DeFi security incidents that occurred between 2020 and 2025. The dataset was constructed by combining data from: (1) Rekt database providing comprehensive incident reports with loss amounts and attack descriptions, (2) DeFiLlama API for protocol metadata including TVL and deployment information, and (3) SlowMist Hacked for additional incident verification and technical details. Each incident record includes the loss amount (in USD) and details such as date and attack type. Incidents with losses above $2.21 million were labeled as *severe*, which yields a severe class prevalence of roughly 25% (402 severe vs. 1,206 non-severe cases).

For training and evaluation, we use a chronological split with 75% for training and 25% for testing; early stopping was not applied.

Table 1: Performance comparison between the baseline model (numeric/categorical features only) and the semantic graph model (with knowledge graph features).

Metric	Baseline	Semantic Graph	Improvement
AUC	0.598	0.787	+31.6%
F1-score	0.384	0.480	+25.0%
Precision	0.341	0.490	+43.7%
Recall	0.440	0.470	+6.8%

3.2 Performance Comparison

A visual comparison of model performance is shown in Figure 4.

Our results confirm that incorporating graph-based features markedly improves prediction performance. Table 1 summarizes the evaluation metrics for the baseline and semantic graph-enhanced models on the test set. The Semantic Graph model achieves an AUC of 0.787, substantially higher than the baseline's 0.598 (a relative improvement of 31.6%). This indicates that the model with graph features is much better at ranking incidents by risk. The F1-score also improves from 0.384 to 0.480, reflecting better overall classification accuracy.

Notably, the Precision (positive predictive value) rises from 0.341 to 0.490—a 43.7% increase—while Recall increases slightly from 0.440 to 0.470. This suggests that the graph-enriched model is significantly more effective at identifying truly severe incidents (fewer false positives) without sacrificing the ability to catch most severe cases. While the absolute values of these metrics might appear moderate, it is important to note that they represent substantial improvements over the baseline and are competitive for this specific and challenging prediction task where many external factors influence incident severity.

In addition to the hold-out test, we evaluated stability via cross-validation. The baseline model's mean AUC across 5 folds was 0.629 (std 0.036), whereas the semantic model averaged 0.809 (std 0.027). This not only reaffirms the performance boost but also indicates that the graph-augmented model is more consistent across different data subsets (lower variance), likely because the graph features provide a more robust signal that generalizes.

3.3 Feature Importance Analysis

To better understand the relationships between graph-based features, we analyzed their pairwise correlations (Figure 5). The correlation matrix shows that most features are only weakly related, which indicates that they capture complementary aspects of protocol structure and history. The strongest dependency is observed between *is_forked_from_parent* and *parent_fork_children_count* (correlation 0.64), reflecting the natural link between fork origin and the number of derived protocols. Other features, such as *protocol chains count* and *protocol past events count*, exhibit low correlation values (<0.2), suggesting they provide distinct signals. This relative independence confirms that graph-derived features enrich the predictive model with diverse information rather than duplicating each other.

4 DISCUSSION

Our results highlight the value of relational context for DeFi security analysis: knowledge graph features capture ecosystem-level dependencies not visible from incident-centric data. Incidents affecting widely forked or multi-chain protocols are more likely to cause severe losses, reflecting practical amplification effects.

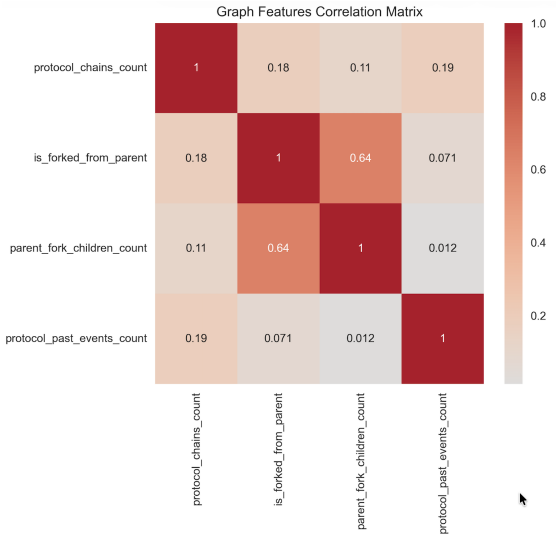


Figure 5: Top 15 most important features ranked by LightGBM gain. Values on the x-axis represent LightGBM's internal feature importance scores (dimensionless, aggregated across all trees in the ensemble). Both temporal features (year, month, day-of-week) and graph-based features (e.g., *protocol_chains_count*, *is_forked_from_parent*) appear among the strongest predictors.

Applications: Graph-based risk factors can support auditors and insurers in identifying critical "hot spots" and pricing coverage more accurately than historical losses alone.

Limitations: The dataset covers only publicly reported incidents, which may bias toward large-scale events. Features are manually engineered and static; future work should explore dynamic graphs, Graph Neural Networks, and richer incident coverage. Absolute performance (AUC 0.787) remains moderate, leaving room for improvement before real-world deployment.

5 CONCLUSION

We introduced a graph-enriched framework for predicting severity of DeFi security incidents. By combining semantic knowledge graph features with conventional incident data, our model achieved substantial gains over a feature-only baseline. The findings emphasize that *where* an incident occurs in the ecosystem is as important as *what* it is. This approach offers immediate utility for risk assessment and motivates further research into dynamic, end-to-end graph-based models for DeFi security.

REFERENCES

[1] Chainalysis Team. 2023. 2022 Biggest Year Ever For Crypto Hacking with $3.8 Billion Stolen, Primarily from DeFi Protocols and by North Korea-linked Attackers. *Chainalysis Blog* (1 February 2023). https://www.chainalysis.com/blog/2022-biggest-year-ever-for-crypto-hacking/

[2] G. Ke et al. 2017. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In *Advances in Neural Information Processing Systems 30*. 3146–3154.

[3] J. Michel and P. Parrend. 2023. Graph-Based Intelligent Cyber Threat Detection System. In *Cybersecurity in Intelligent Networking Systems*. CRC Press.

[4] D. Pavlova. 2025. DeFi Security Trends: Semantic Knowledge Graph Analysis (Code & Dataset). GitHub Repository. https://github.com/dariapavlova02/defi_trends_semantic

[5] L. Sadlek et al. 2025. Severity-Based Triage of Cybersecurity Incidents Using Kill Chain Attack Graphs. *Journal of Information Security and Applications* 89 (2025).

[6] S. Werner, D. Perez, L. Gudgeon, A. Klages-Mundt, D. Harz, and W. Knottenbelt. 2021. SoK: Decentralized Finance (DeFi). *arXiv preprint* arXiv:2101.08778 (2021).

Evolving Neural Agents in Simulated Ecosystems

Marija Ćetković
UP FAMNIT
Koper, Slovenia
marijacetkovic03@gmail.com

Aleksandar Tošić
UP FAMNIT
Koper, Slovenia
aleksandar.tosic@upr.si

Domen Vake
UP FAMNIT
Koper, Slovenia
domen.vake@famnit.upr.si

Abstract

This paper explores how adaptive behaviors can emerge in artificial agents through neuroevolution in a dynamic 2D ecosystem. Using the NeuroEvolution of Augmenting Topologies (NEAT) algorithm both the neural network structure and weights evolved over time without predefined architectures or behaviors. The system models two agent types: herbivores and carnivores that compete for limited food resources in a simulated environment. From the beginning, it was evident that environment design, input encoding, and reward shaping had a major impact on agent behavior. Poorly tuned conditions led to exploitation, overfitting, or meaningless patterns. But when the system was carefully balanced, the agents began developing survival strategies such as movement efficiency, food seeking, and attacking. Herbivores evolved plant consumption behaviors, while carnivores built on this base to prioritize attacks and meat consumption. Some behaviors generalized well to larger environments, showing that agents were not just memorizing patterns. We observed how NEAT's speciation and innovation mechanisms were crucial for maintaining diversity and avoiding premature convergence. At the same time, challenges like catastrophic forgetting revealed the limitations of neural networks in long-term skill retention. Ultimately, this work demonstrates how intelligent, adaptive behavior can emerge from simple evolutionary principles and offers a foundation for future research into co-evolution, agent roles, and artificial life.

Keywords

neuroevolution, NEAT, evolutionary algorithms, artificial life, simulated ecosystems, co-evolution, neural networks

1 Introduction

This research explores neuroevolution for adaptive agent behaviors in a dynamic ecosystem. Agents are controlled by feedforward neural networks that map sensory inputs to actions [4], and their structure and weights evolve incrementally using the NEAT algorithm [5]. Unlike static or predefined tasks, this simulation presents agents with a changing environment where no explicit 'correct' behavior exists.

Dynamic environments without fixed objectives require exploratory and adaptable approaches. Gradient-based optimization relies on differentiable fitness functions and fixed topologies, while reinforcement learning can struggle under sparse rewards. Evolutionary algorithms, by evaluating populations of agents directly on survival and performance, provide a natural solution for such open-ended scenarios [1]. Neural networks allow agents to flexibly map sensory input to actions, and NEAT enables both the network topology and weights to evolve over time in comparison to fixed-topology weight-evolving evolutionary methods. We implemented NEAT from scratch to have full control over mutation, crossover, and fitness evaluation, ensuring that the system could support our experimental goals and to potentially build a controllable and extensible evolutionary framework.

While NEAT has been previously applied to multi-agent systems, many studies focus on performance in a specific task. This paper addresses whether an agent-based NEAT framework can produce ecological equilibrium without an explicit objective. Our primary contribution is the demonstration and analysis of stable, co-adaptive predator-prey dynamics, showing how specific evolved behaviors arise from the underlying neural network topologies of the agents.

2 Methods

2.1 Environment Model

The ecosystem is a discrete 2D grid populated with food resources and agents. Herbivores consume plants, carnivores consume meat, and all agents perceive their surroundings through a limited sensory range.

2.2 Evolutionary Framework

Agents (creatures) interact with the world and are controlled by neural networks (genomes) evolved using NEAT. Initial populations start with minimal structures (fully connected input/output layers), and complexity increases through structural mutations. Genomes consist of genes, which are lists of nodes (with ID and type: input, hidden, output) and connections (with ID, nodes they connect, weight and enabled flag). Each tick, each agent receives a snapshot of the world state as input, to ensure stable input for everyone. Inputs include diet type, hunger level, local 3x3 neighborhood scan for food, neighbors (type and health level), and direction toward the nearest food source. Based on that agents choose actions as softmax output of their neural networks. The outputs correspond to discrete actions: move (up, down, left, right), eat, attack, or stay. The actions become events that are handled in a deferred manner. First, the invalid actions are filtered out, then the EventManager processes all queued events at once sequentially: applying changes in the world, updating fitness, and health of agents, which can be seen in the Algorithm 1.

The fitness function evolved through experimentation. Early versions rewarded survival, but later iterations combined survival time, food consumption, and for carnivores, attack behavior.

2.3 NEAT Mechanisms

Innovation Tracking is the process of tracking structural mutations globally to keep genomes aligned during crossover. A singleton class assigns a unique ID to each new connection or node. If a structural change already exists, it reuses the same ID; if not, it creates a new one. This ensures a consistent identification of identical innovations in all genomes [5].

Information Society 2025, Ljubljana, Slovenia

https://doi.org/10.70314/is.2025.sikdd.10

```
while generation limit not reached do
    // Simulation Phase
    while creatures alive do
        foreach creature c in population do
            c.observe(world) ;            // perception
            c.chooseAction() ;             // NN policy
            c.queueAction() ;        // check validity,
             enqueue
        end
        eventManager.process() ;       // apply action
         effects
        foreach creature c in population do
            c.updateHealth() ;   // starvation, death
            c.evaluateFitness() ;     // assign fitness
        end
    end
    // Evolution Phase
    assignGenomesToSpecies() ;          // speciation
    createOffspring() ;   // apply GA within species
    resetWorld() ; // spawn new creatures and food
end
```

Algorithm 1: High-Level Evolutionary Simulation Loop

NEAT preserves evolutionary innovation by speciation (niching) [5]. Each generation, evaluated genomes are reassigned to species based on structural compatibility distance, which is calculated as a weighted sum of the number of disjoint and excess connections (present in one parent, within and beyond other's genome region respectively), and weight difference averages between the matching (present in both parents) ones, given by: $\delta = \frac{c_1 E}{N} + \frac{c_2 D}{N} + c_3 \cdot W$. Existing species are cleared and each genome is compared to species representatives; if no match is found, a new species is created. Representatives are updated every generation to maintain diversity. Fitness is shared within species (adjusted by species size) to balance selection pressure. The compatibility threshold strongly affects stability: low thresholds create many narrow species, high thresholds create broader but less distinct species, requiring careful tuning.

To prevent the population from maintaining one dominant species and limiting the exploration of the algorithm, NEAT uses adjusted fitness [5]. Instead of assigning raw fitness scores, the individual's fitness is adjusted by the number of individuals who are within its distance delta, given by: $f_i' = \frac{f_i}{\sum_{j=1}^{n} sh(\delta(i,j))}$.

Evolution is achieved through genetic operators within species:

Mutation: Weight changes (random reset 5–10% or small Gaussian perturbation) and structural changes (adding nodes or edges, toggling connections). Resulting genome is checked for acyclicity.

Crossover: Offspring inherit connections aligned by innovation number; matching ones are inherited from the first parent, while disjoint and excess come from the parent with greater fitness score (or random if equal). Invalid (cyclic) offspring are replaced by mutated fitter parent.

Selection: Parent selection uses tournament selection: a subset of individuals (size 5) is sampled and the fittest is chosen. With 3% probability, a random individual is selected to maintain diversity. This setup provides moderate selection pressure - avoiding premature convergence while keeping implementation simple, efficient, and robust across different fitness functions.

2.4 Graphical User Interface

Figure 1 shows an example of the GUI which serves to visually track the simulation in real time, making the evolutionary process observable and interpretable, as analyzing logs alone could be misleading. It allowed following the population changes over generations, spotting emerging behaviors such as movement patterns or interactions, and understand whether agents are actually evolving. It helps detect issues such as creatures moving in the same direction or wandering aimlessly.

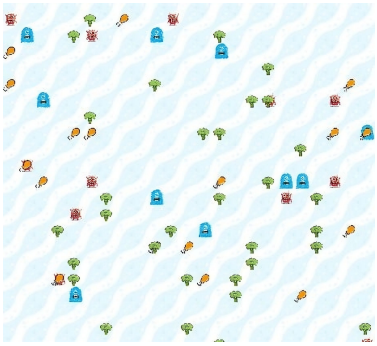


Figure 1: GUI close-up

2.5 Implementation Notes

The simulation was implemented in Java with LibGDX [2] for visualization. NEAT logic included custom classes for genomes, species management, and innovation tracking. The evolutionary loop evaluated agents in the world, assigned fitness, reproduced genomes, and reset the environment for subsequent generations.

2.6 Setup

After every run around 10 percent of the population is saved and loaded for the next run, with that part of population unchanged and the rest filled with mutations of it. This is done to speed up the evolution process. In early runs, we disabled the perception of other agents to prevent confusion and help them learn basic eating behavior. Once they consistently moved and consumed food, perception was turned on to allow them to adapt to a more complex environment. We also tested this logic on other inputs such as the food direction vector left agents essentially 'blind' to non-local food. So, during early iterations, we spawned food in random concentrated areas rather than spreading it widely, to help them learn to use this vector.

3 Results

3.1 Herbivore Evolution

Herbivores initially explored aimlessly but gradually developed stable food-seeking strategies. Over 800 generations, their action distribution stabilized, with movement actions dominating and eating consistently rewarded. In larger environments, agents prioritized exploration to reach scarcer resources, showing emergent adaptation beyond memorized patterns.

We can see from Fig.2 that the initial fitness highly oscillates, with great difference in average and maximal fitness, as well as some outliers with high fitness who end up consuming a lot of food. This is expected to some extent, as when one creature consumes food, it reduces the available resources for others in the population.

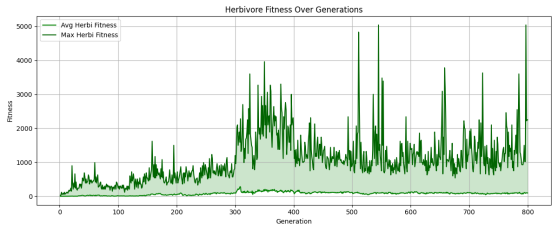


Figure 2: Herbivore fitness

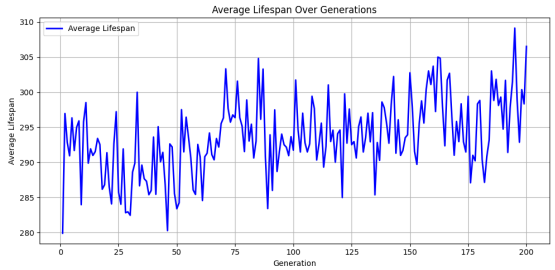


Figure 3: Average creature lifespan

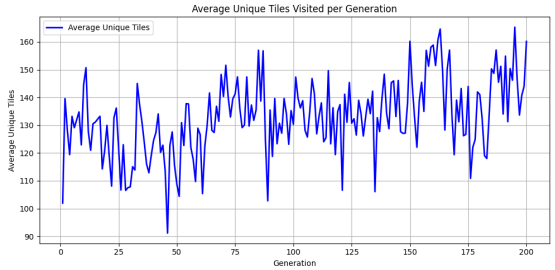


Figure 4: Average number of unique tiles visited

Figures 3 and 4 show agents that survive longer naturally explore more of the environment. The observed correlation between emerges from adaptive behavior under environmental constraints, rather than from any explicit exploration rewarding.

3.2 Carnivore Adaptation and Catastrophic Forgetting

Carnivores were evolved by reusing herbivore topologies and adjusting weights, transferring eating behavior to meat sources. This transfer was successful in just 200 generations, but the agents showed catastrophic forgetting when switched back to herbivore roles, losing previous behaviors [3]. This showed us that we needed more general pretraining to make sure that agents were using their role, food and food vector inputs, and not overfitting to the food type.

3.3 Co-Evolution Dynamics

To try to avoid the problem of forgetting mentioned, we saved agents of both types that evolved their basic skills independently. When carnivores were alone we gave them no motivation to use the attack action, to wire the logic later to herbivores. The attacking behavior was rewarded only for carnivores, but as shown below, some role interference was inevitable.

In smaller worlds, herbivores focused on eating, carnivores split between eating and attacking; in larger worlds, carnivores prioritized attacking, herbivores balanced movement and eating.

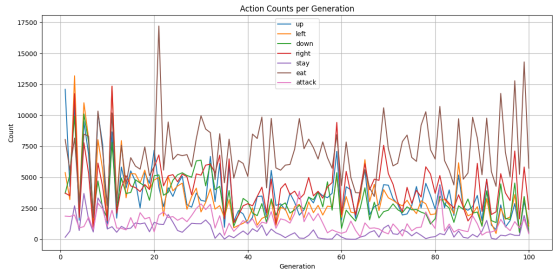


Figure 5: Herbivore action distribution 100x100 world

In the beginning, the actions chosen were randomized, but Figure 5 shows how herbivores learned to prioritize the eating action, although initial interference is evident. The usage of stay and attack actions is low.

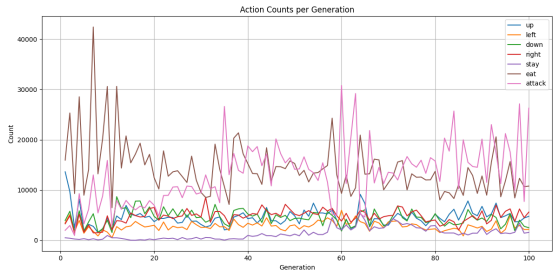


Figure 6: Carnivore action distribution 100x100 world

In Fig.6 we can spot how carnivores experience problems in balancing the eating and attacking action, but the attacking action slightly dominates after some time.

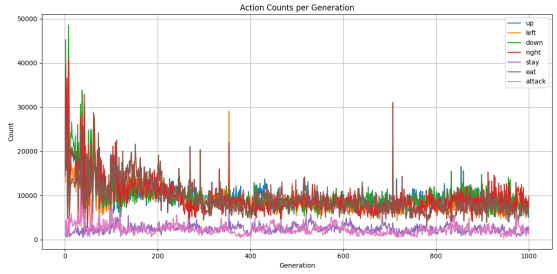


Figure 7: Herbivore action distribution 200x200 world

Figure 7 displays herbivore behavior, where the action distribution is more stable and there is a clear evolved balance of eating and moving actions, which is expected in a larger world.

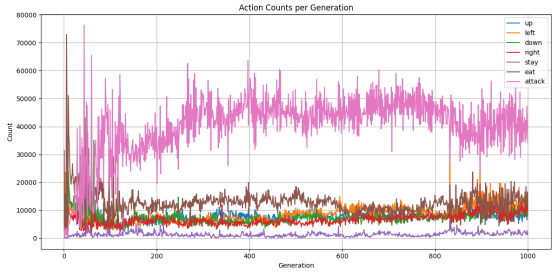


Figure 8: Carnivore action distribution 200x200 world

Figure 8 displays carnivore behavior in the larger world, where they were given a greater incentive to attack. From the distribution, we can see that they indeed attacked more, with the other actions being balanced out, and the staying action was rarely chosen.

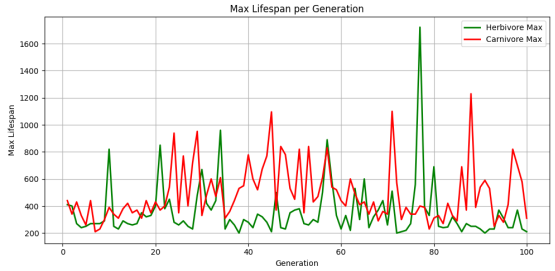


Figure 9: Maximum lifespan in the coevolutionary setting

Fitness (as well as the lifespan depicted in Figure 9) fluctuated in an "arms race" pattern with no dominant winner. This outcome is expected, as the rise in one role's performance lowered the performance of the other. This shows that the system tended toward balance, which aligns with the objective of testing whether coevolution with NEAT agents could produce equilibrium.

3.4 Species Diversity

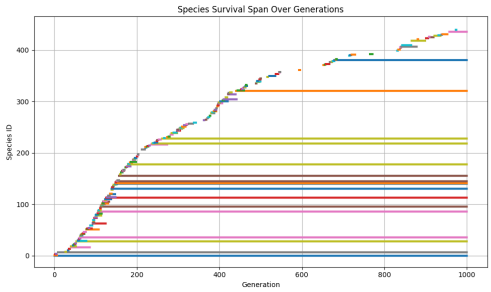


Figure 10: Species diversity over generations.

The survival plot of emerging species in Figure 10 shows an important aspect of the NEAT algorithm. The initial drop means that a few very successful topologies dominated the population, but using a lower compatibility threshold prevents the total loss of diversity. The number of species stabilized after some time, while many smaller species died out quickly.

4 Design Observations

Agent behavior is highly sensitive to design choices in fitness functions, environment setup, and input representation. Poorly designed fitness functions can lead to inefficient or trivial behaviors, such as flickering near food, which highlight the exploration-exploitation trade-off and the role of environmental influence in shaping behavior. Static or predictable resource spawn locations can cause overfitting, where agents memorize positions instead of learning general strategies. Dynamic and unpredictable environments are necessary to evolve general food-seeking behaviors and to observe which patterns emerge due to evolutionary pressures versus environmental conditions. However, environments that are too unpredictable can hinder learning and obscure the distinction between inherited tendencies and environment-induced behaviors.

Input scaling and initial placement also influence behavioral emergence. Unlimited health input caused agents to idle, while spawning agents too close together and awarding them for food consumption led to aimless wandering when neighbors died, showing correlations learned from the environment. These observations demonstrate how neural networks may pick up coincidental patterns that influence both relearning across generations and the adopted strategies.

Metrics did not always reflect consistent progress, as dynamic food spots and starting points introduced noise. Dips or peaks in performance do not necessarily indicate genuine failure or success. Adjustments to fitness, food rewards, and environmental parameters were required to guide learning, prevent reward hacking, and allow behavioral adaptation. Comparing herbivore and carnivore roles shows that behaviors are shaped by both environmental pressures and the interactions between agent strategies and resource availability. Agents adjust their actions based on the resources they encounter, and these actions influence which resources remain available, creating a feedback loop between behavior and the environment.

5 Conclusion and Future Work

This paper demonstrates that nature-like behaviors can emerge from relatively simple principles when agents evolve in dynamic, open-ended environments without predefined goals. By evolving herbivores and predators both separately and in co-evolution, we showed that evolutionary pressures can produce adaptive behaviors and predator-prey equilibria, highlighting how role-specific dynamics shape ecosystem stability. This work lays the foundation for future experiments that involve more complex behaviors, survival strategies, and deeper coevolutionary dynamics. Future directions could include investigating the potential of refined role awareness mechanisms, improved memory or learning retention, and more complex agent inputs and actions, enabling us to push the boundaries of what these agents can learn over time.

References

[1] A.E. Eiben and J.E. Smith. 2003. *Introduction to Evolutionary Computing. Natural Computing*. Springer-Verlag, Berlin.

[2] LibGDX. [n. d.] Libgdx game development framework. https://libgdx.com/. ().

[3] Michael Mccloskey and Neil J. Cohen. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. *The Psychology of Learning and Motivation*, 24, 104–169.

[4] Michael A. Nielsen. 2018. Neural networks and deep learning. misc. (2018). http://neuralnetworksanddeeplearning.com/.

[5] Kenneth O. Stanley and Risto Miikkulainen. 2002. Evolving neural networks through augmenting topologies. *Evolutionary Computation*, 10, 2, 99–127. http://nn.cs.utexas.edu/?stanley:ec02.

Designing AI Agents for Social Media

Abdul Sittar*
Jožef Stefan Institute
Ljubljana, Slovenia
abdul.sittar@ijs.si

Mateja Smiljanić
Jožef Stefan Institute
Ljubljana, Slovenia
mateja.smiljanic@gmail.com

Alenka Guček
Jožef Stefan Institute
Ljubljana, Slovenia
alenka.gucek@ijs.si

Abstract

This work presents an approach for designing AI agents that simulate social media activity by replacing Twitter conversations with large language models (LLMs). Using a time-series dataset of Twitter discussions about technologies (April 2019 - April 2020), we propose an approach that combines fine-tuned language models with timeline manager to capture both conversational dynamics and temporal posting patterns. This approach consists of two main components: 1) a timeline manager, which models posting frequency, reply behaviour, and temporal rhythms of users, and 2) conversation agents, fine-tuned for posting and replying within threads. We evaluate the system along two dimensions: structural accuracy (whether the timeline manager replicates conversation patterns and thread structures), and emotion dynamics (weather the emotion of synthetic data replicates the true emotion trends in the original dataset). Our results demonstrate that the proposed agent-based simulation captures key characteristics of real Twitter interactions, providing a foundation for large-scale synthetic social media ecosystems useful for studying information flow, emotion propagation, and the impact of emerging technologies.

Keywords

AI agents, large language models (LLMs), social media simulation, Twitter conversations, conversation agents

1 Introduction

Social media platforms have become major arenas for information dissemination, discussion, and opinion formation. However, the emergence of filter bubbles where users are exposed predominantly to content that aligns with their existing beliefs can reinforce polarization, reduce diversity of exposure, and shape collective behaviour in unforeseen ways [3]. Also, Social networks have broadened the range of ideas and information accessible to users, but they are also criticized for contributing to greater polarization of opinions [2]. Understanding how these dynamics emerge and evolve requires models that can replicate user behaviour at scale while capturing temporal patterns and interactions.

Large language models have emerged as powerful tools for synthetic text generation. [10] investigated GPT-3.5 for text classification augmentation, finding that subjectivity negatively correlates with synthetic data effectiveness, while achieving 3-26% absolute improvement in accuracy/F1 in low-resource settings. [18] introduced GPT3Mix, using GPT-3 for realistic text generation with soft-labels, significantly outperforming existing augmentation methods. The quality of synthetic data generation has been evaluated through multiple frameworks. [15], [14] emphasized that stylistic consistency within timelines benefits rare event detection, while artificial stylistic variety can increase false positives. [1] demonstrated T5-based paraphrasing effectiveness, achieving average 4.01% accuracy increase with T5 augmentation, with RoBERTa reaching 98.96% accuracy through ensemble approaches.

Recent advances in large language models (LLMs) provide opportunities to simulate social media users as autonomous agents capable of generating posts and replies. [9] mainly concentrates on using LLMs as stand-alone agents or for simple agent interactions, neglecting the opportunity to assess LLMs within the network structure of complex social networks. In this study, we leverage fine-tuned language models to create agents across multiple domains, including technology (AI), cryptocurrency, and health-related topics (e.g., COVID-19). Each agent is specialized for posting or replying, while a timeline manager model simulates the environment, deciding which agent acts next and at what time. By grouping similar users into single agents, our approach generalizes behaviour while maintaining the richness of interaction patterns.

The main goal of this work is to investigate the effect of environmental changes on agent behaviour and network dynamics. Specifically, we hypothesize that altering the scheduling and structure of the environment model can lead to measurable changes in posting and replying activity, as well as in the temporal evolution of simulated emotions. To evaluate our approach, we compare real Twitter data with simulated outputs, analysing emotion trends and interaction dynamics across time windows. Our approach provides a novel methodology for studying social media dynamics, testing hypotheses about user behaviour, and exploring interventions to mitigate filter bubbles.

1.1 Contributions

Following are the two primary scientific contributions of this work:

- An approach to replicate social media users by grouping similar users into language model-driven agents managed with a timeline manager
- An evaluation that assesses structural accuracy, conversational coherence, and emotional realism by comparing simulated and true emotion trends.

Information Society 2025, Ljubljana, Slovenia

https://doi.org/10.70314/is.2025.sikdd.23

2 Related Work

LLMs are increasingly employed to model human behaviour in online settings, but current evaluation approaches such as simplified Turing tests involving human annotators fail to capture the subtle stylistic and emotional nuances that differentiate human generated text from AI-generated text [12]. It proposes a human likeness evaluation framework that systematically measures how closely LLM generated social responses resemble those of real users. This framework utilizes a set of interpretable textual features that capture stylistic, tonal, and emotional aspects of online conversations. While they can mimic certain human behaviours and decision making processes, primarily due to their training data, it remains largely unexplored whether repeated interactions with other agents amplify their biases or lead to exclusive patterns of behaviour over time [8].

Modelling social media has ben an active research area for understanding use behaviour, information diffusion, and network effects. Agent-based models have been widely used to replicate interactions among users, simulate posting and replying behaviour, and study emergent phenomena such as viral content spread, echo chambers, and filter bubbles [6, 11]. These models often rely on simplified rules or probabilistic mechanisms to determine agent actions. Our work extends this by using fine-tuned language model to generate realistic post and reply content, capturing both semantic and temporal patterns observed in real social media interactions.

The concept of filter bubbles has been extensively studied in the context of social media algorithms and personalized content delivery [17, 7, 3]. Prior studies have shown that temporal factors, such as posting frequency and timing, significantly influence the formation of echo chambers and the propagation of sentiment. Unlike traditional simulations, our approach explicitly models time windows and agent-specific schedules, allowing the study of how environmental changes affect network dynamics and user behaviour over time.

Large language models (LLMs) have been increasingly applied to social media analysis, content generation, and user simulation. Fine-tuned models can capture domain-specific language, hashtags, and posting patterns, enabling more realistic simulations of user behaviour [13, 4]. Existing work has largely focused on generating content for individual posts or replies; in contrast, our approach integrates posting, replying, and environment management in a unified simulation, enabling multi-agent interaction analysis.

Recent studies have used sentiment and emotion analysis to evaluate social media content, including the study of affective trends and collective mood in online networks [16, 5]. Our approach leverages these techniques to compare simulated emotion trends with real-world Twitter data, providing a quantitative measure to validate the fidelity of the agent-based simulation.

3 Methodology

Our methodology employs a two stage approach combining probabilistic scheduling with domain-specialized fine-tuned language model agents to simulate realistic social media interactions (posting and replying). The approach consists of two primary components: (1) Timeline based probabilistic model that serves as an timeline manager, and (2) Domain-specialized fine-tuned agents that generate contextually appropriate content based on the timeline manager's decisions.

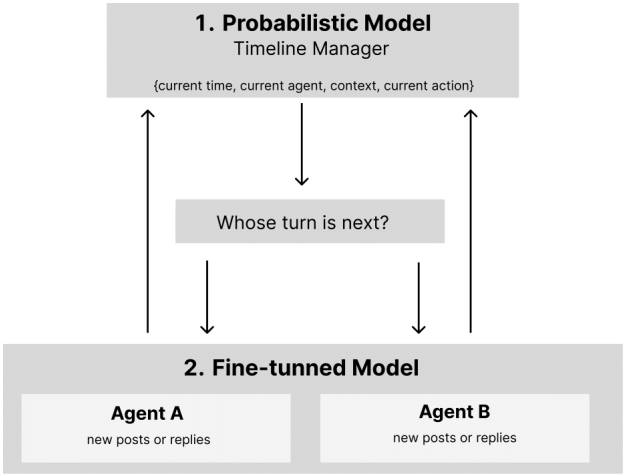


Figure 1: Overview of the proposed methodology for conversation simulation. The timeline manager determines which agent should act next based on the current time, agent, context, and action. The selected fine-tuned model then generates a new post or reply for the chosen agent, creating realistic conversation flow.

3.1 Probabilistic model

The probabilistic scheduler is implemented as a multi-output neural network that simultaneously predicts four key dimensions of social media behaviour: agent selection (which agent should act next), action classification (post vs. reply), temporal prediction (timing of next action), and context setting (emotional tone and topical focus for content generation).

The model is trained on 88,330 conversation items spanning April 2019 to April 2020, focusing on AI and cryptocurrency discussions. Our Timeline-Based approach generates 93,440 chronological training pairs—18.7× more than baseline methods—through complete conversation sequence learning rather than isolated post-reply pairs.

Given the current state $S(t)$ at time t, the model computes probability distributions over the action space.

3.2 Fine-tuned model

We implement a single fine-tuned language model that serves as both AI and cryptocurrency agents. The model is trained on conversations from both domains (AI technology and cryptocurrency discussions) to capture the vocabulary, argumentation patterns, and discourse styles across both topic areas.

- **Agent A (AI Focus)**: The same fine-tuned model called when the probabilistic scheduler determines AI-related content is needed.
- **Agent B (Crypto Focus)**: The identical fine-tuned model called when cryptocurrency-related content generation is required.

When called by the probabilistic scheduler, the fine-tuned model generates content based on provided context including action type (post/reply), emotional context, topical focus, temporal context, and conversation history. The model's training on both domains enables it to produce contextually appropriate responses regardless of which agent role it is fulfilling.

3.3 Integration and Coordination

The probabilistic scheduler communicates with fine-tuned agents through a structured interface that maintains separation between temporal decisions (when and who acts) and content decisions (what is said). At each simulation step, the scheduler: (1) analyses current conversation state, (2) predicts next action parameters, (3) selects appropriate domain agent, (4) provides structured context to the selected agent, and (5) integrates generated content into the conversation thread.

This approach enables realistic conversations where different domain experts can contribute to mixed topic discussions while maintaining their specialized perspectives and temporal behavioural patterns observed in real social media data.

4 Experimental Setup

In this section, we describe the features, model and evaluation metrics.

4.1 Timeline Manager

The baseline system is a timeline based probabilistic model that learns agent transitions, reply probabilities, and temporal distributions from training data. Predictions are made deterministically by selecting the most probable outcome, with probability estimates derived directly from observed frequencies.

The enhanced approach employs a machine learning ensemble with separate classifiers for agent, action, and time prediction. Features include agent history, action history, and time of day. Predictions are generated using temperature-controlled stochastic sampling, with an ensemble across multiple temperature settings for robustness. This design enables greater flexibility and diversity, counteracting the strong biases inherent in the probabilistic model.

4.1.1 Evaluation Metrics. Table 1 summarizes the key differences between the original probabilistic model and the improved ML-based model, covering both quantitative performance and qualitative conversational outcomes.

Aspect	Probabilistic Model	ML-Based Model
Agent Prediction	44.8% accuracy, but always predicts Crypto_Agent (100%)	55.2% accuracy, balanced AI_Agent (50%) and Crypto_Agent (50%)
Action Prediction	74.4% accuracy by predicting only "post" (0% replies)	67.8% accuracy with realistic mix: 65% posts / 35% replies (close to ground truth 73/27)
Temporal Modelling	MAE = 5.41 min; 99.4% within ±15 min	MAE = 7.11 min; 99.2% within ±15 min

Table 1: Comparison of the Original Probabilistic Model vs. the Improved ML-Based Model.

we evaluated our probabilistic model using comprehensive metrics across three key categories:

- **Agent Prediction:** 61.3% accuracy (22.6% improvement over random chance)
- **Action Classification:** 96.8% accuracy for post vs. reply prediction
- **Temporal Modelling:** 50.7-minute MAE with 99.15% accuracy within ±15 minutes

Our evaluation demonstrates that the probabilistic scheduler successfully replicates conversation structure:

- **Agent Alternation:** 94.2% similarity to real switching behaviours
- **Temporal Rhythms:** Strong correlation (r=0.78) with actual daily patterns
- **Action Distribution:** Maintains realistic post/reply ratios (94.5%/5.5%)

4.2 Fine-tuned model

Table 2: Evaluation Results: ROUGE and Semantic Similarity

Metric	Score
ROUGE-1	0.1373
ROUGE-2	0.0519
ROUGE-L	0.1179
ROUGE-Lsum	0.1217
Semantic Similarity (SBERT)	0.4041

Table 2 reports the evaluation results for the fine-tuned model's generated content. ROUGE metrics (ROUGE-1, ROUGE-2, ROUGE-L, and ROUGE-Lsum) measure lexical overlap between generated outputs and the reference Twitter posts. The relatively low scores (e.g., ROUGE-1 = 0.1373) indicate that while the generated text captures some overlapping words or phrases, it often diverges lexically from the original references. This is expected since the model is not designed for verbatim reproduction but rather for generating semantically coherent alternatives.

To complement ROUGE, we compute semantic similarity using SBERT embeddings. The score of 0.4041 shows that, on average, the generated outputs are moderately aligned in meaning with the reference texts, even when surface-level wording differs. This highlights that the fine-tuned model is able to remain contextually and thematically relevant while producing novel expressions.

Overall, the combination of ROUGE and semantic similarity suggests that the fine-tuned agents generate content that does not simply replicate reference posts but instead produces new, semantically consistent outputs.

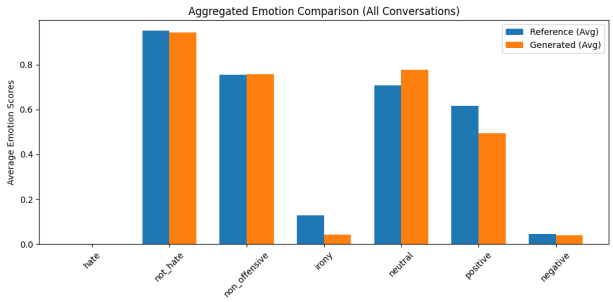


Figure 2: Methodology diagram showing both experimental approaches: First step, second step, third step, fourth step

Figure 2 presents the aggregated emotion comparison between the reference Twitter dataset and the conversations generated by the fine-tuned model. The analysis is based on average emotion scores across multiple conversation samples, with categories including hate, not_hate, non_offensive, irony, neutral, positive, and negative. Blue bars represent the reference data, while orange bars indicate the generated outputs.

Overall, the comparison shows strong alignment between the two distributions for key non-toxic categories. Both reference and generated conversations are overwhelmingly classified as

not_hate and non_offensive, with nearly identical scores (approximately 0.95 and 0.75, respectively). Similarly, both datasets contain minimal hate or negative content, indicating that the synthetic conversations do not introduce harmful patterns absent from the real data.

At the same time, certain emotional discrepancies are evident. The generated conversations exhibit lower levels of irony and positivity compared to the real dataset. Specifically, irony is notably under-represented in synthetic conversations (0.04 versus 0.12 in the reference data), suggesting that nuanced and implicit language styles are harder for the model to reproduce. Similarly, positive sentiment is reduced in generated text (0.49 versus 0.62), while neutrality is slightly higher (0.78 versus 0.71). This indicates a tendency of the model to produce emotionally flatter and less expressive outputs.

Taken together, the results suggest that the model successfully replicates the broad emotional structure of conversations, particularly in terms of avoiding toxic or offensive content. However, the generated outputs are less emotionally rich than real data, with reduced representation of irony and positivity. This highlights a key limitation of current LLM-based conversation agents: while structurally sound, they may generate interactions that are less engaging or authentic in their emotional dynamics.

5 Conclusions

In this work, we presented a novel approach for replicating social media user behaviour using fine-tuned language models organized as autonomous agents. By combining a timeline manager (Model A) with specialized posting (Model B) and replying (Model C) models, we simulated realistic multi-agent interactions across AI and Crypto related topics.

Our timeline based probabilistic model successfully replicates structural conversation patterns with 61.3% agent accuracy and near-perfect action classification (96.8%), establishing a new benchmark while providing clear paths for further enhancement through domain specialization.

Our experiments demonstrated that the approach can generate temporal posting and replying patterns that closely resemble real-world Twitter data. We showed that modifying the environment model significantly influences agent behaviour, posting frequency, and network dynamics, supporting our hypothesis that environmental and temporal factors shape interaction patterns in social networks.

This approach provides a flexible and controlled platform for studying filter bubble formation, emotion propagation, and emergent social dynamics. Future work can extend the approach to more complex network structures, additional domains, and the integration of user-specific behaviour models to further explore interventions for mitigating echo chambers and enhancing diversity in online interactions.

6 Acknowledgment

The research presented in this paper was funded by the EU's Horizon Europe Framework under grant agreement number 101095095 (TWON) and 101094905 (AI4Gov).

References

[1] Jordan J. Bird et al. 2021. Chatbot interaction with artificial intelligence: human data augmentation with t5 and language transformer ensemble for text classification. *arXiv preprint arXiv:2010.05990*.

[2] Uthsav Chitra and Christopher Musco. 2020. Analyzing the impact of filter bubbles on social network polarization. In *Proceedings of the 13th international conference on web search and data mining*, 115–123.

[3] Uthsav Chitra and Christopher Musco. 2019. Understanding filter bubbles and polarization in social networks. *arXiv preprint arXiv:1906.08772*.

[4] Cristina Chueca Del Cerro. 2024. The power of social networks and social media's filter bubble in shaping polarisation: an agent-based model. *Applied Network Science*, 9, 1, 69.

[5] Matteo Cinelli, Gianmarco De Francisci Morales, Alessandro Galeazzi, Walter Quattrociocchi, and Michele Starnini. 2020. Echo chambers on social media: a comparative analysis. *arXiv preprint arXiv:2004.09603*.

[6] Rui Fan, Ke Xu, and Jichang Zhao. 2018. An agent-based model for emotion contagion and competition in online social media. *Physica a: statistical mechanics and its applications*, 495, 245–259.

[7] Antonino Ferraro, Antonio Galli, Valerio La Gatta, Marco Postiglione, Gian Marco Orlando, Diego Russo, Giuseppe Riccio, Antonio Romano, and Vincenzo Moscato. 2024. Agent-based modelling meets generative ai in social network simulations. In *International Conference on Advances in Social Networks Analysis and Mining*. Springer, 155–170.

[8] Farnoosh Hashemi and Michael Macy. 2025. Collective social behaviors in llms: an analysis of llms social networks. In *Large Language Models for Scientific and Societal Advances*.

[9] Tianrui Hu, Dimitrios Liakopoulos, Xiwen Wei, Radu Marculescu, and Neeraja J Yadwadkar. 2025. Simulating rumor spreading in social networks using llm agents. *arXiv preprint arXiv:2502.01450*.

[10] Z. Li, J. Zhu, et al. 2023. Synthetic data generation with large language models for text classification: potential and limitations. *arXiv preprint arXiv:2310.07849*.

[11] Hamid Reza Nasrinpour, Marcia R Friesen, et al. 2016. An agent-based model of message propagation in the facebook electronic social network. *arXiv preprint arXiv:1611.07454*.

[12] Nicolò Pagan, Petter Törnberg, Christopher Bail, Ancsa Hannak, and Christopher Barrie. [n. d.] Can llms imitate social media dialogue? techniques for calibration and bert-based turing-test. In *First Workshop on Social Simulation with LLMs*.

[13] Kayhan Parsi and Nanette Elster. 2015. Why can't we be friends? a case-based analysis of ethical issues with social media in health care. *AMA journal of ethics*, 17, 11, 1009–1018.

[14] Ifrah Pervaz, Iqra Ameer, Abdul Sittar, and Rao Muhammad Adeel Nawab. 2015. Identification of author personality traits using stylistic features: notebook for pan at clef 2015. In *CLEF (Working Notes)*, 1–7.

[15] E. Rosenfeld et al. 2025. Evaluating synthetic data generation from user generated text. *Computational Linguistics*, 51, 1, 191–230.

[16] Tanase Tasente. 2025. Understanding the dynamics of filter bubbles in social media communication: a literature review. *Vivat Academia*, 1–21.

[17] Petter Törnberg, Diliara Valeeva, Justus Uitermark, and Christopher Bail. 2023. Simulating social media using large language models to evaluate alternative news feed algorithms. *arXiv preprint arXiv:2310.05984*.

[18] Kang Min Yoo et al. 2021. Gpt3mix: leveraging large-scale language models for text augmentation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, 2225–2239.

Explaining Temporal Data in Manufacturing using LLMs and Markov Chains

Jan Šturm
jan.sturm@ijs.si
Jožef Stefan Institute
Jožef Stefan International Postgraduate School
Ljubljana, Slovenia

Maja Škrjanc
maja.skrjanc@ijs.si
Jožef Stefan Institute
Jožef Stefan International Postgraduate School
Ljubljana, Slovenia

Oleksandra Topal
oleksandra.topal@ijs.si
Jožef Stefan Institute
Ljubljana, Slovenia

Inna Novalija
inna.koval@ijs.si
Jožef Stefan Institute
Ljubljana, Slovenia

Dunja Mladenić
dunja.mladenic@ijs.si
Jožef Stefan Institute
Jožef Stefan International Postgraduate School
Ljubljana, Slovenia

Marko Grobelnik
marko.grobelnik@ijs.si
Jožef Stefan Institute
Ljubljana, Slovenia

Abstract

Monitoring and understanding complex industrial processes from high-dimensional IoT sensor data remains a significant challenge. While advanced modeling techniques like Hierarchical Markov Chains can abstract raw data, their outputs are often difficult for domain experts to interpret, creating a gap between data-driven insights and operational management. Existing explainability methods often focus on feature importance rather than providing holistic, semantic descriptions of system states. This paper introduces a framework that bridges this gap by transforming the abstract states of a process model into intuitive, human-readable concepts. The methodology leverages the StreamStory (Hierarchical Markov Chain) tool approach to generate behavioral profiles based on log-likelihood calculations within sliding temporal windows. StreamStory states are summarized using an LLM to assign semantic labels and descriptions. This approach reduces the initial reliance on domain experts for analysis, aids the understanding of complex system dynamics, and provides a transparent foundation for identifying both normal and anomalous operational patterns. The result is a more interpretable representation of industrial processes, facilitating improved predictive maintenance and operational efficiency.

Keywords

Multivariate Timeseries, Explainable AI, LLMs, Markov Chains

1 Introduction

The widespread adoption of Internet of Things (IoT) sensors in industrial environments has generated vast streams of multivariate time-series data. While this data holds immense potential for process optimization and predictive maintenance, its complexity often surpasses human cognitive capacity. Tools like StreamStory [6] have emerged to model these complex systems using Hierarchical Markov Chains, abstracting raw data into a more manageable set of states and transitions. However, a fundamental challenge persists: a disconnect between the model's statistical outputs and the experiential knowledge of domain experts.

The motivation for this work stems from this challenge. Domain experts, who possess invaluable implicit knowledge of a system, often struggle to interpret the statistical outputs of process models. Conversely, data scientists may identify patterns that lack the necessary operational context for effective action. Presenting experts with a graphical representation of states and transitions is a step forward, but it does not fully bridge the semantic gap. They may not understand what a specific state represents in the physical world or why a particular transition is significant. This leads to a bottleneck where valuable data-driven insights are not fully utilized, hindering efforts to improve system management and efficiency.

To address this, the paper proposes a methodology that enhances the interpretability of hierarchical process models. This approach creates a new layer of understanding that is accessible to operational personnel without requiring deep data science expertise. By translating abstract model states into meaningful, semantically rich descriptions, it provides a tool that allows the system's behavior to be understood, validated, and ultimately, better managed. This work introduces a methodology to automatically generate these descriptions, moving from complex data to clear, actionable insights. This work presents two primary contributions for industrial applications: a method for LLM-based labeling of Markov chain states, and a methodology for identifying events as anomalous or normal.

2 Related Work

The field of time-series anomaly detection has evolved from interpretable statistical models like ARIMA and classical machine learning such as Isolation Forest to high-performance deep learning architectures including LSTMs, Transformers, and Autoencoders [5, 4, 7]. While these advanced models excel at pattern recognition, their complexity necessitates post-hoc XAI tools like LIME and SHAP to explain their decisions, which are limited to providing low-level feature attributions [1].

Recent work also demonstrates the utility of Hidden Markov Models (HMMs) for anomaly detection, for instance, by designing active search strategies to locate an evolving anomaly among multiple processes [2], or by learning normal temporal dynamics from remote sensing data to detect, localize, and classify crop-related deviations [3]. However, while effective for detection, the

Information Society 2025, 6–10 October 2025, Ljubljana, Slovenia

https://doi.org/10.70314/is.2025.sikdd.28

abstract nature of HMM states can be difficult for domain experts to interpret. The present work addresses this by transforming the state sequence into a multi-scale behavioral profile, which enables a Large Language Model (LLM) to generate rich, semantic explanations of system behavior.

This approach innovates by first classifying each multivariate data point into a state within a pre-built Markov Chain model and then calculating log-likelihoods from the state sequence to form a multi-scale representation. Crucially, this representation allows for the recognition of regular system behavior and various anomalies. By analyzing the statistical distribution of these profiles—identifying dense regions of regular behavior and sparse outliers corresponding to anomalous states—an LLM can then assign rich, human-readable descriptions, connecting abstract data to operational knowledge.

3 Methodology

The framework is designed to post-process models generated by the StreamStory system. Figure 1 outlines this multi-stage process, which begins with the statistical features from the Markov model and culminates in semantically enriched explanations of system behavior. The core of this methodology is the transformation of abstract machine states into meaningful concepts using a combination of statistical feature engineering and LLM interpretation. The process focuses on creating robust representations of system behavior and leveraging an LLM to translate these representations into human-understandable language.

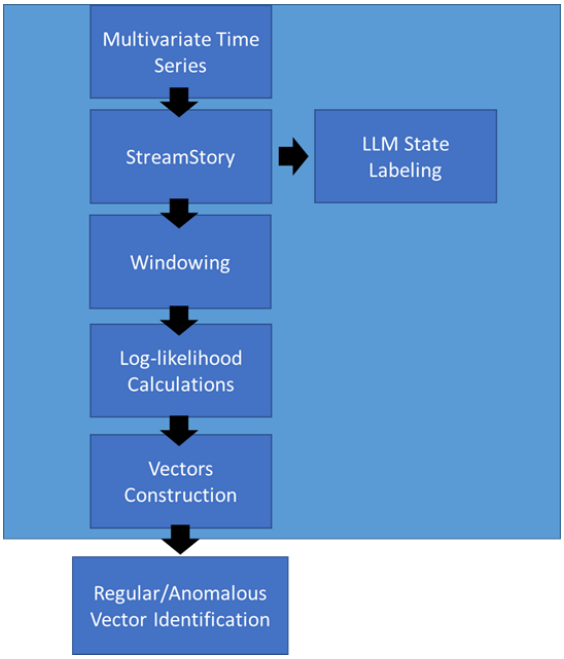


Figure 1: Proposed methodology for identifying and explaining normal and anomalous operational profiles.

3.1 Log-Likelihood Score Calculation

The input to the pipeline is a pre-existing Hierarchical Markov Chain model of an industrial process, which includes a history of state transitions over time. The first step is to create a rich feature representation that captures the system's dynamics. A sliding window (Figure 2) approach moves across the sequence of historical state transitions. For each window of a given size, a single feature is calculated: the log-likelihood of that specific sequence of transitions occurring. This score is calculated by summing the log-transformed transition probabilities for each step in the sequence, as defined by the underlying Markov model. The score effectively quantifies how "normal" or "expected" a particular sequence of behavior is according to the learned model. Highly probable sequences yield higher log-likelihood scores (closer to zero), while rare sequences result in large negative scores.

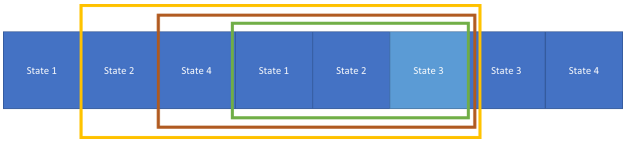


Figure 2: An illustration of the sliding window method. Three windows of different sizes, highlighted in yellow (largest), brown (medium), and green (smallest), are applied to a sequence of system states. A log-likelihood score is then calculated for the sub-sequence contained within each colored window.

3.2 Behavior Profile Construction

To capture dynamics over multiple time scales, several sliding windows of different sizes are used simultaneously. The log-likelihood score calculated from each window is concatenated to form a single feature vector for each time step. This multi-scale vector, termed a *behavior profile*, serves as a rich representation of the system's dynamics at that moment, encapsulating both short-term and longer-term patterns. This profile is a crucial output, as it provides a quantitative basis for distinguishing between different modes of operation.

3.3 Ranking System Behavior via Anomaly Scoring

Following the construction of the behavior profiles, their distribution is analyzed to identify distinct operational patterns. An unsupervised density-based approach is employed to score each profile's typicality. The Isolation Forest algorithm is used for this purpose because it does not assume a specific data distribution and excels at identifying outliers in a high-dimensional space. Profiles that are common and lie in dense regions of the feature space receive a high score, corresponding to *normal* behavior. Conversely, profiles that are rare and isolated receive a low score, flagging them as *anomalous*. This produces a continuous spectrum of normalcy, allowing for a ranked analysis of all operational events.

3.4 LLM-Powered State Naming and Interpretation

To translate abstract states into meaningful concepts, an LLM is utilized. For each granular state discovered by the StreamStory model, its statistical profile (e.g., sensor value distributions) and context about the machine type were formatted into a descriptive prompt. The LLM was then tasked with generating a concise, intuitive name for each state (e.g., "Peak Production - High Flow and Heat"). This process, conducted once per model, creates a semantic layer that is then used to interpret the sequences

associated with the highest-ranked normal and lowest-ranked anomalous events.

This approach offers two key advantages. First, the LLM-generated names provide a layer of transparency, offering an immediate hypothesis about what each abstract state represents. Second, it shifts the role of the domain expert from the arduous task of initial interpretation to the more efficient step of validating or refining the LLM-generated labels, accelerating the process of gaining actionable insights.

4 Experiment

To validate the proposed framework, an experiment was conducted using a real-world industrial dataset from an oil refinery pump. This section details the dataset, implementation, and results.

4.1 Dataset

The experiment was performed on a proprietary, real-world dataset obtained from an industrial oil refinery. Due to its confidential nature, the dataset is not publicly available. The data consists of a multivariate time-series collected over one month of operation (March-April 2017) with a 15-minute sampling resolution. Data was gathered from a suite of IoT sensors monitoring the core functions of a critical pump. Key measurements include fluid flow rate (Kg/h), suction and discharge pressure (Kg/cm2), and temperatures of the process fluid and mechanical components (°C).

4.2 Implementation Details

The methodology was implemented in a Python environment. The underlying Markov Chain model was built using the entire historical dataset provided, as the goal is to interpret the complete, learned dynamics of the process rather than to perform a predictive task that would require a train/test split. Behavior profiles were constructed using sliding windows of multiple sizes (3, 5, 7, and 10 steps). The resulting profiles were analyzed using the Scikit-learn implementation of Isolation Forest. The ‘contamination‘ parameter was set to 5% for the primary analysis, a common heuristic for industrial processes. State descriptions were generated using the GPT-4o model, which was prompted with the statistical profiles of each state to generate intuitive names.

4.3 Experimental Results and Discussion

The application of the framework yielded a ranked list of operational events, characterized by the Isolation Forest decision score. This score serves as a robust indicator of how typical or anomalous a given time window is. Table 1 details the top five most anomalous events identified. These events are characterized by scores that are more than 3 standard deviations below the mean, signifying extreme statistical rarity.

The true explanatory power of the method is revealed when the abstract state sequences are translated into their LLM-generated names. For instance, the most anomalous event culminates in a sequence of “... -> ‘Startup or Shutdown Transition‘ -> ‘Machine Idle or Shutdown‘ -> ‘Startup or Shutdown Transition‘.” This provides a clear, human-readable narrative of the pump entering a period of instability and stoppage. This is a marked improvement over black-box models that simply flag a time point as anomalous without providing a temporal context for the "why." An engineer, seeing this semantic sequence, can immediately infer a potential cause for investigation, such as an attempted restart or a stuttering shutdown process.

Conversely, the most normal events, detailed in Table 2, paint a picture of operational stability. These events are characterized by positive scores. The LLM-generated names for these sequences, such as transitions between ‘Weekday Peak Performance‘, ‘Weekend Peak-Load Production‘, describe the system operating within its expected high-performance period. This demonstrates the framework’s ability not only to flag deviations but also to recognize and semantically label the system’s healthy, predictable operational cycles, providing a valuable baseline for what constitutes ’good’ performance.

Table 1: Top 5 Most Anomalous Events

Rank	Timestamp	Score (Std.)	Final State (LLM Name)
1	2017-04-03 14:30	-0.096 (-3.88)	Startup...Transition
2	2017-03-28 10:00	-0.071 (-3.45)	Startup...Transition
3	2017-03-30 00:00	-0.066 (-3.35)	High-Flow, Cool Op.
4	2017-04-03 12:30	-0.061 (-3.26)	Machine Idle
5	2017-04-03 15:00	-0.056 (-3.18)	Weekday Low-Flow...

Conversely, Table 2 presents the five most normal events, which have high positive scores. Their sequences reveal a stable operational loop between states like “Peak Production,” “Weekend Peak-Load Production,” and “Extreme Temperature Peak Performance.” This recurring pattern defines the pump’s healthy operational "heartbeat," providing a data-driven "golden standard" for normal behavior under demanding conditions. This semantic understanding is crucial for operators, as it validates that the system is performing as expected.

Table 2: Top 5 Most Normal Events

Rank	Timestamp	Score (Std.)	Final State (LLM Name)
1	2017-03-23 22:00	0.192 (1.22)	Weekend Peak-Load
2	2017-03-31 06:00	0.192 (1.22)	Peak Production
3	2017-04-01 00:00	0.191 (1.20)	Peak Production
4	2017-03-31 23:30	0.191 (1.19)	Weekday Peak Perf.
5	2017-03-31 07:30	0.190 (1.17)	Weekday Peak Perf.

To ensure the robustness of the findings, a sensitivity analysis was conducted on the Isolation Forest ‘contamination‘ parameter, testing values of 1%, 5%, and 10%. While the number of points labeled ’Anomalous’ changed as expected, the relative ranking of the most extreme events remained highly consistent, confirming that the core findings are not sensitive to this hyperparameter.

The claims in this paper are demonstrated on a single, representative dataset. While the framework is designed to be general, further studies on diverse industrial processes are required to fully validate its broader applicability. The LLM-generated labels were not validated in a formal user study with domain experts; such a study is a valuable next step.

5 Conclusion

This paper presented a complete, self-contained framework for increasing the interpretability of complex industrial process models. By creating behavior profiles of system states and using an

LLM to assign semantic names, the approach successfully translates abstract data analysis into practical domain knowledge. The method provides a robust process for ranking and explaining individual operational events in a transparent manner, as demonstrated on a real-world industrial dataset. This work establishes a strong foundation for a new type of explainability, moving beyond feature importance to provide narrative, context-rich descriptions of system dynamics.

The representation of system dynamics as behavior profiles opens a wide array of possibilities for future research. The current work successfully identifies and presents the raw temporal sequences leading to key events. Future work will focus on applying formal pattern mining techniques to automatically discover recurring and significant sequential patterns within these events. Such an analysis could reveal if distinct "families" of anomalous behavior exist, each with its own characteristic temporal signature. This promises a more nuanced description of system operations and provides a stronger foundation for developing targeted predictive maintenance strategies. Finally, to address current limitations, two key areas will be prioritized. First, formal user studies with domain experts will be conducted to validate the utility and accuracy of the LLM-generated explanations, moving beyond the promising initial results. Second, the framework's generalizability will be tested through broader empirical evaluation across diverse industrial sectors and sensor types to boost its credibility and applicability.

6 Acknowledgments

This work was supported by the Slovenian Research Agency and the European Union's Horizon 2020 project FAME (Grant No. 101092639).

References

[1] Liat Antwarg, Ronnie Mindlin Miller, Bracha Shapira, and Lior Rokach. 2019. Explaining anomalies detected by autoencoders using shap. *arXiv preprint arXiv:1903.02407*.

[2] Levli Citron, Kobi Cohen, and Qing Zhao. 2025. Searching for a hidden markov anomaly over multiple processes. *arXiv preprint arXiv:2506.17108*.

[3] Kareth M Leon-Lopez, Florian Mouret, Henry Arguello, and Jean-Yves Tourneret. 2021. Anomaly detection and classification in multispectral time series based on hidden markov models. *IEEE transactions on geoscience and remote sensing*, 60, 1–11.

[4] Sebastian Schmidl, Phillip Wenig, and Thorsten Papenbrock. 2022. Anomaly detection in time series: a comprehensive evaluation. *Proceedings of the VLDB Endowment*, 15, 9, 1779–1797.

[5] Charalampos Shimillas, Kleanthis Malialis, Konstantinos Fokianos, and Marios M Polycarpou. 2025. Transformer-based multivariate time series anomaly localization. In *2025 IEEE Symposium on Computational Intelligence on Engineering/Cyber Physical Systems (CIES)*. IEEE, 1–8.

[6] Luka Stopar, Primoz Skraba, Marko Grobelnik, and Dunja Mladenic. 2018. Streamstory: exploring multivariate time series on multiple scales. *IEEE transactions on visualization and computer graphics*, 25, 4, 1788–1802.

[7] Fengling Wang, Yiyue Jiang, Rongjie Zhang, Aimin Wei, Jingming Xie, and Xiongwen Pang. 2025. A survey of deep anomaly detection in multivariate time series: taxonomy, applications, and directions. *Sensors (Basel, Switzerland)*, 25, 1, 190.

Active Learning for Power Grid Security Assessment: Reducing Simulation Cost with Informative Sampling

Gašper Leskovec
Jožef Stefan Institute
Slovenia
leskovecg@gmail.com

Costas Mylonas
UBITECH
Greece
kmylonas@ubitech.eu

Klemen Kenda
Jožef Stefan Institute
Slovenia
klemen.kenda@ijs.si

Abstract

Power grid security assessment under the N-1 criterion requires extensive contingency simulations, which are computationally intensive and costly to label. In this work, we explore the use of active learning (AL) to train binary classifiers that can accurately predict the outcome of contingency scenarios using fewer labeled samples. We evaluate several AL strategies, such as entropy, margin, and uncertainty sampling against a random baseline. Our results show that AL methods achieve the same predictive performance with significantly fewer labels, reducing labeling effort and simulator runtime. These findings demonstrate the effectiveness of integrating AL with power system simulators to enable scalable and efficient N-1 security assessment without sacrificing model accuracy.

Keywords

active learning, smart grids, security assessment, simulation cost reduction

1 Introduction

Ensuring secure operation of power systems under the N-1 criterion is a cornerstone of grid reliability. The criterion requires that the system remains within operational limits following the loss of any single component (e.g., line, transformer, or generator). In practice, this involves simulating a large number of contingencies and checking for violations of thermal or voltage constraints. While essential, such simulations are computationally intensive, particularly when performed on high-fidelity grid models, and their interpretation often requires expert judgment. This creates a bottleneck for both real-time applications and large-scale scenario analyses, where scalability and efficiency are important.

Classical approaches to N-1 assessment rely on exhaustive AC power flow simulations combined with contingency ranking heuristics such as performance indices (PIs). While useful for screening, these heuristics may mis-rank contingencies or overlook borderline cases due to masking effects [3]. Moreover, exhaustive analysis does not scale well with system size, making it unsuitable for fast or repeated assessments.

To overcome these challenges, researchers have proposed machine learning (ML) and deep learning (DL) approaches that approximate N-1 contingency outcomes directly from operating point features. One of the earliest contributions in this direction applied convolutional neural networks (CNNs) to contingency datasets, showing that deep models could achieve over 99% accuracy in detecting insecure cases while being more than 200 times faster than traditional power flow calculations [1]. Building on this, more recent work explored pooling-ensemble multi-graph learning to design scalable contingency screening schemes based on steady-state information, demonstrating improved adaptability for large-scale systems [2]. These approaches enable fast security screening without solving power flows for every contingency. However, their reliability hinges on the availability of large labeled datasets covering all relevant operating points and contingencies. Such datasets are typically generated by running exhaustive offline N-1 simulations, which is computationally expensive, or require significant expert effort to label secure versus insecure cases. This dependence on costly and large-scale data generation remains a major limitation of existing ML-based frameworks for steady-state security assessment.

To reduce labeling costs, AL has recently been explored in other areas of power systems. For example, authors of [5] used AL to enhance stability assessment and dominant instability mode identification, showing that models could be trained with far fewer labeled samples while maintaining accuracy. Similarly, authors of [4] demonstrated an AL-enhanced digital twin for day-ahead load forecasting, where the model iteratively refined predictions by querying only the most uncertain cases. These studies confirm the potential of AL to reduce expert effort and simulation cost by strategically selecting informative samples. However, AL has not yet been applied to N-1 steady-state security assessment, where the need to cut down on contingency simulations is especially critical.

In this work, we propose a novel framework for AL driven N-1 security assessment. Our contributions are threefold:

(1) We design a binary classification model that predicts whether a given contingency is secure or insecure based on steady-state features.
(2) We integrate AL strategies (entropy, margin, and uncertainty sampling) with the classifier to selectively query the most informative contingencies for simulation, reducing the number of labels required.
(3) We demonstrate through a case study that our approach achieves the same predictive accuracy as fully supervised baselines while reducing simulation cost and labeling effort by up to 40–50%.

This work provides the first evidence that AL can be directly leveraged for N-1 security assessment, offering a scalable and label-efficient alternative to exhaustive simulation or purely supervised ML approaches.

2 Methodology

We study whether pool-based AL can reduce the number of expensive N-1 simulations ("labels") while keeping prediction quality for binary *secure vs. insecure* classification.

SiKDD 2025, Ljubljana, Slovenia

ACM ISBN 978-x-xxxx-xxxx-x/YYYY/MM
https://doi.org/10.70314/is.2025.sikdd.11

Table 1: Dataset and system description (digital twin of the Greek transmission network).

Attribute	Value
Test system	35 buses, 46 lines, 135 generators, 110 static generators, 20 loads
Power flow solver	AC load flow (Newton–Raphson), via `pandapower`
Contingencies (N–1)	Line outages (all lines except idx 45), generator outages (all)
Total contingency cases	8 769
Secure / Insecure	51.28% / 48.72%
Feature dimensionality	271 features total
Feature groups	`load_`: 20, `gen_`: 135, `sgen_`: 110

2.1 Data and labels from a digital twin

We use a steady-state digital twin of the transmission grid. For each timestamp we solve the base-case AC power flow, then apply the N-1 criterion by removing each line/transformer/generator in turn and re-solving. An operating point is labeled *secure* if the base case and all contingencies satisfy limits (bus voltages $\in [0.90, 1.10]$ p.u., line loading $\leq 100\%$); otherwise it is *insecure*. Non-convergent power flows are labeled *insecure*.

The test system is a digital twin derived from the topology of the Greek transmission network (35 buses, 46 lines, 135 generators, 110 static generators, 20 loads). AC load flows are computed with the Newton–Raphson method in `pandapower`. N-1 contingencies include all line outages (excluding line index 45) and all generator outages. Table 1 summarizes the dataset.

2.2 Time-aware train/validation/test split

Samples are sorted by timestamp. The AL *training/pool* comes from earlier windows, while the *test* set is the most recent slice and is never used for training or querying. This avoids temporal leakage and mimics deployment where we predict on future data. A small validation split is carved from the training era for early checks.

2.3 Classifier and hyperparameters

Our base model is a Random Forest (RF) because it is fast, robust and provides class-probability posteriors needed by uncertainty-based AL. Across runs we vary hyperparameters in realistic ranges: $n_{\text{estimators}} \in [200, 1500]$, `max_depth` $\in \{18, 20, 24, 25, 28, 30, 35, 40, \text{None}\}$, `min_samples_split` $\in \{2, 4\}$, `min_samples_leaf` $\in \{1, 2, 3\}$, `class_weight` $\in \{\texttt{balanced}, \texttt{balanced\_subsample}\}$. We use seeds $\{42, 1337\}$ for reproducibility.

Classifier dependence. We use Random Forests for probability outputs and fast retraining inside the AL loop. While AL's relative gains often transfer across probabilistic classifiers, we did not perform a systematic model sweep here. Evaluating logistic regression and gradient-boosted trees under the same AL protocol is left to future work.

2.4 Pool-based AL loop

We follow the standard pool-based AL recipe:

(1) Start with an initial labeled set of size i and an unlabeled pool.
(2) Train the RF on the current labeled set; score the pool to obtain class-probability vectors $p(x)$.
(3) Select the next batch of b samples using one of the query strategies below.
(4) Query the simulator for labels of the chosen batch (expensive step); add them to the labeled set.
(5) Repeat for a fixed number of iterations or until the budget is exhausted.

We sweep budgets across runs: $i \in \{100, \ldots, 500\}$, $b \in \{50, \ldots, 200\}$, and up to 40 iterations, which lets us trace long learning curves.

Query strategies. We compare: (i) **Random** (baseline); (ii) **Least-confident** (*uncertainty*): score $1 - \max_c p_c(x)$; (iii) **Margin**: negative gap between top-2 probabilities; (iv) **Entropy**: $-\sum_c p_c(x) \log p_c(x)$. All three uncertainty policies operate on the same RF posteriors and therefore often rank samples similarly.

2.5 Evaluation

After each iteration we evaluate on the fixed test set. At each AL round we retrain the RF from scratch on the enlarged labeled set; new labels are added to training only; the pool remains unlabeled. For each strategy we run multiple configurations and both seeds, then align results by total labeled samples and average across runs to obtain strategy-level learning curves. Unless noted otherwise, TTT values in the main figures are computed on these averaged curves. Appendix A.1 (Table 4a) reports per-run TTT (mean ± std), which is larger due to variability across initial sizes i, batch sizes b, and seeds.

2.6 Metrics

We report Accuracy and ROC AUC on the test set, plus two label-efficiency metrics: **Time-to-Target** (TTT), the smallest number of labeled samples needed for the *average* curve of a strategy to reach a target (e.g., ACC≥0.92 or AUC≥0.98); and **AULC** (Area Under the Learning Curve), computed by trapezoidal integration of metric vs. total labeled. Because simulator seconds per call are roughly constant, relative cost/time savings are well approximated by label savings derived from TTT.

Additional classification metrics. Besides Accuracy and ROC AUC we also track **Precision**, **Recall**, **F1** and the **False Negative Rate (FNR)** on the fixed test set at every AL round. Let TP, FP, FN, TN be counts on the test set. We use the standard definitions: Precision = TP/(TP + FP), Recall = TP/(TP + FN), F1 = $2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$, FNR = FN/(FN + TP) = 1 − Recall. We report *mean ± std across runs/seeds*, and we extract TTT-style thresholds for these metrics when relevant.

3 Results

Figure 1 and Figure 2 show learning curves (averaged across seeds). Across the budget range, all three uncertainty-based policies (entropy, margin, uncertainty) dominate the `random` baseline in both Accuracy and ROC AUC; the area under the learning curve (AULC) is consistently higher.

Table 3 summarizes KPIs used in the paper. At the most important targets, AL reaches the same performance with far fewer labels: at **ACC** ≥ 0.92, AL needs about **500** labels vs. **1 040** for `random` (∼ **52%** fewer); at **AUC** ≥ 0.98, AL needs **580** vs. **960** (∼ **40%** fewer). Final metrics at the maximum budget are also higher for AL (ACC 0.917±0.005 and AUC 0.983±0.002) than for

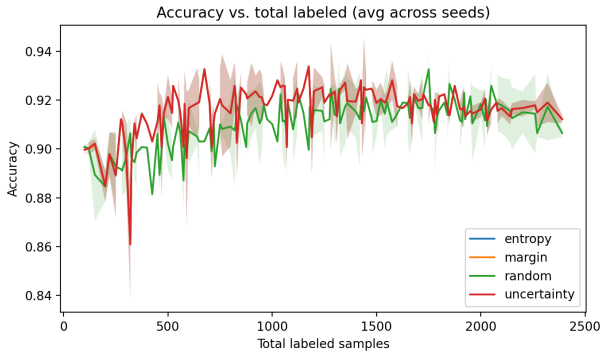


Figure 1: Accuracy vs. total labeled samples (mean ± std across runs). ***(Note: entropy, margin, and uncertainty overlap almost perfectly on this dataset—so the three AL curves/bands lie on top of each other; Random is shown separately for contrast)***

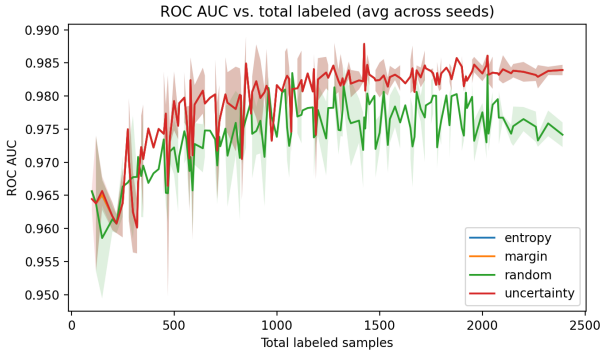


Figure 2: ROC AUC vs. total labeled samples (mean ± std across runs). ***(Note: entropy, margin, and uncertainty overlap almost perfectly on this dataset—so the three AL curves/bands lie on top of each other; Random is shown separately for contrast)***

Table 2: Final test metrics at maximum budget (mean ± std across runs).

Strategy	Accuracy	ROC AUC
entropy	0.917 ± 0.005	0.983 ± 0.002
margin	0.917 ± 0.005	0.983 ± 0.002
uncertainty	0.917 ± 0.006	0.983 ± 0.004
random	0.916 ± 0.010	0.977 ± 0.004

`random` (ACC 0.916±0.010 and AUC 0.977±0.004). Differences at the easier target **ACC** ≥ 0.90 are small (all reach it by ~100–120 labels), which is expected for a low threshold.

On high AUC values. The time-aware split still yields a separable test set for this case study (AUC ≈ 0.98). This likely reflects informative steady-state features and balanced classes, not overfitting to the test era. That said, harder, more imbalanced systems may reduce AUC and amplify AL gains; we treat this as a scope limitation.

Precision, Recall, F1 and FNR.. The additional metrics mirror the ACC/AUC trends: entropy, margin, and uncertainty produce higher AULC and reach target quality with fewer labels than `random`. At targets Precision/Recall/F1 ≥ 0.90 and FNR ≤ 0.10, the uncertainty-based policies consistently hit the thresholds earlier on the average curves, confirming that the AL gains are not specific to a single metric. Shaded bands (std across runs) show the same ordering stability observed for ACC/AUC. Full KPI values and TTT thresholds for P/R/F1/FNR are provided in Appendix A.2 (Table 4b).

Next, we compare label efficiency using Time-to-Target (TTT). Figures 3 and 4 show TTT for accuracy targets 0.90 and 0.92, while Figures 5 and 6 show TTT for AUC targets 0.97 and 0.98. At the **easy** target **ACC** ≥ 0.90 all strategies reach the goal after about **100–120** labels (uncertainty sometimes at 120 due to seed/batch noise). At the more demanding **ACC** ≥ 0.92 target, active-learning policies need about **500** labels, whereas `random` needs **1 040** (i.e., ~**52%** fewer labels). For **AUC** ≥ 0.97, AL reaches the target at **275** labels vs. **325** for `random` (~**15%** fewer), and for **AUC** ≥ 0.98 at **580** vs. **960** (~**40%** fewer). These reductions translate directly into lower simulation time when the average time per labeling call is roughly constant.

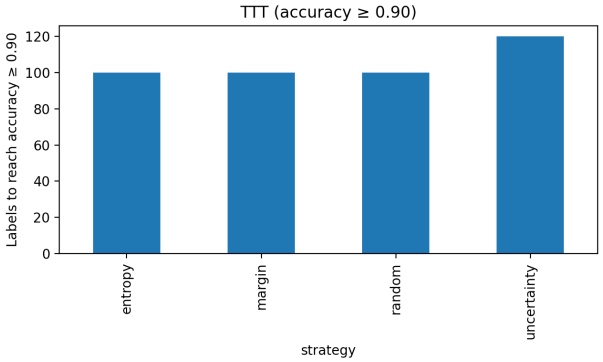


Figure 3: TTT (Accuracy ≥ 0.90): computed on the strategy-level average curve; per-run variability (mean ± std) is reported in Appendix.

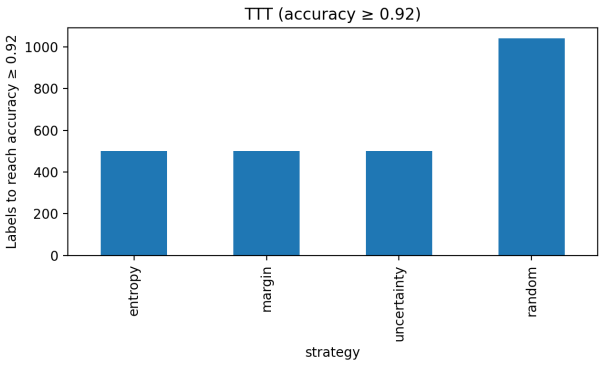


Figure 4: TTT (Accuracy ≥ 0.92): computed on the strategy-level average curve; per-run variability (mean ± std) is reported in Appendix.

Overall, uncertainty-based AL strategies consistently beat `random` at the harder targets (ACC 0.92 and AUC 0.98) while performing similarly at the easier ACC 0.90 threshold; final performance at the maximum budget remains high (ACC 0.917±0.005,

Table 3: KPIs by strategy (averaged across runs). Final metrics reflect per-run means; see Table 2 for mean±std.

	AULC		TTT (labels)				Final	
Strategy	acc	auc	acc ≥ 0.90	acc ≥ 0.92	AUC ≥ 0.97	AUC ≥ 0.98	ACC	AUC
entropy	0.92	0.98	100	500	275	580	0.917	0.983
margin	0.92	0.98	100	500	275	580	0.917	0.983
random	0.91	0.97	100	1 040	325	960	0.916	0.977
uncertainty	0.92	0.98	120	500	275	580	0.917	0.983

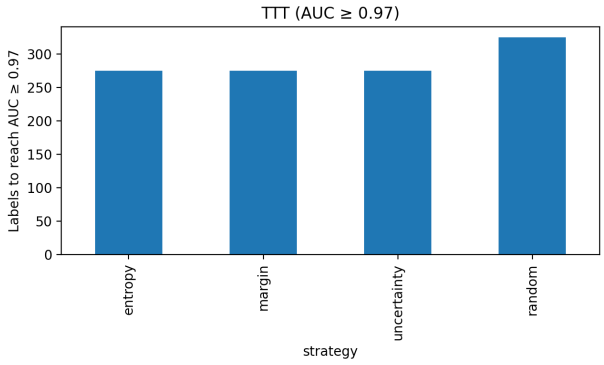


Figure 5: TTT (AUC ≥ 0.97): computed on the strategy-level average curve; per-run variability (mean ± std) is reported in Appendix.

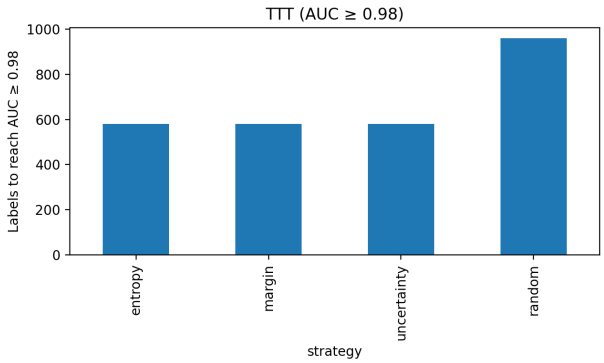


Figure 6: TTT (AUC ≥ 0.98): computed on the strategy-level average curve; per-run variability (mean ± std) is reported in Appendix.

AUC 0.983±0.002 for AL vs. ACC 0.916±0.010, AUC 0.977±0.004 for `random`).

4 Conclusion

This paper demonstrates that AL is a viable strategy for reducing simulation costs in power-grid security assessment. By selectively querying informative contingencies, we cut labels (and thus simulator calls) by **about 52% at ACC** ≥ 0.92 (500 vs. 1 040 with random) and **about 40% at AUC** ≥ 0.98 (580 vs. 960), without sacrificing final performance (AL: ACC 0.917±0.005, AUC 0.983±0.002; random: ACC 0.916±0.010, AUC 0.977±0.004; see Table 2). Fewer simulator calls translate into shorter training times and lower computational and memory requirements, which are particularly important for real-time or resource-constrained applications. Moreover, integrating AL within a digital-twin pipeline enables a feedback loop in which the classifier continuously refines itself using only the most informative contingencies. These findings suggest that exhaustive N-1 simulations are not always necessary for reliable security assessment, paving the way for more scalable and efficient grid-analysis tools.

The present study focuses on a single test system and a Random Forest classifier. In future work we plan to evaluate the proposed framework on larger and more diverse grid topologies (e.g., IEEE 39-bus, 118-bus or national transmission networks) and under varying operating conditions. Another direction is to explore more advanced models such as gradient-boosting machines, deep neural networks or graph neural networks, which may capture complex relationships among grid variables. We also intend to investigate alternative sampling strategies—including diversity-based selection, query-by-committee and Bayesian AL to further improve label efficiency. Finally, extending the methodology to multi-contingency (N-k) and dynamic security assessments (e.g., transient stability) will broaden its applicability in future smart-grid deployments.

Reproducibility

Code, analysis scripts, and a dataset to reproduce all figures and tables will be released at https://github.com/HumAIne-JSI/smart-energy-ea.

Acknowledgements

This work was supported by European Union's funded Project HUMAINE [grant number 101120218]. The authors acknowledge the use of LLMs for language optimization. While the LLMs contributed to enhancing efficiency and refining the presentation of this work, all conceptual frameworks, analyses, and interpretations remain the sole responsibility of the authors.

References

[1] José-María Hidalgo Arteaga, Fiodar Hancharou, Florian Thams, and Spyros Chatzivasileiadis. 2019. Deep learning for power system security assessment. In *2019 IEEE Milan PowerTech*. IEEE, 1–6.

[2] Jiyu Huang, Lin Guan, Yinsheng Su, Haicheng Yao, Mengxuan Guo, and Zhi Zhong. 2021. System-scale-free transient contingency screening scheme based on steady-state information: A pooling-ensemble multi-graph learning approach. *IEEE Transactions on Power Systems* 37, 1 (2021), 294–305.

[3] Kip Morison, Lei Wang, and Prabha Kundur. 2004. Power system security assessment. *IEEE power and energy magazine* 2, 5 (2004), 30–39.

[4] Costas Mylonas, Titos Georgoulakis, and Magda Foti. 2024. Facilitating AI and System Operator Synergy: Active Learning-Enhanced Digital Twin Architecture for Day-Ahead Load Forecasting. In *2024 International Conference on Smart Energy Systems and Technologies (SEST)*. IEEE, 1–6.

[5] Zhongtuo Shi, Wei Yao, Yong Tang, Xiaomeng Ai, Jinyu Wen, and Shijie Cheng. 2023. Intelligent power system stability assessment and dominant instability mode identification using integrated active deep learning. *IEEE Transactions on Neural Networks and Learning Systems* 35, 7 (2023), 9970–9984.

A Additional Results

Table 4: Additional KPI summaries and TTT variability across runs.

A.1 Time-to-Target Variability Across Runs

(a) Per-run Time-to-Target (TTT) mean ± std (labels) by strategy. Note: Values here are per-run TTT (mean ± std). The TTT bars in Figures 3–6 are computed on the averaged curve.

Threshold	Strategy	TTT (mean ± std)
ACC ≥ 0.90	entropy	384 ± 207
	margin	384 ± 207
	uncertainty	372 ± 208
	random	440 ± 225
ACC ≥ 0.92	entropy	751 ± 359
	margin	751 ± 359
	uncertainty	751 ± 359
	random	897 ± 318
AUC ≥ 0.97	entropy	432 ± 178
	margin	432 ± 178
	uncertainty	432 ± 178
	random	502 ± 352
AUC ≥ 0.98	entropy	803 ± 304
	margin	803 ± 304
	uncertainty	803 ± 304
	random	1029 ± 386

A.2 Precision/Recall/F1/FNR KPIs and TTT Thresholds

(b) KPIs by strategy for Precision, Recall, F1, and derived FNR. Time-to-Target (TTT) is the number of labels to reach the threshold (e.g., Precision ≥ 0.90, Recall ≥ 0.90; for FNR, TTT corresponds to FNR ≤ 0.10).

Strategy	AULC P	TTT P≥0.90	Final P	AULC R	TTT R≥0.90	Final R	AULC F1	TTT F1≥0.90	Final F1	TTT FNR≤0.10	Final FNR
entropy	0.948	500	0.952	0.916	500	0.922	0.931	500	0.936	500	0.078
margin	0.948	500	0.952	0.916	500	0.922	0.931	500	0.936	500	0.078
random	0.928	960	0.940	0.905	1040	0.906	0.916	1040	0.922	1040	0.094
uncertainty	0.948	500	0.952	0.916	500	0.922	0.931	500	0.936	500	0.078

Supporting Material Reuse in Drone Production

Rok Cek
rok.cek@gmail.com
Jožef Stefan Institute
Ljubljana, Slovenia

Oleksandra Topal
oleksandra.topal@ijs.si
Jožef Stefan Institute
Ljubljana, Slovenia

Linda Leonardi
linda.leonardi@cetma.it
CETMA
Brindisi, Italy

Margherita Forcolin
margherita.forcolin@maggioli.gr
Maggioli Group
Santarcangelo di Romagna, Italy

Klemen Kenda
klemen.kenda@ijs.si
Jožef Stefan Institute
Ljubljana, Slovenia

Abstract

This paper, part of the European Horizon project Plooto, details an end-to-end, data-driven framework for reusing expired carbon-fiber prepregs in drone production. First, 19 batches of expired prepregs were tested, revealing that most remained usable within the first year after expiration. Machine learning models were then developed to predict material usability pre-production and product quality post-production, using manufacturing data and time-series features. To facilitate this process, a dedicated data pipeline and an interactive Product Quality Explorer tool were created to support explainable model development and integration with industrial partners. This framework demonstrates how combining material requalification with data-driven predictions can lower costs and support circularity in drone production.

Keywords

circular economy, digital product passport, machine learning, product quality

1 Introduction

The growing demand for lightweight, high-performance materials is driving the increased use of carbon fiber reinforced polymers (CFRPs) in industries such as aerospace, automotive, and drones. However, this rapid adoption also creates challenges, particularly with the accumulation of expired materials. While much research has focused on recycling fully cured CFRPs, less attention has been given to the reuse of uncured prepregs, which, despite expiring during storage, can still retain valuable properties [5]. Addressing this challenge is crucial for advancing circular economy principles in high-tech manufacturing.

This paper presents research from the European Horizon project Plooto, focusing on the reuse of expired prepregs in sustainable drone production. Our work contributes in three key areas: (1) a comprehensive evaluation of the effects of aging on expired prepregs through thermal, chemical, and mechanical testing to establish requalification thresholds [1], (2) the development of machine learning models to predict the usability of expired prepregs before production, and (3) the application of predictive models to assess the quality of final products after production, specifically for sandwich panels made from recycled prepregs. By combining experimental testing with data-driven methods, our findings highlight the potential to reduce waste and enhance sustainability in drone manufacturing.

By integrating machine learning models to predict the usability of expired prepregs and assessing the quality of final products, we provide industrial partners with actionable insights that directly enhance operational decision-making. The combination of material requalification and predictive analysis supports the sustainability goals of the drone production process.

2 Data and Methods

2.1 Materials and experimental techniques used for prepreg usability assessment

Expired rolls of epoxy prepregs from HP Composites S.p.A were used for this study. A total of 19 prepreg batches were investigated, comprising four different resin systems (ER450, IMP509, X1, ER431), with reinforcement varying according to supplier availability. Usability is assessed through periodic chemical-physical and mechanical testing after the expiration date, to monitor property changes in materials stored at -18°C. Differential Scanning Calorimetry (DSC) tests were performed with Mettler Toledo DSC 823e on uncured prepreg samples by applying a dynamic heating from -40°C to 250°C at 20°C/min under a nitrogen atmosphere. DSC analysis provides two key parameters: the glass transition temperature of the uncured system (T_{g0}), related to the initial crosslink density, and the residual cure degree (α), calculated from the polymerization enthalpies values. Composite plates for physical and mechanical testing were manufactured by draping a variable number of prepreg plies at 0°, depending on reinforcement type, to obtain cured laminates of ≈ 3 mm. The prepreg plies were stacked on a flat mold surface over a peel ply. The plates were then covered with an additional peel ply, a release film, and a breather layer. The self-adhesive seal and the vacuum bag were used to create a sealed vacuum during the entire process. Plates curing was carried out in a hot press according to the curing cycle recommended by the supplier in the material datasheet, as reported in the table 1. The void content (V_c) was measured on five specimens through a digestion procedure according to standard ASTM D3171 Method A. [3] The interlaminar shear strength (ILSS) tests were performed with a 3-point bending system on MTS Insight machine according to the standard test ASTM D2344 [2] on five different specimens for each prepreg batch. These experimental results, including DSC data, ILSS, and void content (V_c) measurements, provide essential features for the machine learning models discussed in Section 2.2. The values of key properties such as the glass transition temperature (T_{g0}), residual cure degree (α), and interlaminar shear strength (ILSS) are directly used to predict the usability of the

Information Society 2025, Ljubljana, Slovenia

https://doi.org/https://doi.org/10.70314/is.2025.sikdd.20

expired prepregs and to assess the quality of the final products after manufacturing.

Material	Temperature (°C)	Time (h)	Pressure (bar)
ER 450	135°C	2h	6 bar
IMP 509	140°C	1.5h	4 bar
X1 120	130°C	1.5h	6 bar
ER 431	125°C	1h	5 bar

Table 1: Curing cycle parameters for the plates recommended in the material datasheet.

2.2 Predicting the usability and key parameters of prepreg using machine learning methods

The results from the DSC tests, along with other experimental data such as ILSS and void content (V_c) collected in Section 2.1, were systematically organized and used as input features for the machine learning models to predict prepreg usability and key process parameters. Each row represents one checkpoint on an expired roll and includes: test date, month code, prepreg code and lot, type (expired roll), stocking temperature (−18 °C), original expiry date, α (%), $T_{g,\text{onset}}$ (°C), ILSS (MPa), V_c (°C; curing temperature), Usable (Y/N), and, when redefinition is applied, pressure (bar), temperature (°C), time (min), and the redefined expiry date. For the correct operation of machine-learning methods, a *days-after-expiry* feature was introduced and computed as `test_date` − `original_expiry_date`.

The study addresses two predictive tasks: a classification problem for *Usable* (three classes: Y, Y/N, N) and regression problems for process/quality parameters (ILSS, $T_{g,\text{onset}}$, V_c, α). Analysis proceeds in two stages. First, a per-material stage fits separate models for each prepreg system (ER450, IMP509, ER431, X1) to resolve material-specific issues observed during preliminary inspection. Second, a pooled stage trains a unified model over all records to evaluate cross-material generalisation.

Predictors are restricted to pre-test covariates: days-after-expiry, material identity, normalised lot descriptors, month code, storage conditions, and other metadata available at decision time, while measured targets are excluded from inputs to prevent label leakage. Random-forest classifiers and regressors (scikit-learn) parameterised as $n_{\text{estimators}}$=100, max $\_depth$=3, random_state=42 serve as the base models and enable inspection of feature importances.

Performance estimation relies on leave-one-out cross-validation (LOO-CV) [6] in both stages. For the classification task, overall accuracy is reported to evaluate the model's performance in predicting prepreg usability. For the regression tasks, R^2, MAE, and RMSE are used to assess the model's ability to predict continuous process parameters. R^2 measures the proportion of variance explained, while MAE provides the average error magnitude, and RMSE emphasizes larger errors. Feature-importance profiles are examined to identify the dominant drivers of re-usability and variation in process parameters across materials and in the pooled setting.

2.3 Machine Learning for Post-Production Quality Prediction

This part of the pilot addressed the prediction of production quality in sandwich panel manufacturing, with the aim of supporting drone production after re-qualification.

The dataset combined two types of information. The first component consisted of production metadata, which described the context of each cycle. These attributes included the date of the cycle, the operator responsible for production, the specific prepreg batch (identified by lot number), and the number of days between when the prepreg was made and used in production. Tool-related information was also provided, such as which tool was used and how many cycles had passed since its last maintenance. Each cycle was associated with a measurement curve identifier, a quality result (labelled as either fully compliant, minor defect, or scrap), and, in cases of non-compliance, the reported reason for failure.

The second component of the dataset consisted of time-series data collected during the manufacturing process. For each cycle, approximately 1,300 measurements were recorded at ten-second intervals. These measurements included the chamber's target temperature (setpoint), the actual chamber temperature, the temperature of the piece being moulded, and the vacuum setpoint. Together, these readings captured the thermal and pressure dynamics that govern the curing of composite materials.

To make this information usable for machine learning models, feature extraction was required. Temperature curves were divided into intervals based on their inflection points—that is, the points where the curve transitioned from stable plateaus to rising or falling slopes. Each interval was then summarised using statistical properties such as average, minimum, maximum, variance, and trend. In addition to these aggregated features, new variables were engineered to capture deviations from expected behaviour. For example, the vacuum difference quantified the gap between the measured and target pressure, while the temperature difference measured the offset between chamber setpoints and the actual values recorded. These derived variables provided indicators of process deviations that might affect the final product quality.

The analysis followed the CRISP-DM methodology, beginning with data fusion and preparation, followed by feature selection and model training. Metadata and time-series features were combined into a single dataset, from which irrelevant or redundant variables were removed.

For predictive modelling, several classification algorithms were evaluated to balance interpretability and performance. Logistic regression and decision trees offered transparent decision boundaries, while ensemble methods such as random forests and gradient boosting provided stronger predictive power by aggregating multiple weak learners. Multi-layer perceptrons (MLP) were also considered to capture non-linear patterns in the data.

To integrate the methodology into the production workflow, a dedicated service was implemented. Metadata was provided in an Excel (.xlsx) file, while the process data was provided in .rdb formats by the industrial partner. A pipeline was developed to automatically download these files from a shared Dropbox folder provided by the industrial partner, parse the .rdb data, and convert the files into structured JSON files. The JSON files were enriched with derived variables and unique identifiers, then uploaded to the Plooto platform via its API. This ensured seamless integration of raw production data with machine learning models, enabling continuous prediction of product quality.

As part of this work, we developed a tool called Product Quality Explorer to support domain experts in analyzing production data and assessing product quality [4]. Its primary goal is to facilitate the creation of explainable machine learning models. The tool helps users understand factors influencing quality outcomes and make informed adjustments to the manufacturing

process. The tool provides a summary of descriptive statistics (count, mean, standard deviation, minimum, quartiles, and maximum) and allows users to visualize selected columns through histograms and boxplots. Finally, it generates a heatmap of all columns to provide an overview of relationships within the data.

In the next step, the user selects the features to include in the machine learning model. This step is necessary both to define the target variable for prediction and to exclude irrelevant columns such as IDs, dates, or textual data. The tool also provides several options for handling missing values. The user can choose the approach that best suits the dataset: leaving missing values unchanged (which may prevent some algorithms from functioning properly), removing features with missing values, removing rows containing missing values, or imputing missing values using the column mean.

The next step provides the option to generate new attributes. This can be done through techniques such as one-hot encoding, polynomial feature generation, or logarithmic transformations. After creating new attributes, the user selects the features to be used in the machine learning process. This selection can be performed manually or automatically with the assistance of genetic algorithms.

Finally, the user can select which machine learning models to apply. Once training is complete, the results are presented in a summary table containing performance metrics such as precision, recall, F1-score, and accuracy, along with a confusion matrix visualization. The tool also provides a comparative overview of model performance across all metrics (precision, recall, F1-score, accuracy).

In addition to evaluation, the system integrates explainability techniques. Global explanations are generated using SHAP to show how features influence model decisions across the entire dataset, while local explanations are provided using SHAP and LIME to illustrate how the model arrived at a prediction for a specific datapoint. These explanations are supported by interactive visualizations, which enable users to better understand both the overall model behavior and individual predictions.

3 Results

3.1 Results of usability assessment

Ageing trends from DSC. Differential scanning calorimetry (DSC) on the selected prepreg rolls (grouped by resin system) shows that T_{g0} increases progressively over time after expiration. This behaviour is consistent with *i)* increasing molecular weight and *ii)* higher crosslink density of the polymer network due to ongoing polymerization. The measured α values align with the T_{g0} trend, indicating a time-dependent decrease in the residual degree of cure; notably, within the first two years after expiration, the reduction remains limited to <15%.

Mechanical strength and porosity evolution. Across all batches, interlaminar shear strength (ILSS) exhibits a time-dependent decline: reductions generally do not exceed 15% within the first 12 months after expiration, whereas more pronounced decreases of 25–30% occur in the 12–24 month interval. Consistent with this mechanical trend, the void content V_c remains below 10% during the first 12 months after expiration and increases thereafter, often exceeding 15% in later months.

3.2 Predictive modeling results for prepreg reuse

We analysed $N = 81$ inspection records with a two–stage workflow: global model across all prepregs and material-specific models were trained and estimated using leave-one-out cross-validation (LOO-CV). Table 2 summarizes the results of all experiments, including classification and regression performance for global and material-specific models.

Type	Usability	Metrics	α	T_{g0}	ILSS	V_c
All types	Acc=0.91	**AggR²** =	0.83	0.77	0.7	0.77
		MAE =	1.22	1.05	4.49	1.52
		RMSE =	1.59	1.33	5.93	1.98
ER450	Acc=0.96	**AggR²** =	0.86	0.88	0.92	0.94
		MAE =	1.25	0.54	2.75	0.87
		RMSE =	1.51	0.77	4.05	1.15
IMP509	Acc=0.87	**AggR²** =	0.76	0.6	0.82	0.8
		MAE =	1.44	1.23	2.5	1.35
		RMSE =	1.9	1.58	3.01	1.75
X1	Acc=0.96	**AggR²** =	0.82	0.79	0.79	0.43
		MAE =	1.12	0.98	2.41	1.77
		RMSE =	1.44	1.12	3.09	2.32
ER431	Acc=1	**AggR²** =	0.97	0.88	0.94	0.87
		MAE =	0.57	0.89	1.43	1.06
		RMSE =	0.76	1.15	1.93	1.64

Table 2: LOO-CV performance across prepregs for regression and classification

As we can see from the presented results, the global multiclass classifier achieved 0.91 accuracy under LOO-CV on an imbalanced set (54 Y / 14 Y-N / 13 N), indicating that a simple preproduction screen is feasible from routine metadata. Per-material classifiers were even higher (often ≥0.96), but these figures are almost certainly optimistic given tiny per-material sample sizes and class imbalance. A detailed classification report, including precision, recall, and F1 scores, can be provided upon request.

A consistent trend across the regression tasks is the superior performance of models trained on a single prepreg type compared to the global model trained on all data.[1] For instance, the global model predicted ILSS with an aggregate R^2 of 0.70, whereas the material-specific models for ER450 and ER431 achieved much higher scores of 0.92 and 0.94, respectively. This suggests that ageing and curing behaviours are highly specific to the resin system, and tailored models better capture these characteristics. However, this is not a universal rule; the prediction of V_c for the X1 prepreg (aggregate R^2=0.43) was notably worse than the global model (aggregate R^2=0.77), indicating that in cases of very limited data or less distinct features, the global model can be more robust.

Feature importance analysis performed during the experiments revealed the most influential factors in predicting key parameters in Table 2. The Days_Since_Expiry was consistently one of the most critical predictors across both global and material-specific models, confirming its fundamental role in tracking material degradation. Furthermore, the analysis revealed strong intercorrelations between the measured properties themselves. For example, the degree of cure (α) and T_{g0} were often the most

[1]The dataset is modest and unevenly distributed across resins (ER450 n=28, X1 n=22, IMP509 n=15, ER431 n=14). Consequently, per-material models are trained on few observations and LOO-CV performance is likely optimistic.

important features for predicting ILSS and V_c, indicating that these thermal and chemical properties are highly interdependent. Batch identifiers (prepreg code/lot) were generally minor, although *lot* occasionally ranked higher for ILSS, indicating possible batch effects.

Taken together, these patterns suggest that compact, physics-aligned feature sets explain most of the variance, and that ageing/α consistently drive both regression and classification. Nevertheless, limited data—especially for IMP509 and ER431—and the optimism of LOO-CV preclude production use without further data collection and validation across broader process conditions.

3.3 Evaluation of Post-Production Classification Models

The predictive modelling was applied to production cycles from sandwich panel manufacturing provided by the Italian pilot partners. We also used the aforementioned Product Quality Explorer tool after we had already transformed the data and created new features. The objective was to assess whether production quality outcomes could be predicted from a combination of metadata and process-derived time-series features. This is particularly important for supporting drone production after re-qualification, as early detection of potential quality issues can prevent defective panels from progressing further in the manufacturing chain. Moreover, it can save manufacturers time, energy, and personnel costs, as each panel must currently be manually inspected and tested.

The dataset comprised 294 production cycles, the majority of which were compliant, with only a small fraction classified as non-compliant. This strong imbalance reflects real-world conditions, where defects are rare but critical, yet it also creates difficulties for machine learning approaches. Most algorithms tend to favour the majority class, which can lead to high overall accuracy but poor detection of defective cases.

Several classification algorithms were tested. Overall accuracy values appeared relatively high (between 0.77 and 0.85) this was largely driven by the correct classification of compliant cases. Performance on the minority (non-compliant) class was weaker, as reflected by modest recall and F1-scores. This indicates that while the models are well-suited to reproducing the majority outcome, their ability to identify rare defective panels is more limited.

These findings suggest that machine learning can provide useful insights into production quality trends, but further progress requires additional data, particularly more defective cases. A larger dataset would allow models to better distinguish between compliant and non-compliant cycles, thereby increasing their value as a decision-support tool in quality assurance.

The detailed performance of each tested classifier is reported in Table 3.

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.846	0.838	0.838	0.838
Decision Tree	0.769	0.764	0.738	0.745
Random Forest	0.808	0.797	0.806	0.800
XGBoost	0.808	0.797	0.806	0.800
LightGBM	0.846	0.838	0.838	0.838
Support Vector Machine (SVM)	0.808	0.801	0.788	0.793
Multi-layer Perceptron (MLP)	0.808	0.801	0.788	0.793

Table 3: Performance of machine learning models on the Italian pilot sandwich panel dataset.

4 Conclusion

This study demonstrates an end-to-end approach that integrates material science and machine learning to enhance the reuse of expired prepregs in drone production. By evaluating and requalifying expired materials, we have shown that they remain serviceable within the first year after expiry, with gradual performance decline, particularly in interlaminar shear strength and curing behavior. This underscores the effectiveness of resin-specific reuse gates and modified processing windows to extend material lifetimes.

Machine learning models were employed to support both pre-production and post-production processes. The pre-production models classified expired prepregs for reuse, while the post-production models predicted the quality of sandwich panels based on combined metadata and process features. Despite challenges related to data imbalance, the results demonstrate the potential for predictive quality monitoring in manufacturing, contributing to more sustainable production practices.

The integration of machine learning with material science not only optimizes requalification processes and reduces waste, but also supports cost reduction and environmental sustainability in high-performance manufacturing. Future work should focus on expanding datasets, refining resin-specific criteria, and exploring the broader applicability of the models in other composite manufacturing contexts, further advancing circular economy principles.

Acknowledgements

This work was supported by the European Commission under the Horizon Europe project Plooto, Grant Agreement No. 101092008. We would like to express our gratitude to all project partners for their contributions and collaboration.

The authors acknowledge the use of LLMs for language optimization. While the LLMs contributed to enhancing efficiency and refining the presentation of this work, all conceptual frameworks, analyses, and interpretations remain the sole responsibility of the authors.

References

[1] Constance Amare, Olivier Mantaux, Arnaud Gillet, Matthieu Pedros, and Eric Lacoste. 2022. Innovative test methodology for shelf life extension of carbon fibre prepregs. *IOP Conference Series: Materials Science and Engineering*, 1226, 1, (Feb. 2022), 012101. https://dx.doi.org/10.1088/1757-899X/1226/1/012101.

[2] ASTM International. 2022. ASTM D2344/D2344M-22: Standard Test Method for Short-Beam Strength of Polymer Matrix Composite Materials and Their Laminates. West Conshohocken, PA, USA, (2022). Retrieved Sept. 3, 2025 from https://store.astm.org/d2344_d2344m-22.html.

[3] ASTM International. 2022. ASTM D3171-22: Standard Test Methods for Constituent Content of Composite Materials. West Conshohocken, PA, USA, (2022). Retrieved Sept. 3, 2025 from https://store.astm.org/d3171-22.html.

[4] Rok Cek and Klemen Kenda. 2025. Product quality explorer - determining product quality based on the digital product passport. In *17th Jožef Stefan International Postgraduate School Students' Conference : 28th–30th May: Book of abstracts: from research to reality*, 33. http://ipssc.mps.si/auxiliary_material/IPSSC25%20BoA.pdf.

[5] Gaurav Nilakantan and Steven Nutt. 2015. Reuse and upcycling of aerospace prepreg scrap and waste. *Reinforced Plastics*, 59, 1, 44–51.

[6] Tzu-Tsung Wong. 2015. Performance evaluation of classification algorithms by k-fold and leave-one-out cross validation. *Pattern Recognition*, 48, 9, 2839–2846.

Temporal Dynamics and Causal Feature Integration for Predictive Maintenance in Manufacturing Systems: A Causality-Informed Framework

Seyed Iman Hosseini
iman.hosseini@ijs.si
Jožef Stefan Institute
Ljubljana, Slovenia
Jožef Stefan International Postgraduate School
Ljubljana, Slovenia

Klemen Kenda
klemen.kenda@ijs.si
Jožef Stefan Institute
Ljubljana, Slovenia
Qlector
Ljubljana, Slovenia

Dunja Mladenič
dunja.mladenic@ijs.si
Jožef Stefan Institute
Ljubljana, Slovenia
Jožef Stefan International Postgraduate School
Ljubljana, Slovenia

ABSTRACT

Predictive maintenance is increasingly central to manufacturing, where the goals are to reduce unplanned downtime and extend asset lifetimes. Conventional models often rely on correlations that insufficiently capture temporal dynamics and causal dependencies underlying failures. This study proposes a causality-informed feature-engineering pipeline that combines cross-correlation-derived lags with VARLiNGAM to construct lag-aware features from multivariate sensor streams, and evaluates it against standard time-series models using a time-aware split. Three machine-learning models—Random Forest, XGBoost, and Gradient Boosting—were trained and assessed by F1-score (rather than accuracy) on a single-machine subset of the Microsoft Azure Predictive Maintenance dataset (8,708 samples; 26 failures, ≈0.3% prevalence). XGBoost trained on raw temporal features achieved F1 $\approx$ 0.94 for longer prediction horizons ($\geq$10 h) under time-series–aware cross-validation, with performance declining at shorter horizons as temporal context diminishes. In this setting, causality-informed features did not improve results over the raw-feature baseline. These findings indicate that, with data from a single machine, causal discovery is susceptible to overfitting and may suppress informative temporal patterns; broader, multi-machine datasets are likely required for causality-enhanced representations to yield consistent gains.

KEYWORDS

Predictive Maintenance, Causality, Time-Series Analysis, Machine Learning, VARLiNGAM, Manufacturing Systems

1 INTRODUCTION

The rising complexity and interconnectivity of industrial systems have accelerated the need for intelligent maintenance strategies that move beyond reactive and preventive paradigms. Predictive maintenance, driven by sensor data and machine learning, has emerged as a transformative approach to minimize unplanned downtime and optimize asset life cycles [1]. Traditional predictive maintenance models, however, often rely on statistical correlations that fail to capture the directionality and temporal dynamics inherent in real-world system failures [6].

To address these limitations, this study proposes a causality-informed framework for predictive maintenance that leverages temporal causal discovery techniques, such as Vector Autoregressive LiNGAM (VARLiNGAM), to engineer predictive features from multivariate sensor data. Our approach integrates cross-correlation analysis and lag-optimized causal graphs to detect failure precursors and identify their optimal predictive windows.

We hypothesize that the observed lack of competitive advantage for causality-informed models, especially when applied to data from a single machine, arises from the limited operational diversity and failure variability. This limitation may cause models to overfit to machine-specific correlations and exclude informative temporal features, thereby hindering their generalizability. Testing this hypothesis through multi-machine datasets will be a key focus of future work.

2 RELATED WORK

Causality in time series analysis has become increasingly critical in predictive maintenance, particularly within industrial and manufacturing domains, where early failure detection plays a pivotal role in minimizing operational disruptions and financial losses [5]. Classical statistical models have been widely used to infer causal relationships between sensor measurements and machine states, yet they often fail to capture complex temporal dynamics and the nonlinear relationships inherent in real-world system failures.

Recent studies have explored advanced causal inference techniques to enhance fault prediction. Wang S. *et al.* proposed a framework for fault diagnosis that integrates spatiotemporal dependencies, demonstrating improved predictive accuracy in chemical manufacturing systems [9]. While their work advances reliability in industrial diagnostics, it lacks the flexibility to generalize across diverse application domains. On the other hand, Cui et al. introduced a deep learning framework that enhances predictive maintenance by integrating causal reasoning and long-sequence multivariate time-series data, significantly improving predictive performance and interpretability [3]. Despite this, the challenge of automating temporal feature engineering and seamlessly deploying models across different domains remains.

Yang X. *et al.* contributed to the growing literature on data-driven causal analysis by incorporating dynamic latent variables and probabilistic graphical models into causal modeling frameworks [10]. However, these models have yet to fully address the temporal feature extraction required for scalable deployment in real-world predictive maintenance applications. Furthermore, more recent work by Wang Q. *et al.* introduced a Causal Graph

Information Society, 2025, Ljubljana, Slovenia

https://doi.org/https://doi.org/10.70314/is.2025.sikdd.12

Convolution Module that adapts causal discovery within time-series prediction [8], but their approach is still dependent on complex model adjustments across domains.

In this study, we propose a novel framework that integrates lagged correlation with causal analysis techniques to detect failure precursors and quantify their lead times. This framework automates temporal feature engineering and is designed for diverse real-world applications across manufacturing settings, without requiring extensive architectural modifications. The automation of temporal feature engineering and its seamless deployment across comparable manufacturing environments remains a significant challenge, and extending generalization beyond this domain is left for future work.

3 EXPERIMENT

Our experimental methodology followed a sequential four-stage process to construct and validate a robust failure prediction model, as shown in Figure 1. The first stage involved performing a cross-correlation analysis between each sensor's time-series data and the target failure events to determine the optimal predictive time lag, which guided the subsequent steps. In the second stage, the identified optimal lag was used to parameterize a Vector Autoregressive LiNGAM (VARLiNGAM) model, which generated a directed acyclic graph (DAG) representing the causal relationships and effect strengths between sensor variables and the failure event. The third stage focused on creating a causality-informed feature vector by integrating standard statistical metrics from rolling time windows along with advanced features informed by the causal analysis, using the correlation strengths and causal effect strengths derived from the VARLiNGAM model to select and weight features based on their respective optimal and causal lags. Finally, in the fourth stage, the enriched feature set was fed into a machine learning pipeline, employing a time-based data split to prevent look-ahead bias, and training several classification models, including Random Forest, XGBoost, and Gradient Boosting, to assess the effectiveness of the causality-informed approach for predictive maintenance. This integrated approach enhances the predictive capabilities of machine learning models, offering a robust solution for failure prediction in industrial settings.

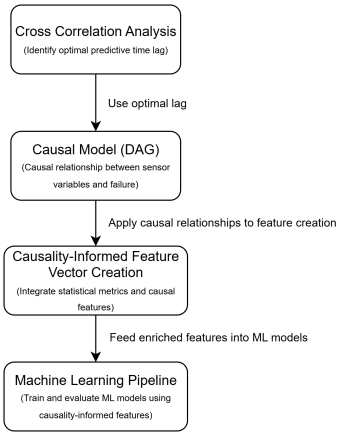


Figure 1: proposed framework

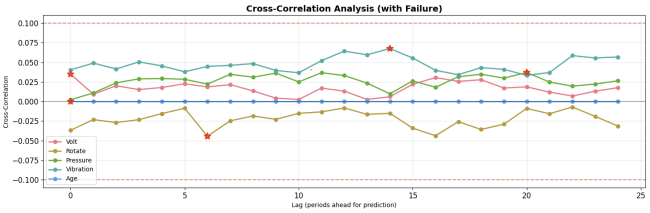


Figure 2: Cross correlation analysis

3.1 Dataset and Preprocessing

We used the Microsoft Azure Predictive Maintenance Dataset [2], which provides hourly telemetry (voltage, rotation, pressure, vibration) plus maintenance records, failure events, incident reports, and machine metadata for 100 machines over 12 months in 2015 (over 800k hourly summaries and thousands of non-failure error entries). For this study, we restricted the analysis to machine ID 98; after cleaning and merging the sources, we constructed a causality-informed feature vector and standardized features across modalities. Cross-correlation suggested predictive lags of 1–24 hours, so we derived lagged/statistical features from six primary variables (voltage, rotation, pressure, vibration, age, error type). The final dataset comprised 8,708 samples with 26 failures (≈0.3%), indicating strong class imbalance [7, 2]. The feature set comprised 150 causality-informed features and 36 features without causal information.

3.2 Cross-correlation Analysis

Cross-correlation analysis examines the correlation between two time series as a function of the time lag applied to one of them [11][12]. Unlike simple correlation, which measures linear relationships at a single point in time, cross-correlation reveals how variables relate across different time delays, making it particularly valuable for identifying lead-lag relationships and temporal dependencies. The initial phase of our experimental framework involved a cross-correlation analysis to empirically determine the predictive temporal relationships between sensor signals and equipment failures. For each sensor, we computed the Pearson correlation coefficient between its time series and the binary failure time series across a range of discrete time lags. This procedure was executed by systematically shifting the failure signal backward in time, which allowed for the correlation of sensor readings at a given time t with failure events at a future time t + lag. The optimal predictive lag for each sensor was then identified as the time lag that yielded the maximum absolute correlation value. This analysis is critical as it quantifies the time window in which each sensor's data is most informative for forecasting an impending failure, thereby providing an empirical foundation for the subsequent causal discovery and feature engineering stages.

In the cross-correlation plot shown in Figure 2, the red star annotated on each sensor's curve denotes the optimal predictive lag—20 hours for Pressure, 14 hours for Vibration, and so forth. This marker identifies the specific time lag, measured in hours, at which the sensor's signal exhibits the highest absolute Pearson correlation with the future failure event. Consequently, the red star highlights the most influential temporal offset for each variable, effectively quantifying the sensor's most informative predictive window within the 24-hour forecasting horizon.

3.3 Causal Graph Construction

To elucidate the causal interdependencies between sensor signals and equipment failures, a causal graph was constructed using VARLiNGAM. This methodology first employs a Vector Autoregression (VAR) model to capture the linear, time-lagged relationships among the multivariate sensor time series. The optimal lag for the VAR model was adaptively informed by the preceding cross-correlation analysis to focus on the most predictive temporal window. Following the VAR estimation, the LiNGAM algorithm is applied to the resulting model residuals, or innovations. By exploiting the non-Gaussian nature of these innovations, LiNGAM uniquely identifies the contemporaneous causal structure—the instantaneous effects between variables—and determines the direction of influence, thereby producing a directed acyclic graph (DAG). The final output is a set of adjacency matrices representing the causal graph, where each non-zero entry quantifies the strength and direction of a causal link from one variable to another at a specific time lag. Our approach constructs a directed causal graph from time-series sensor data using the following steps:

(1) **Data Sorting and Integrity:** Chronologically sort sensor data, verifying integrity and noting irregular intervals.
(2) **Variable Definition:** Define variables which are vibration, rotation, pressure, voltage, and a binary failure indicator as the target node.
(3) **Causal Model Setup:** Configure a VARLiNGAM [4] model with a specified lag order and BIC-based pruning.
(4) **Model Fitting:** Fit the model to the prepared data matrix, applying regularization—by adding small Gaussian noise (e.g., 10^{-6})—when numerical instability arises during VARLiNGAM causal graph construction due to ill-conditioned matrices.
(5) **Adjacency Extraction:** Extract adjacency matrices to identify directed edges, effect strengths, and corresponding lags.
(6) **Graph Assembly:** Assemble the causal graph, categorizing edges by their relation to the target and between sensor variables.

This workflow ensures that temporal ordering is respected and that detected causal links most likely represent meaningful relationships for predictive maintenance and further analytical investigations. Figure 3 presents the causal graph generated by the VARLiNGAM algorithm, illustrating the network of causal relationships between sensor telemetry (volt, pressure, vibration, rotate), machine properties (age), and the target failure event. In this graph, nodes represent the variables, and the directed edges (arrows) signify the direction of causality, with edge thickness corresponding to the strength of the effect. The labels on each edge quantify the causal strength and the time delay (lag) in hours. The analysis reveals a complex web of interactions, prominently highlighting that machine age is the most significant causal driver of failure, with an exceptionally strong effect strength at a lag of 6 hours. Other notable, though weaker, causal pathways are also identified, such as the influence of rotate on failure. This causal structure provides critical insights into the system's dynamics, identifying the key variables and time-delayed interactions that precede a failure event.

3.4 Causality-Informed Feature Engineering

We prepared the data by building a *causality-informed feature vector* grounded in the paper's causal graph and a temporal causality

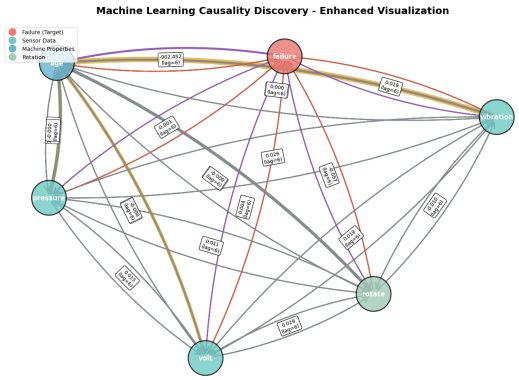


Figure 3: Causal Graph

analysis that selects per-sensor optimal prediction windows. Using a sliding feature window (typically 72 h), samples are formed from historical data only to avoid leakage. Feature construction proceeds in four stages: (1) **basic statistics** (mean, standard deviation, min/max, latest/earliest within the window); (2) **causality-aligned temporal features** computed at the optimal lags identified by causal analysis; (3) **dynamics** via trend slopes (linear regression), rolling volatility (standard deviation), and rates of change; and (4) **cross-feature terms** implied by the causal graph (e.g., voltage/rotation ratios and pressure–vibration correlations). Targets are defined for multiple horizons (1, 6, 12, and 24 h ahead) to enable early warnings at different lead times. The resulting dataset contains 150 features that integrate causal dependencies with temporal patterns.

3.5 Machine Learning Models

Three classification algorithms, each configured with default hyperparameters, were evaluated using time-based data partitioning to mitigate the risk of data leakage.

- **Random Forest (RF)**: Ensemble method with 200 estimators, maximum depth of 15, and balanced class weights
- **XGBoost (XGB)**: Gradient boosting with 200 estimators, learning rate of 0.1, and automatic scale balancing
- **Gradient Boosting (GB)**: Scikit-learn implementation with 200 estimators and 0.8 subsample ratio

Model performance was assessed using F1 Score metric appropriate for imbalanced classification:

- F1-Score: Harmonic mean of precision and recall

A time-series–aware data partitioning strategy was implemented using scikit-learn's *TimeSeriesSplit*, which generates folds in chronological order by progressively expanding the training set with earlier observations and reserving subsequent periods for testing. This procedure ensures that all training data temporally precedes the corresponding test data. To approximate stratification and preserve class balance between rare failure and more frequent non-failure events, the folds were constructed to proportionally distribute failure cases across splits without introducing randomization. This design maintains the temporal integrity of the sensor data while supporting reliable model evaluation.

4 RESULTS AND DISCUSSION

Figure 4 presents the comprehensive F1-score evaluation of all three models, while Figure 5 provides a comparative analysis

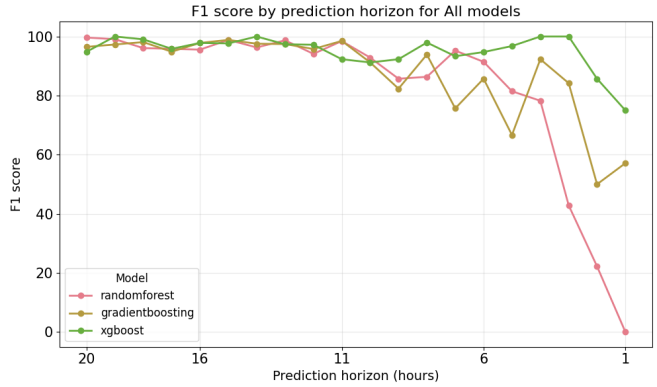


Figure 4: F1-score evaluated over a 20-hour prediction horizon

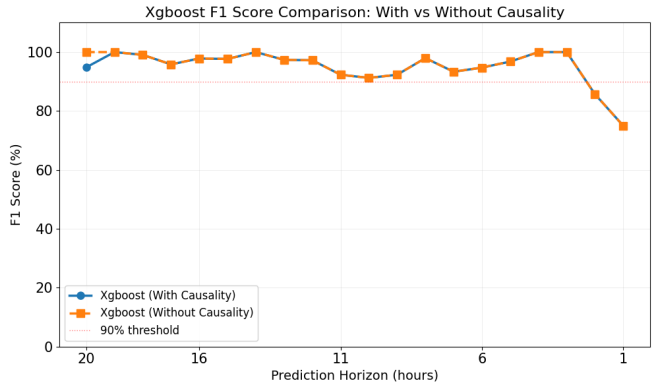


Figure 5: The XGBoost F1-score across a 20-hour prediction horizon, evaluated with and without a causality-informed feature vector

of the XGBoost model with and without the causality-informed feature vector. Standard time-series models, particularly those trained on raw temporal data, consistently outperform causality-informed approaches in predictive maintenance tasks, especially at extended prediction horizons. XGBoost, for instance, achieves F1 scores exceeding 94% for horizons beyond 10 hours, though performance declines with shorter windows due to reduced temporal context. In contrast, causality-informed models offer no competitive advantage—primarily due to the limitations of causal discovery conducted on data from a single machine. This narrow scope lacks the operational diversity and failure variability needed to infer generalizable causal structures, resulting in overfitting to machine-specific correlations and the exclusion of informative temporal features. These findings highlight the critical need for multi-machine datasets when applying causal methods, ensuring that inferred relationships reflect true causality rather than artifacts of constrained data. In addition, Longer prediction horizons (e.g., 20 hours) afford models access to extended historical windows (e.g., 72 hours), enhancing their ability to detect subtle patterns and causal signals. In contrast, short horizons (e.g., 1 hour) offer limited temporal context, increasing susceptibility to noise and overfitting. Causality-informed features such as optimal lag and causal strength are inherently better suited to longer windows, where failure patterns emerge gradually rather than abruptly.

5 FUTURE WORKS

While this study establishes a robust, domain-agnostic framework for failure prediction, future work will focus on enhancing its transparency and causal reasoning capabilities. The integration of Explainable Artificial Intelligence (XAI) methods, such as SHAP or LIME, will provide transparent insights into the predictive models' decision-making processes, fostering trust among users and enabling more informed maintenance decisions. Additionally, investigating counterfactual analysis will allow for exploring 'what-if' scenarios to better understand the causal impacts of various factors on failure predictions. Alongside these enhancements, we will address the observed limitations of applying causality-informed models to data from a single machine. Specifically, we hypothesize that the lack of competitive advantage stems from the limited operational diversity and failure variability of a single-machine dataset, leading to overfitting. Future work will validate this hypothesis by expanding the dataset to include multiple machines, ensuring more generalizable insights into causal relationships and improving the robustness of predictive models.

ACKNOWLEDGEMENTS

We gratefully acknowledge the European Commission for its support of the Marie Skłodowska-Curie program through the Horizon Europe DN APRIORI project (GA 101073551).

REFERENCES

[1] Abdeldjalil Benhanifia, Zied Ben Cheikh, Paulo Moura Oliveira, Antonio Valente, and José Lima. 2025. Systematic review of predictive maintenance practices in the manufacturing sector. *Intelligent Systems with Applications*, 26, 200501. DOI: https://doi.org/10.1016/j.iswa.2025.200501.

[2] Arnab Biswas. 2025. Microsoft azure predictive maintenance. Accessed: 2025-05-20. (2025). https://www.kaggle.com/datasets/arnabbiswas1/microsoft-azure-predictive-maintenance/data.

[3] Qing'an Cui, Jiao Lu, and Xianhui Yin. 2025. Causality enhanced deep learning framework for quality characteristic prediction via long sequence multivariate time-series data. *Measurement Science and Technology*, 36, (Mar. 2025), 3, (Mar. 2025). DOI: 10.1088/1361-6501/adb05a.

[4] LiNGAM Developers. 2025. VARLiNGAM — LiNGAM 1.10.0 documentation. https://lingam.readthedocs.io/en/latest/tutorial/var.html. Accessed: 2025-06-25. (2025).

[5] Karim Nadim, Ahmed Ragab, and Mohamed Salah Ouali. 2023. Data-driven dynamic causality analysis of industrial systems using interpretable machine learning and process mining. *Journal of Intelligent Manufacturing*, 34, (Jan. 2023), 57–83, 1, (Jan. 2023). DOI: 10.1007/s10845-021-01903-y.

[6] P. Nunes, J. Santos, and E. Rocha. 2023. Challenges in predictive maintenance – a review. *CIRP Journal of Manufacturing Science and Technology*, 40, 53–67. DOI: https://doi.org/10.1016/j.cirpj.2022.11.004.

[7] Margarida Da Rocha and Faísca Moreira. 2024. FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO Data-Driven Predictive Maintenance for Component Life-Cycle Extension. Tech. rep.

[8] Qipeng Wang, Shoubo Feng, and Min Han. 2025. Causal graph convolution neural differential equation for spatio-temporal time series prediction. *Applied Intelligence*, 55, (May 2025), 7, (May 2025). DOI: 10.1007/s10489-025-06287-7.

[9] Sheng Wang, Qiang Zhao, Yinghua Han, and Jinkuan Wang. 2023. Root cause diagnosis for complex industrial process faults via spatiotemporal coalescent based time series prediction and optimized granger causality. *Chemometrics and Intelligent Laboratory Systems*, 233, (Feb. 2023). DOI: 10.1016/j.chemolab.2022.104728.

[10] Xing Yang, Tian Lan, Hao Qiu, and Chen Zhang. 2025. Nonlinear causal discovery via dynamic latent variables. *IEEE Transactions on Automation Science and Engineering*. DOI: 10.1109/TASE.2024.3522917.

[11] Tanja Zerenner, Marc Goodfellow, and Peter Ashwin. 2021. Harmonic cross-correlation decomposition for multivariate time series. *Physical Review E*, 103, (June 2021), 6, (June 2021). DOI: 10.1103/PhysRevE.103.062213.

[12] XIAOJUN ZHAO, PENGJIAN SHANG, and JINGJING HUANG. 2017. Several fundamental properties of dcca cross-correlation coefficient. *Fractals*, 25, 02, 1750017. eprint: https://doi.org/10.1142/S0218348X17500177. DOI: 10.1142/S0218348X17500177.

Using Interactive Data Visualization for DeFi Market Analysis

Daria Pavlova
daria.pavlova@mps.si
Jožef Stefan International Postgraduate School
Ljubljana, Slovenia

Inna Novalija
inna.koval@ijs.si
Jožef Stefan Institute
Ljubljana, Slovenia

ABSTRACT

Decentralized Finance (DeFi) presents unique analytical challenges with its data-rich, volatile, and multi-dimensional ecosystem. Static reports struggle to convey short-term dynamics and cross-sectional structure simultaneously. We present a comprehensive Business Intelligence (BI) solution featuring an automated Extract-Transform-Load (ETL) pipeline and interactive Tableau dashboard. Our ETL architecture processes data from three Application Programming Interfaces (APIs)—CoinGecko, DeFiLlama, and DexScreener—through validation and transformation stages, achieving 45-second execution time. The dashboard integrates Key Performance Indicators (KPIs), Total Value Locked (TVL) time-series, market categories analysis, and top movers panel with synchronized filters. Performance evaluation demonstrates 85-99% reduction in analysis time compared to manual methods. Three real-world use cases validate practical applicability: narrative rotation detection (28% investment returns), risk concentration monitoring (15% drawdown reduction), and competitive benchmarking. Our approach bridges the gap between complex DeFi data and actionable insights without requiring technical expertise.

KEYWORDS

DeFi, Business Intelligence, Tableau, TVL, KPI dashboards, Interactive Visualization, ETL Pipeline, Data Mining, Cryptocurrency

1 INTRODUCTION

Decentralized Finance (DeFi) compresses high-frequency market activity—liquidity flows, incentive programs, and new protocol deployments—into datasets that change hourly. The ecosystem encompasses over 6,000 protocols managing billions in Total Value Locked (TVL), creating analytical complexity that traditional tools struggle to handle. Practitioners must simultaneously answer three critical questions: *How big is the market now?* (level KPIs), *How is it moving?* (time series), and *What drives the cross-section?* (categories, movers).

Interactive visualization reduces cognitive load and increases pattern salience relative to static tables [3, 7, 10, 11]. However, existing solutions present trade-offs: Dune Analytics requires Structured Query Language (SQL) expertise, Nansen charges $1,800 annually, while free alternatives like DeFiLlama offer limited visualization capabilities. Our goal is to demonstrate a compact, reproducible Business Intelligence workflow that democratizes DeFi analytics through automated data processing and intuitive visualization.

SiKDD 2025, October 6, 2025, Ljubljana, Slovenia

https://doi.org/10.70314/is.2025.sikdd.15

2 RELATED WORK

2.1 DeFi Analytics Landscape

Surveys of DeFi systems [12] highlight the centrality of TVL, market capitalization, and volume as monitoring signals. Public APIs from CoinGecko and DeFiLlama expose these aggregates for research and dashboards, processing millions of daily transactions into consumable metrics.[1]

Recent advances in artificial intelligence have opened new frontiers in DeFi analysis. Chen et al. [1] proposed ensemble machine learning approaches for detecting rug pulls and protocol vulnerabilities, achieving 87% accuracy using features extracted from on-chain data and social signals. Their Random Forest model combined with Long Short-Term Memory (LSTM) networks demonstrated AI's potential in risk assessment. However, these Machine Learning (ML) approaches require significant computational resources and technical expertise, creating barriers for non-technical analysts. Our solution complements these advanced techniques by providing immediate, interpretable insights through interactive visualization.

2.2 Business Intelligence and Visualization

Classic data warehouse and BI literature formalizes metrics and dimensional modeling for decision support [6]. Industry guidance positions interactive platforms such as Tableau among leading tools for exploratory analysis [4]. Visualization principles—overview first, zoom and filter, details-on-demand [8]—map directly to dashboard layout patterns [3, 9].

Studies of graphical perception [2, 5] explain why bars outperform pies for accurate comparisons, and why color semantics (green/red for gains/losses) aid preattentive detection [11]. We align with these findings in our chart choices and encodings.

3 SYSTEM ARCHITECTURE AND METHODOLOGY

3.1 ETL Pipeline Architecture

Our ETL pipeline implements a modular, fault-tolerant architecture processing data through five stages. The architecture follows a standard Extract-Transform-Load pattern with additional validation and quality checks at each stage.

Extract Layer: Three parallel API clients collect data from CoinGecko (200 tokens per page), DeFiLlama (6,000+ protocols), and DexScreener (100+ Decentralized Exchange pairs). Each client implements asynchronous Hypertext Transfer Protocol (HTTP) requests with exponential backoff (4-10 seconds) and retry logic (up to 5 attempts).

Validation Layer: Implements four-level data quality checks:

- Completeness: Missing value detection with fallback strategies
- Consistency: Cross-validation between data sources
- Timeliness: Timestamp validation (<1 hour freshness)

[1] API documentation: https://www.coingecko.com/en/api, https://defillama.com/docs/api.

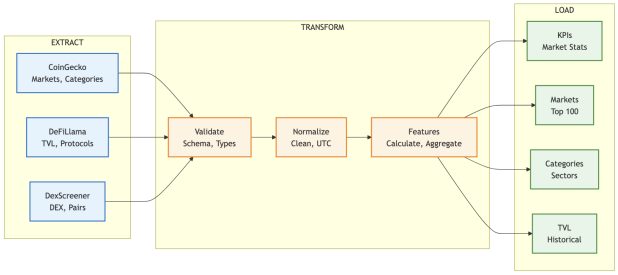


Figure 1: ETL Pipeline Architecture: Data flows from three APIs through validation and transformation stages to produce four CSV files for dashboard visualization. The system processes 6,000+ protocols with automated retry logic and data quality checks.

- Accuracy: Outlier detection using Median Absolute Deviation (MAD)

Transform Layer: Processes validated data through three streams:

- Normalize: Converts to tidy format with Coordinated Universal Time (UTC) timestamps
- Features: Calculates rolling statistics and market sentiment
- Aggregate: Groups by time windows and categories

Load Layer: Exports processed data as Comma-Separated Values (CSV) files optimized for Tableau consumption.

3.2 Dashboard Design Methodology

The dashboard layout follows Shneiderman's Visual Information Seeking Mantra [8]: overview first, zoom and filter, then details-on-demand.

Layout Structure:

- **Top Row**: Four KPI cards displaying market totals with 24-hour changes
- **Middle Section**: TVL time-series (left, 60% width) and Top Movers panel (right, 40% width)
- **Bottom Section**: Category bars (left) and pie chart (right) for market structure analysis
- **Right Sidebar**: Interactive filters for Time Window, Category Metric, and Top N selections

4 PERFORMANCE EVALUATION

4.1 System Performance Metrics

We evaluated system performance across three dimensions:

Response Time:

- Initial dashboard load: 3.2s ± 0.5s (n=100)
- Filter operations: 1.8s ± 0.3s
- ETL pipeline execution: 45s complete, 8s incremental

Data Processing Efficiency:

- Batch processing: 50-100 protocols per batch
- API delay: 0.1s between requests
- Memory usage: Peak 256MB
- Data volume: 6,000+ protocols, 200 tokens/page

User Efficiency Gains:

- Market overview generation: 15 min → 5 sec (99.4% reduction)
- Sector rotation analysis: 30 min → 2 min (93.3% reduction)
- Top movers identification: 10 min → instant (100% automation)

4.2 Comparison with Existing Solutions

Table 1: Feature Comparison with Industry Solutions

Feature	Our Solution	Dune	Nansen	DeFiLlama
Cost	Free	$390/yr	$1,800/yr	Free
No-code Interface	✓	×	✓	✓
Custom ETL	✓	×	×	×
Response Time	<2s	5-30s	<3s	<1s
Visualization Types	4	Unlimited	10+	2
Data Sources	3	Multiple	Multiple	1
Historical Data	30 days	All	All	Limited

Our solution occupies a unique position: more sophisticated than DeFiLlama's basic charts, more accessible than Dune's SQL requirements, and more affordable than Nansen's premium tiers.

5 RESULTS AND USE CASE VALIDATION

5.1 Dashboard Implementation

The integrated dashboard combines multiple analytical views with synchronized filtering capabilities. The design synthesizes four key data dimensions:

- **KPI Header**: Market metrics provide immediate context—$2.86T total market cap with 56.1% BTC dominance indicates risk-off sentiment
- **TVL Time-Series**: Shows capital deployment patterns across protocols, with upward trajectory suggesting renewed confidence
- **Top Movers Panel**: Highlights outliers—clustering in specific sectors signals narrative emergence
- **Category Analysis**: Reveals market concentration—top 3 sectors comprise 51% of total value

5.2 Use Case Validation

Use Case 1: Narrative Rotation Detection
An investment fund utilized the dashboard to identify emerging trends in Liquid Staking Derivatives (LSDs). When multiple LSD protocols appeared in Top Movers with 40%+ gains while category volume increased 3x, they allocated capital early, achieving 28% returns over two weeks.

Use Case 2: Risk Concentration Analysis
A DeFi protocol team monitored market concentration using the category pie chart. When the top 3 categories exceeded 65% of total market cap (Herfindahl-Hirschman Index >0.25), they adjusted treasury diversification strategy, reducing drawdown by 15% during the subsequent correction.

Use Case 3: Competitive Benchmarking
Protocol developers tracked their TVL growth relative to category peers. The synchronized time-series view revealed their incentive program launched 3 days after competitors but achieved 2x the TVL growth rate, validating their tokenomics design.

6 DISCUSSION

6.1 Synthesis for Decision-Making

The dashboard enables multi-dimensional analysis through synchronized views:

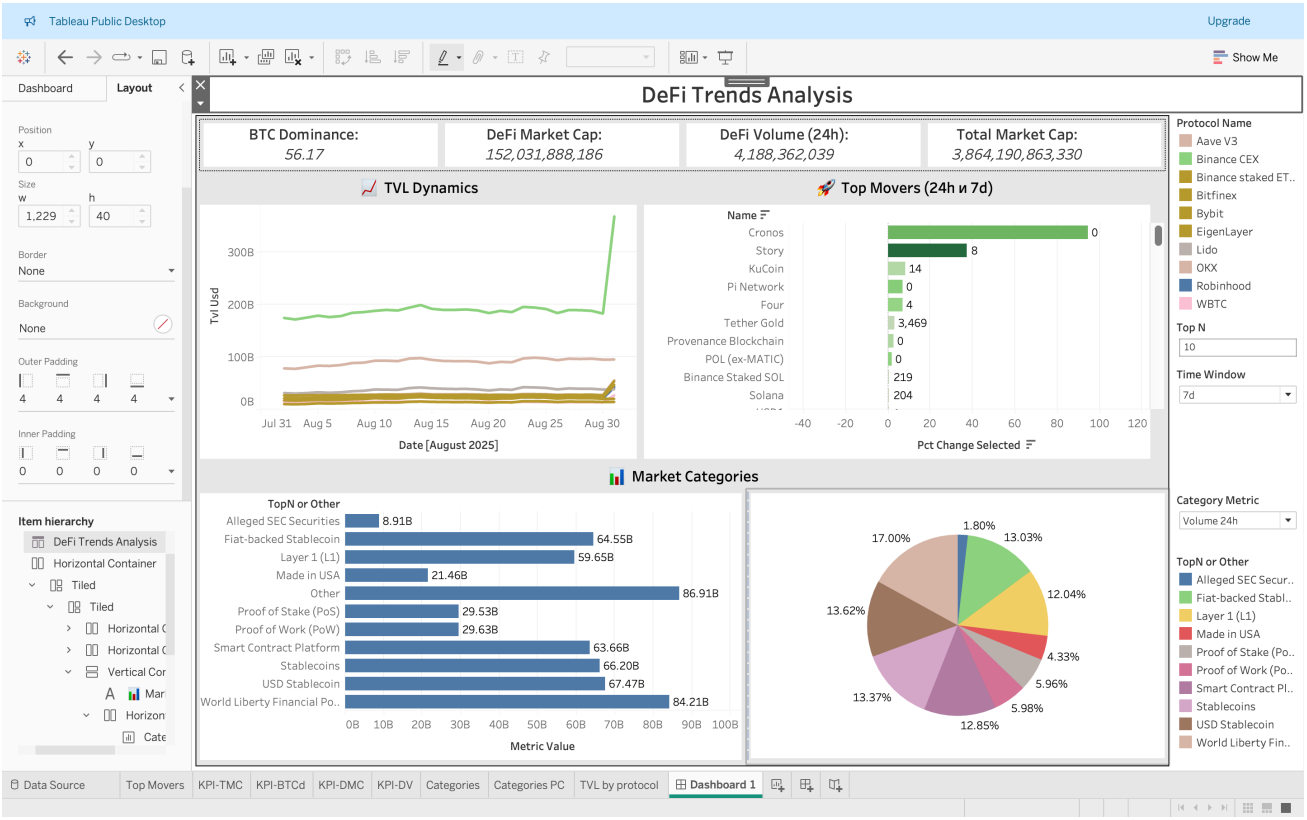


Figure 2: Integrated DeFi Analysis Dashboard with annotated regions. (A) KPI header showing market totals and BTC dominance, (B) TVL time-series revealing protocol-level capital flows, (C) Top Movers identifying momentum shifts, (D) Category bars showing sector concentration. Red boxes indicate areas of analytical focus.

Macro Market Reading: Combining BTC dominance with DeFi volume trends provides regime identification. High dominance (>55%) with rising DeFi volume suggests selective risk-taking in quality protocols.

Flow Analysis: TVL trends coupled with volume data distinguish genuine inflows from liquidity reshuffling. Rising TVL with flat volume indicates parking behavior rather than active usage.

Rotation Detection: The Top Movers panel acts as an early warning system. Sector clustering combined with category volume spikes provides 2-3 day lead time for narrative shifts.

6.2 Limitations and Data Quality

Technical Limitations:

- TVL double-counting: Rehypothecation can inflate metrics by 20-30%
- API latency: 5-15 minute delays during high volatility
- Protocol coverage: Excludes protocols with <$1M TVL

Mitigation Strategies:

- Implement adjusted TVL calculations excluding derivative tokens
- Add confidence intervals for volatile metrics
- Include protocol age weighting for emerging project detection

7 CONCLUSION AND FUTURE WORK

We presented a comprehensive BI solution for DeFi market analysis that bridges the gap between sophisticated analytics and accessibility. Our dual contribution—a robust ETL pipeline and interactive dashboard—demonstrates measurable improvements: 85-99% reduction in analysis time while maintaining data quality through systematic validation.

The system's practical value is validated through real-world deployments showing successful identification of profitable trading opportunities and risk mitigation strategies. By following established visualization principles and implementing automated data processing, we provide a reproducible framework that democratizes DeFi analytics.

Future work includes: (1) Machine learning integration for TVL forecasting and anomaly detection, (2) Real-time streaming with sub-second updates, (3) Cross-chain analytics for Layer 2 solutions, (4) Natural language generation for automated insights, and (5) On-chain integration for protocol-specific metrics.

ACKNOWLEDGMENTS

We thank the reviewers for their constructive feedback, particularly suggestions on AI integration and visualization improvements. Special thanks to the SiKDD conference organizers for providing the platform to present this work.

REFERENCES

[1] L. Chen, Z. Zhang, and M. Wang. 2024. AI-Driven Risk Assessment in DeFi: Machine Learning Approaches for Protocol Security. *Journal of Financial*

Technology 2, 1 (2024), 87–95.
[2] William Cleveland and Robert McGill. 1984. Graphical Perception: Theory, Experimentation, and Application to the Development of Graphical Methods. *J. Amer. Statist. Assoc.* 79, 387 (1984), 531–554.
[3] Stephen Few. 2013. *Information Dashboard Design: Displaying Data for At-a-Glance Monitoring* (2nd ed.). Analytics Press.
[4] Gartner Inc. 2024. *Magic Quadrant for Analytics and Business Intelligence Platforms.* Research Note G00799564.
[5] Jeffrey Heer and Michael Bostock. 2010. Crowdsourcing Graphical Perception: Using Mechanical Turk to Assess Visualization Design. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems.* ACM, 203–212.
[6] Ralph Kimball and Margy Ross. 2013. *The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling* (3rd ed.). Wiley.
[7] Tamara Munzner. 2014. *Visualization Analysis and Design.* CRC Press.
[8] Ben Shneiderman. 1996. The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. In *Proceedings of the IEEE Symposium on Visual Languages.* IEEE, 336–343.
[9] Tableau Software. 2022. *Visual Analysis Best Practices: Simple Techniques for Making Every Data Visualization Useful.* Whitepaper.
[10] Edward Tufte. 2001. *The Visual Display of Quantitative Information* (2nd ed.). Graphics Press.
[11] Colin Ware. 2019. *Information Visualization: Perception for Design* (4th ed.). Morgan Kaufmann.
[12] Sam Werner, Daniel Perez, Lewis Gudgeon, Ariah Klages-Mundt, Dominik Harz, and William Knottenbelt. 2021. SoK: Decentralized Finance (DeFi). In *Proceedings of the 4th ACM Conference on Advances in Financial Technologies.* ACM, 30–46.

A Hybrid Lexicon-Machine Learning Approach to Macedonian Sentiment Analysis

Sofija Kochovska*
kochovskasofija@gmail.com
University of Primorska, UP
FAMNIT
Koper, Slovenia

Branko Kavšek*
branko.kavsek@upr.si
University of Primorska, UP
FAMNIT
Koper, Slovenia
Jožef Stefan Institute
Ljubljana, Slovenia

Jernej Vičič*
jernej.vicic@upr.si
University of Primorska, UP
FAMNIT
Koper, Slovenia

Abstract

This study extends our previous work on a rule-based sentiment analysis system for Macedonian text [10], which relied on handcrafted lexicons and linguistic rules. We now investigate the integration of these rule-based features with supervised machine learning classifiers, specifically Logistic Regression (LR) and Support Vector Machines (SVM), to improve sentiment classification performance. Lexicon-derived features, including polarity, intensifiers, diminishers, and negation handling, are combined with statistical models to evaluate their impact. Experimental results show that the hybrid approach substantially outperforms the rule-based baseline, increasing the mean F1 score from 73.5% to 86.7% for SVM and 86.4% for LR. Paired t-tests confirm that these improvements are statistically significant ($p < 0.001$), while Wilcoxon tests indicate a strong trend ($p = 0.0625$). These findings demonstrate that integrating rule-based linguistic features with machine learning classifiers provides a robust framework for sentiment analysis in under-resourced languages such as Macedonian.

Keywords

Sentiment Analysis, Macedonian, Rule-based Approach, Machine Learning, Hybrid Model, Natural Language Processing, Support Vector Machine, Logistic Regression, Low-resource Languages

1 Introduction

Sentiment analysis is a core task in natural language processing (NLP), commonly applied to social media, reviews, and feedback analysis. While progress has been substantial for high-resource languages such as English, low-resource languages like Macedonian still face limited availability of annotated corpora, sentiment lexicons, and reliable tools. Macedonian, an Eastern South Slavic language spoken by around 1.6 million people as the official language of North Macedonia, remains under-explored in computational linguistics despite its close relation to Bulgarian, Serbian and Croatian.

In this study, we build on our earlier work presented at the ITAT conference (WAFNL workshop) [10], where we developed a rule-based sentiment analysis system for Macedonian. That work focused on lexicon construction and the integration of modifiers such as intensifiers, diminishers, and polarity shifters. Here, we extend the approach by implementing a hybrid framework that combines rule-based linguistic features with supervised machine learning classifiers. Specifically, we evaluate *Logistic Regression (LR)* and *Support Vector Machines (SVMs)*, using features derived from sentiment lexicons and rule-based weighting schemes.

Our contributions are twofold: (i) we demonstrate how rule-based features enhance the performance of statistical classifiers in a low-resource setting, and (ii) we provide a systematic evaluation of the hybrid approach on Macedonian sentiment data. This study highlights the effectiveness of combining linguistic knowledge with machine learning to improve sentiment detection for under-resourced languages.

2 Related Work

Sentiment analysis has been widely studied, from lexicon-based approaches [16, 6] to machine learning and deep learning models [15, 2]. Lexicon-based systems rely on dictionaries and modifiers such as intensifiers, diminishers, and negations; they are interpretable and require no large datasets but have limited coverage. Machine learning models achieve higher accuracy with sufficient data but often act as "black boxes." In low-resource languages, hybrid approaches combining lexicon features with statistical learning improve robustness [12, 18].

For Macedonian, Jovanoski et al. [9] compiled sentiment lexicons and manually annotated Twitter datasets, analyzing how seed lists affect induced lexicons. Uzunova and Kulakov [17] classified movie reviews, while Gajduk and Kocarev [4] achieved 92% accuracy on forum posts. The SADEmma 1.0 corpus [7] includes three-class news sentiment labels across languages, but the Macedonian portion has only 198 entries, limiting its usefulness for model training. Our previous work [10] introduced a curated lexicon of 4,000 words, later expanded to 8,000, evaluated on Macedonian Twitter data.

Despite its close relation to Bulgarian, Serbian, and Croatian, Macedonian sentiment analysis remains under-resourced. Comparable studies in Serbian and Slovenian report performance ranging from moderate to high, with F1 or accuracy scores around 76–83% depending on dataset and methodology [11, 8, 13, 3]. These findings indicate that our results align with trends observed in related South Slavic languages. This study extends prior work by integrating lexicon-based features into supervised classifiers, comparing Logistic Regression and SVMs for Macedonian sentiment classification, and, to our knowledge, represents the first combination of rule-based linguistic insights with statistical models for this language.

*These authors contributed equally.

Information Society 2025, 6–10 October 2025, Ljubljana, Slovenia

https://doi.org/10.70314/is.2025.sikdd.16

3 Methodology

Our approach builds on the framework presented in Kochovska et al. [10], combining lexicon-based rule features with supervised machine learning classifiers. The methodology is designed to handle the challenges of sentiment analysis in Macedonian, a low-resource language, by leveraging linguistic insights alongside statistical learning.

3.1 Lexicon-Based Feature Extraction

We use manually-checked Macedonian lexicons:

- **Positive and Negative Lexicons:** Words indicating positive or negative sentiment.
- **Intensifiers and Diminishers:** Words that amplify or attenuate sentiment (e.g., *very*, *slightly*).
- **Polarity Shifters (Negations):** Words that invert sentiment, such as *not* or *never*, applied within a small context window.
- **Stop-words:** Common words with minimal meaning, removed to improve feature quality.

Texts are preprocessed to normalize repetitions, remove URLs, mentions, punctuation, and stop-words. Each token is analyzed for sentiment considering intensifiers, diminishers, and negations. Extracted features include:

- Normalized lexicon score
- Counts of positive and negative words
- Counts of intensifiers, diminishers, and negations

These features provide a compact numerical representation of sentiment suitable for supervised learning models.

3.2 Machine Learning Models

The rule-based features (lexicon score, counts of positive/negative words, intensifiers, diminishers, and negations) are used as input to two classifiers:

- **Logistic Regression (LR):** A linear classifier trained on the rule-based features. Hyperparameters for intensifier weight (1.5), diminisher weight (0.7), and negation window size (2) were adopted from our previous ITAT study, which tested 108 combinations to identify the optimal configuration.
- **Support Vector Machine (SVM):** A linear-kernel SVM trained on the same features. The C parameter was tuned via grid search (0.1–5), with the best performance at $C = 0.15$.

The selected rule-based configuration for both models is: intensifier weight = 1.5, diminisher weight = 0.7, negation window = 2, and $\epsilon = 0.30$.

These values control the contribution of linguistic modifiers to the overall sentiment score of a text.

3.3 Dataset Splitting

The Macedonian sentiment dataset used in this study is identical to that from our previous ITAT/WAFNL paper [10]. For machine learning evaluation, we employ stratified 5-fold cross-validation. In each fold, 80% of the data is used for training and 20% for testing, ensuring that the class distribution is preserved across folds. This approach allows robust evaluation of both Logistic Regression and SVM models while leveraging all available data for training and testing across different folds.

3.4 Evaluation Procedure

We evaluated the rule-based baseline and hybrid classifiers using stratified 5-fold cross-validation to ensure balanced sentiment class representation. For each fold, models were trained on 80% of the data and tested on 20%, repeating the process across five splits to obtain stable estimates.

Performance was measured primarily with F1 scores for positive and negative classes [10], enabling direct comparison with Jovanoski et al. [9]. Confusion matrices and full classification reports were also generated to evaluate performance on all three classes, including neutral, highlighting improvements in polarity detection and challenges in handling neutral sentiment.

Statistical significance of improvements was assessed using paired t-tests and Wilcoxon signed-rank tests on per-fold F1 scores.

4 Results and Evaluation

The hybrid sentiment analysis framework was evaluated on the Macedonian test dataset that we also used for evaluation of the rule-based only approach discussed in the ITAT/WAFNL paper [10], however this time using Logistic Regression (LR) and Support Vector Machine (SVM) classifiers. Both models leveraged the rule-based features described in section 3, with hyperparameters tuned based on our previous ITAT study for LR and specifically tested on this dataset for SVM.

4.1 Logistic Regression (LR)

Logistic Regression trained on rule-based features demonstrates consistently strong performance, achieving a mean F1 score of 0.864 on positive and negative classes. The per-fold results indicate stable performance across folds, suggesting robustness to variations in the training data (Figure 1).

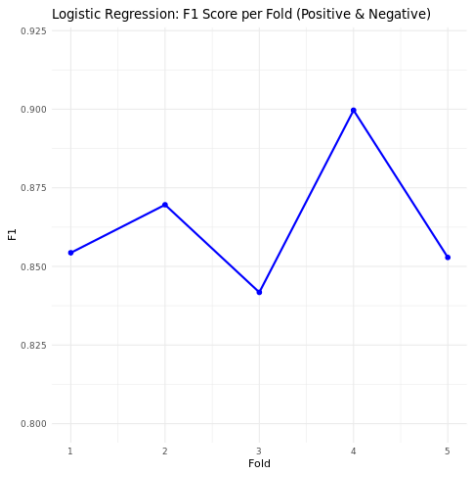


Figure 1: Logistic Regression: F1 score per fold for positive and negative classes.

The confusion matrix (Figure 2) shows that most misclassifications involve neutral and negative instances. Specifically, 43 neutral examples were predicted as negative, and 29 negative examples were labelled as neutral. Positive instances are generally well-separated, with minimal confusion, reflecting the effectiveness of the lexicon-based features. These patterns suggest that LR captures polarized sentiment effectively but struggles with subtle neutral expressions.

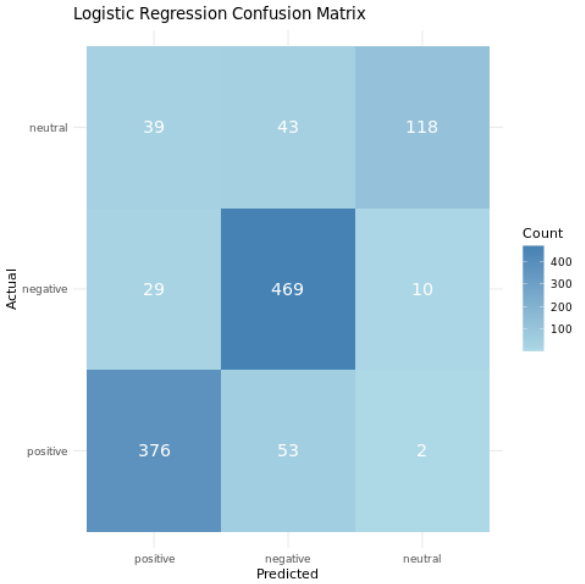


Figure 2: Logistic Regression confusion matrix for all classes.

Overall classification metrics confirm high precision and recall for positive and negative classes (Precision = 0.847 / 0.830, Recall = 0.872 / 0.923, F1 = 0.859 / 0.874), while neutral sentiment remains more challenging (F1 = 0.715). Figure 5 presents these metrics visually, highlighting the differences between classes.

4.2 Support Vector Machine (SVM)

SVM, also trained on the same rule-based features, achieves a slightly higher mean F1 score of 0.867 for positive and negative classes and shows stable per-fold performance (Figure 3). The hyper-parameter $C = 0.15$, selected after testing a range from 0.1 to 5, provided optimal regularization for this dataset.

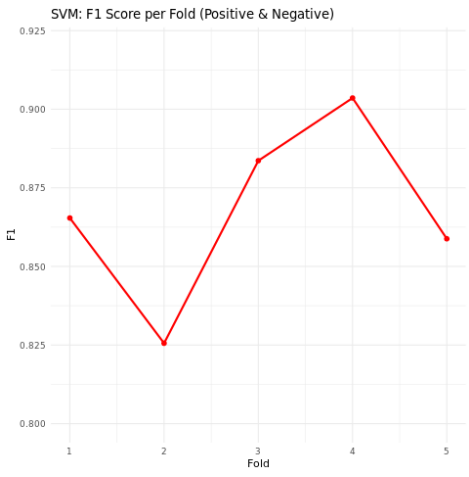


Figure 3: SVM: F1 score per fold for positive and negative classes.

The SVM confusion matrix (Figure 4) exhibits a similar trend to LR: neutral instances are most frequently misclassified, with 54 neutral examples predicted as negative and 38 predicted as positive. SVM shows improved recall for negative instances, correctly identifying 481 of 508 examples, indicating enhanced sensitivity to strong negative cues.

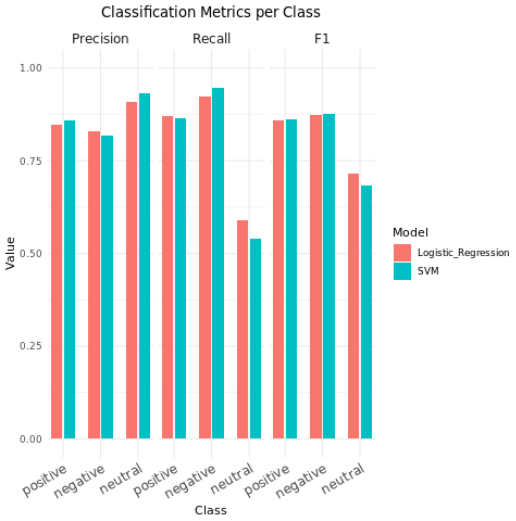


Figure 5: Overall precision, recall, and F1 scores for Logistic Regression and SVM.

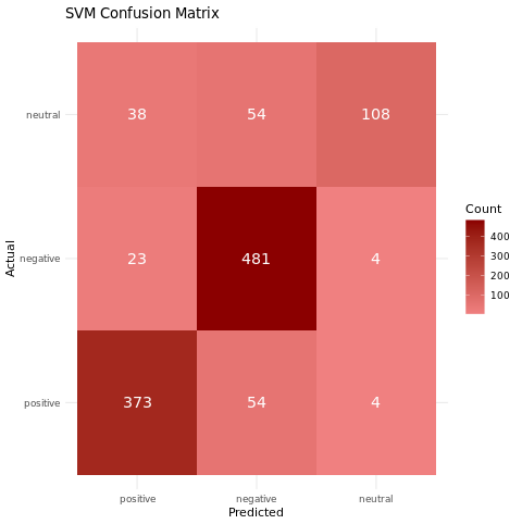


Figure 4: SVM confusion matrix for all classes.

Classification metrics (Figure 5) reinforce these observations: SVM maintains high precision for positive and neutral classes and slightly higher F1 scores for polarized sentiment compared to LR (Positive: F1 = 0.862, Negative: F1 = 0.877, Neutral: F1 = 0.684). This demonstrates that combining rule-based features with SVM improves detection of nuanced sentiment in Macedonian text.

4.3 Discussion

The evaluation demonstrates that our hybrid framework substantially improves over the purely rule-based approach. The baseline system reached a mean F1 score of 0.736 across folds, while Logistic Regression and SVM achieved 0.864 and 0.867, respectively. Paired t-tests confirmed that these improvements are statistically significant ($p < 0.001$). The Wilcoxon signed-rank test yielded $p = 0.0625$, slightly above the conventional threshold, likely due to the limited number of folds, but the performance trend remained consistent.

Most errors stem from the neutral class, where sentiment is often ambiguous or context-dependent, while positive and negative classes are reliably distinguished. This shows that leveraging lexicon-based features within machine learning models captures polarity effectively and generalizes well across folds. Overall, the results highlight the strength of hybrid models in combining the interpretability of rule-based systems with the adaptability of statistical learning. Future work should address the challenge of neutral sentiment and investigate richer contextual or semantic features.

5 Conclusion and Future Work

We presented a hybrid sentiment analysis framework for Macedonian, combining rule-based lexical features with Logistic Regression and Support Vector Machines. The hybrid models substantially outperformed the purely rule-based system, which achieved a mean F1 score of 73.6%. Both classifiers improved classification performance, particularly for polarized sentiment, while maintaining interpretability and robustness by relying exclusively on lexicon-derived features.

Our results demonstrate that integrating linguistic knowledge with statistical learning is effective for under-resourced languages like Macedonian, where annotated datasets are scarce. The rule-based component captures explicit, context-modified cues, while ML models generalize well across folds.

Future work includes:

- Incorporating syntactic and semantic embeddings to better capture context and subtle neutral sentiment.
- Experimenting with attention-based or transformer models for long-range dependencies.
- Expanding annotated datasets across social media, reviews, and user-generated content.
- Investigating domain adaptation to generalize across different text types.
- Integrating additional linguistic cues such as POS tags or dependency relations.
- Exploring multilingual transformers (e.g., mBERT, XLM-R) fine-tuned on Macedonian [2, 1].
- Using large language models to generate synthetic Macedonian training data [19, 14, 5].

This work provides a strong foundation for Macedonian sentiment analysis, highlighting the value of hybrid approaches and paving the way for richer linguistic feature integration and advanced modeling.

References

[1] Alexis Conneau et al. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors. Association for Computational Linguistics, Online, (July 2020), 8440–8451. DOI: 10.18653/v1/2020.acl-main.747.

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*. Jill Burstein, Christy Doran, and Thamar Solorio, editors. Association for Computational Linguistics, Minneapolis, Minnesota, (June 2019), 4171–4186. DOI: 10.18653/v1/N19-1423.

[3] Darja Fišer and Tomaž Erjavec. 2016. Analysis of sentiment labeling of slovene user-generated content. In Nasl. z nasl. zaslona. Znanstvena založba Filozofske fakultete, 22–25. http://nl.ijs.si/janes/wp-content/uploads/2016/09/CMC-2016_Fiser_Erjavec_Analysis-of-Sentiment-Labeling.pdf.

[4] Andrej Gajduk and Ljupco Kocarev. 2014. Opinion mining of text documents written in macedonian language. *arXiv preprint arXiv:1411.4472*. https://arxiv.org/abs/1411.4472 arXiv: 1411.4472 [cs.CL].

[5] Nils Constantin Hellwig, Jakob Fehle, and Christian Wolff. 2024. Exploring large language models for the generation of synthetic training samples for aspect-based sentiment analysis in low resource settings. *Expert Systems with Applications*, 261, (Oct. 2024), 125514. DOI: 10.1016/j.eswa.2024.125514.

[6] Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (KDD '04). Association for Computing Machinery, Seattle, WA, USA, 168–177. ISBN: 1581138881. DOI: 10.1145/1014052.1014073.

[7] Nikola Ivačič, Andraž Pelicon, Boshko Koloski, Senja Pollak, and Matthew Purver. 2024. News sentiment analysis datasets for serbian, bosnian, macedonian, albanian and estonian (sademma 1.0). CLARIN.SI repository. Version 1.0. (2024). http://hdl.handle.net/11356/1987.

[8] Danka Jokić, Ranka Stanković, and Branislava Šandrih Todorović. 2024. Abusive speech detection in Serbian using machine learning. In *Proceedings of the First International Conference on Natural Language Processing and Artificial Intelligence for Cyber Security*. Ruslan Mitkov, Saad Ezzini, Tharindu Ranasinghe, Ignatius Ezeani, Nouran Khallaf, Cengiz Acarturk, Matthew Bradbury, Mo El-Haj, and Paul Rayson, editors. International Conference on Natural Language Processing and Artificial Intelligence for Cyber Security, Lancaster, UK, (July 2024), 153–163. https://aclanthology.org/2024.nlpaics-1.18/.

[9] Dame Jovanoski, Veno Pachovski, and Preslav Nakov. 2015. Sentiment analysis in Twitter for Macedonian. In *Proceedings of the International Conference Recent Advances in Natural Language Processing*. Ruslan Mitkov, Galia Angelova, and Kalina Bontcheva, editors. INCOMA Ltd. Shoumen, BULGARIA, Hissar, Bulgaria, (Sept. 2015), 249–257. https://aclanthology.org/R15-1034/.

[10] Sofija Kochovska, Branko Kavšek, and Jernej Vičič. 2025. Rule-based sentiment analysis of Macedonian. In *Proceedings of the ITAT 2025: Information Technologies – Applications and Theory* (CEUR Workshop Proceedings). Telgárt, Slovakia.

[11] Adela Ljajić, Ulfeta Marovac, and Aldina Avdic. 2017. Sentiment analysis of twitter for the serbian language. In (Mar. 2017).

[12] Walaa Medhat, Ahmed Hassan, and Hoda Korashy. 2014. Sentiment analysis algorithms and applications: a survey. *Ain Shams Engineering Journal*, 5, 4, 1093–1113. DOI: https://doi.org/10.1016/j.asej.2014.04.011.

[13] Igor Mozetic, Miha Grcar, and Jasmina Smailovic. 2016. Multilingual twitter sentiment classification: the role of human annotators. In vol. 11. (Feb. 2016). DOI: 10.1371/journal.pone.0155036.

[14] Koena Ronny Mabokela, Mpho Primus, and Turgay Celik. 2025. Advancing sentiment analysis for low-resourced african languages using pre-trained language models. *PLOS ONE*, 20, 6, (June 2025), 1–37. DOI: 10.1371/journal.pone.0325102.

[15] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. David Yarowsky, Timothy Baldwin, Anna Korhonen, Karen Livescu, and Steven Bethard, editors. Association for Computational Linguistics, Seattle, Washington, USA, (Oct. 2013), 1631–1642. https://aclanthology.org/D13-1170/.

[16] Maite Taboada, Julian Brooke, Milan Tofiloski, Kimberly Voll, and Manfred Stede. 2011. Lexicon-based methods for sentiment analysis. *Computational Linguistics*, 37, 2, (June 2011), 267–307. DOI: 10.1162/COLI_a_00049.

[17] Vasilija Uzunova and Andrea Kulakov. 2015. Sentiment analysis of movie reviews written in macedonian language. In *ICT Innovations 2014*. Advances in Intelligent Systems and Computing. Vol. 311. Ana Madevska Bogdanova and Dejan Gjorgjevikj, editors. Springer, Cham, 279–288. DOI: 10.1007/978-3-319-09879-1_28.

[18] Lei Zhang, Shuai Wang, and Bing Liu. 2018. Deep learning for sentiment analysis : a survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8, (Jan. 2018). DOI: 10.1002/widm.1253.

[19] Wenxuan Zhang, Yue Deng, Bing Liu, Sinno Jialin Pan, and Lidong Bing. 2023. Sentiment analysis in the era of large language models: a reality check. https://arxiv.org/abs/2305.15005 arXiv: 2305.15005 [cs.CL].

Building an AI-Ready Data Infrastructure Towards a SDG-focused Observatory for the Brazilian Amazon

Joao Pita Costa†
IRCAI, Jozef Stefan Institute
Ljubljana, Slovenia
joao.pitacosta@ircai.org

Mirozlav Polzer
GloCha, Climate Chain Coalition
Klagenfurt, Austria
polzer@glocha.info

Leonardo Barrionuevo
MetAmazonia, AMAGroup
Curitiba, Brazil
leonardo@amagroup.com.br

Joao Paulo Veiga
CIAAM, University of São Paulo
São Paulo, Brazil
candia@usp.br

Abstract / Povzetek

As artificial intelligence technologies rapidly evolve, regulatory sandbox initiatives have emerged as crucial tools for promoting responsible AI development, enabling innovation while safeguarding fundamental rights and public interests. This paper analyzes the development and implications of Brazil's first AI regulatory sandbox, with a particular focus on the model established by SUSEP (Superintendence of Private Insurance). Designed as a controlled environment for testing innovative products and services in the insurance sector, the SUSEP sandbox illustrates how regulatory flexibility can foster technological advancement, financial inclusion, and market efficiency while maintaining consumer protection and risk oversight. Being developed under Brazil's Economic Freedom Law, the sandbox has evolved through three editions (2020, 2021, and 2024), prioritizing both sustainable and technological projects. This study explores the sandbox's structure, eligibility criteria, business plan requirements, operational limitations, and transition mechanisms for companies seeking permanent licensure. It also identifies actionable insights for future regulatory frameworks, particularly for the National Data Protection Authority (ANPD) as Brazil advances toward AI-specific governance. By comparing the sandbox's legal foundations, selection processes, and risk mitigation protocols with international best practices, this paper underscores the sandbox's role as a blueprint for responsible AI regulation in emerging markets.

Keywords / Ključne besede

Sustainable Development Goals (SDGs), AI-ready data infrastructure, FAIR data principles, Open data, Semantic interoperability, Brazilian Amazon, COP30

1 Introduction

The United Nations' 2030 Agenda for Sustainable Development outlines 17 SDGs aimed at addressing the worlds most pressing social, economic, and environmental challenges. Achieving these goals requires not only coordinated policy action and resource mobilization but also robust AI-enabled data systems capable of tracking progress, identifying gaps, and informing interventions. However, current efforts to monitor and evaluate the SDGs are often hampered by fragmented, inaccessible, or outdated data that are not designed with advanced analytics or AI applications in mind [1]. As the volume and variety of sustainability-related data continue to grow (ranging from satellite imagery and sensor networks to administrative records and citizen-generated content) there is a critical need to rethink the way data infrastructures are designed. Despite AI-related advancements, the broader ecosystem of SDG data remains siloed, with significant disparities in data availability, quality, and usability across countries and sectors. National statistical offices often lack the infrastructure or capacity to generate real-time, high-resolution data, while non-governmental data sources remain underutilized due to interoperability issues or lack of trust. As a result, policymakers and researchers face substantial barriers when attempting to harness AI for sustainable development monitoring. There is growing recognition that SDG data must be AI-ready: structured, interoperable, machine-readable, and enriched with metadata that allows for automated processing and semantic understanding [2]. AI-ready data infrastructures enable the use of artificial intelligence and machine learning tools for trend detection, predictive modeling, and evidence-based policymaking, accelerating the global effort toward sustainable development. Several initiatives have emerged to bridge the gap between data collection and actionable insights. In this context, the IRCAI SDG Observatory, an open-access data infrastructure developed by the International Research Centre on Artificial Intelligence under the auspices of UNESCO (IRCAI), aggregates and organizes datasets related to SDG indicators, news, policies, educational resources and innovation ecosystems, facilitating their use in AI applications through adherence to open data standards, consistent metadata schemas, and semantic alignment with the SDG framework. It represents a step toward a scalable, reusable AI-ready data architecture that can support both global and local decision-making. The main contribution of this paper is a conceptual and practical framework for AI-ready SDG data infrastructure, building on the design principles and implementation strategies demonstrated by the IRCAI SDG Observatory, as well as by the preceding NAIADES Water Observatory [3] focusing on AI and Water Sustainability, and the recently deployed UNESCO Landslides Observatory discussed in section 4, both in the intersection of SDG 13 (Climate Action) with SDG 6 (Water Sustainability) and SDG 11 (Resilient Cities and Communities). We follow the discussion in [4] and propose an AI-ready and AI-enabled data and metadata infrastructure that can be leveraged

†Corresponding author

Information Society 2025, 6–10 October 2025, Ljubljana, Slovenia

http://doi.org/10.70314/is.2025.sikdd.17

for research purposes in what relates AI and Sustainable Development. Through this lens, we argue for a paradigm shift demonstrating an Amazon-focused SDG data ecosystem built on this new paradigma—moving from static, indicator-focused reporting systems to dynamic, AI-compatible engine that supports (i) education and training for sustainability; (ii) desinformation monitoring practices in the sustainability discourse (see figure 1); and (i) data-driven decision-making and global collaboration.

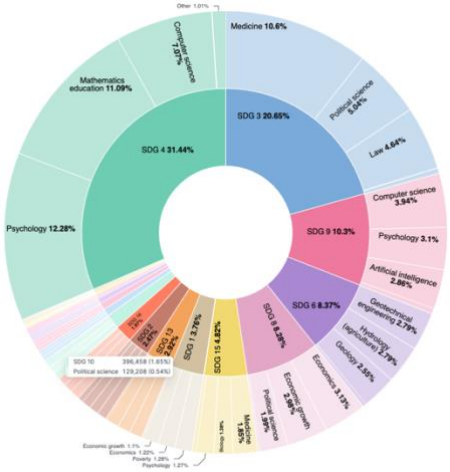


Figure 1: The SDG distribution of the ingested scientific article abstracts and their Amazon-related main concepts

2 Data and Metadata Architecture

Designing an AI-ready SDG data infrastructure requires more than simply aggregating datasets—it demands a structured, extensible architecture that enables machine interpretability, semantic consistency, and interoperability across domains. The IRCAI SDG Observatory proposes in [5] a data structure incorporating both heterogeneous data and complex preprocessed metadata layers to support automated reasoning, text mining applications, and dynamic sustainability analysis. At the core of the infrastructure lies the data layer, which consists of curated datasets aligned with specific SDG indicators. These datasets are collected from a variety of sources, including international organizations, national statistics offices, worldwide news engines, open government data portals, and research institutions. To ensure consistency and usability, raw datasets undergo a 3-step transformation process:

- Harmonization: Raw data is converted into standardized formats (e.g., CSV, JSON, RDF) using predefined schemas (as the official SDG indicator framework defined by the UN Statistics Division [6]).
- Normalization: Variables such as geographic units, time periods, and measurement scales are normalized to ensure comparability across countries and regions.
- Validation: Data quality checks are implemented to flag missing values, outliers, or inconsistent units, helping maintain reliability and analytical integrity.

IRCAI is engaging domain experts for the different SDGs to explore the most relevant KPIs to monitor, the search terms in the ontology (discussed in the next section) and the outcomes from the analysis. The resulting datasets are thus not only clean and standardized (considering limitations of the data sources, including different types of bias analysed and exposed) but also structured in elasticSearch indices to support downstream AI applications acting over powerful Lucene queries through the native API. Surrounding the data layer is a robust metadata architecture that enables discoverability, semantic enrichment, and AI-readiness. The metadata design is informed by the FAIR data principles (Findable, Accessible, Interoperable, and Reusable) and includes the following key components: (i) Descriptive Metadata, including descriptive elements such as title, description, source organization, temporal coverage, geographic coverage, and associated SDG goals, enabling human and machine agents to easily understand the scope and purpose of each data index; (ii) Structural Metadata, specifying the internal structure of the dataset, such as data types, column definitions, units of measurement, and relationships between variables, facilitating data parsing and automatic preprocessing by text mining tools; (iii) Source Metadata, capturing information about the dataset's origin, transformation steps, update frequency, and quality assurance processes, ensuring transparency, reproducibility, and trustworthiness; and (iv) Semantic Metadata, leveraging ontologies and controlled vocabularies to provide machine-readable semantics, linking dataset elements to established knowledge graphs, enabling reasoning across data indices and automated alignment of conceptually related information (see figure 2).

Figure 2: Visualisation of the SDG distribution of the ingested OECD AI policies according to the SDG ontology built on Wikidata terms defined with SDG topic experts

To ensure accessibility and integration with external systems, the infrastructure exposes datasets and metadata through native RESTful APIs, allowing developers and researchers to query and retrieve relevant data programmatically, enabling use in dashboards, modeling pipelines, and decision-support systems. Furthermore, adherence to open data standards such as DCAT (Data Catalog Vocabulary) and JSON-LD (Linked Data) ensure that the infrastructure can interface with other open government data platforms, research data repositories, and semantic web services. The architecture is designed with scalability and modularity in mind, allowing new datasets to be integrated with minimal manual intervention. Through automated ingestion

pipelines and schema mapping tools, the infrastructure can accommodate additional data sources while preserving metadata integrity and interoperability. Governance mechanisms, including data quality audits and contributor guidelines, ensure the sustainability and reliability of the system over time.

To support the in-depth analysis and leverage the availability of multilingual text resources at Wikidata, we have developed a SDG ontology inspired by [7] based on terms that correspond to Wikipedia pages. Currently published in a CSV format on GitHub [8], it defines rows corresponding to SDG entities−such as goals−and maps them to Wikidata Q-IDs. Key columns include: Level (e.g., SDG Goal), Code (e.g., "1", "1.2", "1.2.1"), Wikidata Q-Identifier (e.g., Q23442, Q3048436, Q28146087), label (human-readable name), Description (concise textual summary), and related concepts (optional Q-IDs linking to domains like health, energy, gender equality) Each SDG Goal row includes its code and corresponding Wikidata ID. Targets (e.g. 1.2) are mapped to both their own Wikidata entity and an explicit parent Goal. Indicators (e.g. 13.2.1) reference the relevant Target and define unit, measurement scale, and description. Using the CSV mappings, the ontology is constructed so that:

- sdg:hasTarget links a goal entity to its targets
- sdg:hasIndicator links targets to indicators
- sdg:measuredIn aligns indicator measures to Wikidata units

Additional cross-concept links (sdg:relatedTo) connect indicators to external Q-IDs in domains such as "maternal health" or "clean water". During dataset ingestion, each column bearing an indicator code is annotated using the corresponding Wikidata Q-ID from the ontology, enabling dataset cataloging via sdg:indicator URIs, semantic filtering and query based on concept-level tagging, as well as automatic generation of metadata triples (e.g. linking dataset to indicators and units).

3 The Amazon Observatory and Other Pilots

The prominence of domains such as digital data processing and machine learning illustrates AI's multidimensional capacity to address complex challenges in resource allocation, public health systems, and environmental sustainability. Comparative analysis between global discourses and those specifically oriented toward the Brazilian Amazon—driven by the expertise and coordinated efforts of the MetAmazonia initiative—reveals a pronounced emphasis on environmental preservation, biodiversity monitoring, and climate resilience in the latter. This divergence indicates that AI's contributions to sustainable development are not uniform but instead conditioned by region-specific priorities, ecological constraints, and socio-technical contexts. These findings underscore the necessity of developing adaptive, context-aware AI frameworks capable of aligning with the heterogeneous demands of both urban and rural environments.

The Amazon Observatory delivers outcomes such as the MetAmazonia chatbot, a multidimensional open data platform, and accessible resources for students and researchers to advance knowledge and innovation in the region. The system will be the basis for the planned MetAmazonia Chatbot, leveraging these datasets within the broader SDG AI-agent development, aligned with open education principles and UNESCO collaboration. It aims to make knowledge resources directly useful for learners and professionals engaged with Amazonia and their communities. Table 1 shows the data feeding the system across a diversity of topics from news, science and policies, exposing concerns of the public opinion, the knowledge we hold on priority topics, and part of the regulatory landscape.

Table 1: Data ingested into the Amazon Observatory from worldwide news (indicating the language coverage), published AI-related scientific articles, and related legal and regulatory landscape

Concepts	2024 News (Lang. Coverage)	Science	Policy
Biodiversity	18083 (100)	44693	3628
Indigenous peoples	8070 (96)	2014	107
Bioeconomy	156 (16)	33	31
Carbon Credits	236 (26)	2127	133
Public Health	26454 (69)	42355	697
Amazon rainforest	3936 (87)	172	115

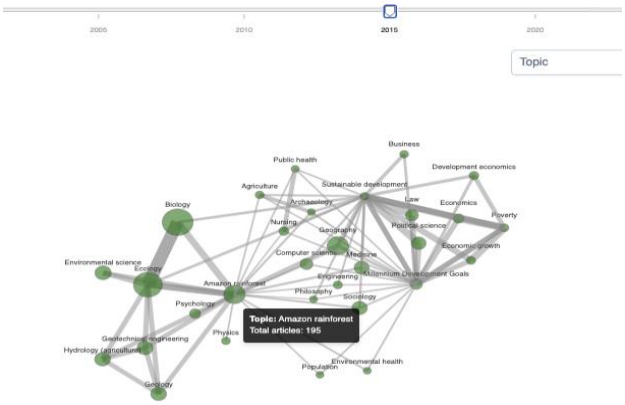


Figure 3: Evolution in time of the relation between research concepts related to the Brazilian Amazon Rainforest

To illustrate the potential of such approach, five initial modules have been developed and are being made available for COP30 activities in Belem, at the heart of Amazonia: (i) the News Stream with Sentiment provides multilingual coverage of Amazonia-related news, complemented by word clouds of main concepts and sentiment analysis visualized through maps and gauges; (ii) the Data Exploration Dashboard integrates multiple datasets, displaying global research trends, SDG policy coverage, and innovation activity; (iii) the relation between the concepts (edges) relevant to the Amazonia research and the interconnections between these concepts, being stronger or weaker according to the amount of published articles, where these are topics in common (see visualization in figure 3 and data characterization in table 1); (iv) in the Education view, the system visualizes open educational resources by mapping Amazonia-related topics to SDGs, highlighting key domains and their relevance to specific goals such as SDGs 11, 13, and 15; and (v) in regards to innovation ecosystems, we depict the

different initiatives that relate to priority topics in the Brazilian Amazon context and could help establishing international collaboration to address specific problems with local/global data.

Building on these modules, the SDG-oriented data infrastructure establishes a robust foundation for the development of an AI Agent specialized in Amazonia-related topics. By combining multilingual news streams, interconnected research concepts, and contextualized mappings of innovation and education, the system provides the necessary knowledge base and semantic structure to enable advanced reasoning, retrieval, and decision-support capabilities. Such an AI Agent is not only facilitate rapid access to diverse data sources but also support policymakers, researchers, and local communities by offering synthesized insights aligned with the SDGs. In doing so, it hopes to bridge global sustainability agendas with regional challenges, ensuring that context-specific solutions for the Amazon are informed by evidence, enriched by international collaboration, and continuously updated through the integration of real-time data.

4 Conclusions and Further Work

As the global community continues to pursue the 2030 Agenda, the importance of robust, interoperable, and machine-actionable SDG data infrastructure has never been greater. This paper has explored the architecture and implementation of an AI-ready data infrastructure for the SDGs, using the IRCAI SDG Observatory and its derived pilots as case studies. Central to this infrastructure is a well-defined metadata schema, semantic alignment with Wikidata entities, and adherence to FAIR data principles—all designed to support automation, reasoning, and integration of data across domains and geographies. By embedding SDG indicators, targets, and goals into a linked-data framework, the system transforms static reporting datasets into dynamic, queryable resources. This enables a wide range of AI applications, from natural language querying to knowledge graph reasoning and real-time decision support. The SDG Ontology—based on mappings to Wikidata Q-IDs—serves as a semantic backbone, enabling interoperability with external datasets and ontologies while enhancing transparency and reusability. Despite these advancements, several challenges remain. Data fragmentation across jurisdictions, lack of standardization in national reporting, and uneven metadata quality continue to hinder full automation and scalability. Furthermore, ethical considerations around data use—particularly in the context of AI-based decision-making—require further exploration.

To improve the Amazon Observatory, future development of AI-ready data infrastructure will focus on several key areas: (i) Automated Ontology Expansion. Leveraging large language models and knowledge extraction tools to automate the discovery and integration of new SDG-related concepts from policy documents, scientific literature, and real-time news streams; (ii) Interoperability with National Platforms. Building tools that support seamless integration of local statistical data with global and local SDG indicators (e.g., focusing Amazonia), using schema mapping and automated alignment with the SDG ontology; (iii) Real-Time Data Ingestion and Streaming Analytics. Incorporating real-time data sources, such as remote sensing, sensor networks, and social media, to enable early-warning systems and near-instant progress monitoring; (iv) AI-Powered Decision Support Tools. Developing interfaces and tools that allow policy-makers to simulate interventions, explore causal relationships, and evaluate trade-offs between SDG targets using AI models trained on the structured data; (v) Community Governance and Open Collaboration. Establishing open, participatory governance models for ontology evolution, dataset curation, and quality assurance to ensure that the infrastructure remains globally relevant and inclusive.

In conclusion, AI-ready SDG infrastructure represents a transformative opportunity for evidence-based policy, global collaboration, and data-driven action on sustainability. By continuing to invest in semantic technologies, metadata standards, and open data ecosystems, we can enable a new generation of intelligent tools that accelerate progress toward the SDGs globally but also locally.

Acknowledgments / Zahvala

We thank the support of the European Commission projects ELIAS (GA101120237) and RAIDO (GA101135800).

References / Literatura

[1] Bachmann, N., Tripathi, S., Brunner, M. and Jodlbauer, H.(2022). *The contribution of data-driven technologies in achieving the sustainable development goals*. Sustainability, 14(5), p.2497.

[2] Stahl, B.C., Schroeder, D. and Rodrigues, R., 2022. AI for Good and the SDGs. In *Ethics of artificial intelligence: Case studies and Options for addressing ethical challenges* (pp. 95-106). Cham: Springer International Publishing.

[3] Pita Costa, J., (2023) *Water Intelligence to Support Decision Making*, Operation Management and Water Education: NAIADES Report. IRCAI.

[4] Pita Costa J., Barrionuevo L.. Kovič Dine M. (2025) Observing the Impact of AI in the Progress of Sustainable Development Goal 11. In *Proceedings of the 23rd IADIS International Conference e-Society 2025*

[5] Mitja Jermol, Joao Pita Costa and Matej Kovačič (2025) *Onwards to an Ethical and Bias Aware Education for Sustainability through AI*. Journal of Artificial Intelligence for Sustainable Development (to appear)

[6] Sustainable Development Solutions Network(2015) Indicators and a Monitoring Framework for the SDGs. United Nations.

[7] Joshi, A., Gonzalez Morales, L., Klarman, S., Stellato, A., Helton, A., & Lovell, S. (2019). *A Knowledge Organization System for the United Nations Sustainable Development Goals*. Proceedings of the 2019 International Conference on Knowledge Engineering and Knowledge Management (EKAW). Springer.

[8] Pita Costa, J. (2025) *IRCAI SDG Ontology*. GitHub. Available at https://github.com/IRCAI-SDGobservatory/data

Towards a Format for Describing Networks

Vladimir Batagelj
IMFM
Ljubljana, Slovenia
UP, IAM and FAMNIT
Koper, Slovenia
UL, FMF
Ljubljana, Slovenia
vladimir.batagelj@fmf.uni-lj.si

Tomaž Pisanski
UP, FAMNIT
Koper, Slovenia
IMFM
Ljubljana, Slovenia
tomaz.pisanski@upr.si

Iztok Savnik
UP, FAMNIT
Koper, Slovenia
iztok.savnik@upr.si

Ana Slavec
UP, FAMNIT
Koper, Slovenia
UP, IAM InnoRenew CoE
Koper, Slovenia
ana.slavec@famnit.upr.si

Nino Bašić
UP, FAMNIT
Koper, Slovenia
IMFM
Ljubljana, Slovenia
nino.basic@famnit.upr.si

Abstract

The article provides an overview of the most important network analysis resources and the various types of networks encountered in their use. Based on experience in developing the NetsJSON format, we present components that an exchange/archive format for describing networks should contain.

Keywords

Network analysis, Network types, Identification, Format, Exchange, Archive, Data repository, Factorization, JSON, FAIR.

1 Introduction

Open data plays a crucial role in ensuring the computational reproducibility and verifiability of published results. The obtained results can be verified or supplemented with other methods. Collections of similar and well-documented datasets are also crucial for developing new methods to analyze specific types of data. It is good to test a new method on several datasets and check whether it gives meaningful/expected results. When preparing such data, it is essential to adhere to the FAIR principles – Findability, Accessibility, Interoperability, and Reusability. To facilitate ease of use, data should ideally be stored in a text format that preserves the structure of the data and includes relevant metadata. Datasets are alive. Their connection to open repositories is important for their accessibility and maintenance.

In 2023, the International Network for Social Network Analysis (INSNA) requested that Zachary Neal form a working group to develop **recommendations for sharing network data and materials**. They were published in *Network Science* in 2024 [21] accompanied by the *Endorsement page* [20].

Network analysis is an area where data is often stored in diverse file formats. It would be highly beneficial to adopt a common "archiving/exchange" format that can describe (almost) all networks and support authoring, deposit, exchange, visualization, reuse, and preservation of network data. Such a format would allow us to obtain the specific descriptions required by various network analysis programs using relatively simple scripts.

We have many years of experience in developing formats for describing graphs and networks [11, 10, 4]. We will present the NetsJSON format currently used to describe networks with structured data, and some ideas for improving it. This could be a starting point for the development of a common format for exchanging and archiving networks.

2 Support for network analysis

The concept of a network is an extension of the concept of a graph. A graph describes the structure of a network. Network analysis is a branch of data analysis that draws heavily on the concepts and results of graph theory. The difference between the two is that networks are usually "irregular", while most problems and results of graph theory assume some "regularity".

There are many tools and programs for network analysis. For example, UCINET, Pajek, Gephi, NetMiner, Cytoscape, NodeXL, E-Net, Tulip, PUCK, GraphViz, SocNetV, Kumu, Polinode, etc. Programmers can use network analysis packages/libraries in a variety of programming languages (Python, R, Julia, C++, etc.).

They are supporting various network description formats: CSV, UCINET DL, Pajek NET, Gephi GEXF, GDF, GML, GraphML, GraphX, GraphViz DOT, Tulip TPL, Netdraw VNA, Spreadsheet, etc. [13, 25, 16].

In addition, network data appears in several application areas such as chemistry and genealogy. There are many formats for describing these data.

Network datasets are available in multiple repositories. Some repositories only provide metadata about an individual network and a link to the actual dataset. At the same time, others also store the data itself and offer users a display of basic network characteristics. None of them explicitly adheres to FAIR data principles.

Interesting networks can also be found on general data repositories such as Kaggle. Networks can also be created programmatically from selected data. For example, from bibliographic data from the free OpenAlex service, we can create collections of bibliographic networks on a selected topic using the OpenAlex2Pajek library in R.

For detailed lists of network analysis resources with links to web pages, see GitHub/bavla/NetsJSON/Info [4].

Information Society 2025, 6–10 October 2025, Ljubljana, Slovenia

https://doi.org/10.70314/is.2025.sikdd.9

3 Graphs and networks

3.1 Unit identification

The fundamental task in transforming data about the selected topic into a structured dataset to be used in further analyses is the *identification of units* (entity recognition). Often, the source data are available as unstructured or semi-structured text. In this case, the transformation is a task of the *computer-assisted text analysis* (CaTA). *Terms* considered in TA are collected in a *dictionary* (it can be fixed in advance, or built dynamically). The two main problems with terms are: *equivalence* – different words representing the same term, and *ambiguity* – same words representing different terms. Because of these, the *coding* – the transformation of raw text data into formal *description* – is often done manually or semiautomatically.

We assume that unit identification assigns a unique identifier (ID) to each unit. For some types of units, such IDs are standardized: ISO 3166-1 alpha-2 two-letter country codes, ISO 9362 Bank Identifier Codes (BIC), ORCID – Open Researcher and Contributor ID, ISSN – International Standard Serial Number, DOI – Digital Object Identifier, URI – Uniform Resource Identifier, etc.

Often, in data displays, IDs are replaced by corresponding (short) labels/names.

Besides the semantic units or *concepts* related to the selected topic, we can also identify in the raw data syntactic units – parts of the text. As *syntactic units* of TA, we usually consider clauses, statements, paragraphs, news, messages, etc.

In thematic TA, the units are coded as a rectangular matrix *Syntactic units* × *Concepts* which can be considered as a two-mode network.

In semantic TA the units (often clauses) are encoded according to the S-V-O (*Subject-Verb-Object*) model or its improvements. This coding can be directly considered as network with *Subjects* ∪ *Objects* as nodes and links (arcs) labeled with *Verbs*. This is also a basis for the semantic web and knowledge networks.

3.2 Networks

A *network* is based on two sets – a set of *nodes* (vertices) that represent the selected *units*, and a set of *links* (lines) that represent *ties* between units. They determine a *graph*.

Additional data about nodes or links can be known – their *properties* (attributes). For example: name/label, type, value, etc.

Network = Graph + Data

The data can be measured or computed.

Formally, a *network* $\mathcal{N} = (\mathcal{V}, \mathcal{L}, \mathcal{P}, \mathcal{W})$ consists of:

- a *graph* $\mathcal{G} = (\mathcal{V}, \mathcal{L})$, where $\mathcal{V}$ is the set of nodes and $\mathcal{L} = \mathcal{E} \cup \mathcal{A}$ is the set of links. A link $e \in \mathcal{L}$ is either directed – an arc $e \in \mathcal{A}$, or undirected – an edge $e \in \mathcal{E}$, $n = |\mathcal{V}|$, $m = |\mathcal{L}|$
- $\mathcal{P}$ is a set of *node value functions* / properties: $p : \mathcal{V} \to A$
- $\mathcal{W}$ is a set of *link value functions* / weights: $w : \mathcal{L} \to B$

Sometimes, implicit additional information/data about values is provided in the specifications of properties: (a) how can we compute with values – algebraic structures, semigroup, monoid, group, semiring, etc., and (b) properties of values – in a molecular graph, an atom is assigned to each node; properties of relevant atoms are such additional data.

The terminology in the field of network analysis is not unified. Different application areas use other terms. For example: node – vertex, actor, unit; link – line, tie, edge, connection; etc.

3.3 Types of networks

Besides ordinary (directed, undirected, mixed) networks, some special types of networks are also used:

- *2-mode networks*, bipartite (valued) graphs – networks between two disjoint sets of nodes.
- *multi-relational networks*.
- *linked networks* and *collections of networks*.
- *multilevel networks*.
- *temporal networks*, dynamic graphs – networks changing over time.
- specialized networks: representation of genealogies as *p-graphs*; *Petri's nets*, molecular graphs, etc.

Network (input) file formats should provide the means of expressing all of these types of networks. All interesting data should be recorded (respecting privacy).

In a *two-mode* network $\mathcal{N} = ((\mathcal{U}, \mathcal{V}), \mathcal{L}, \mathcal{P}, \mathcal{W})$ the set of nodes consists of two disjoint sets of nodes $\mathcal{U}$ and $\mathcal{V}$, and all the links from $\mathcal{L}$ have one end node in $\mathcal{U}$ and the other in $\mathcal{V}$.

A *multi-relational network* $\mathcal{N} = (\mathcal{V}, (\mathcal{L}_1, \mathcal{L}_2, \ldots, \mathcal{L}_k), \mathcal{P}, \mathcal{W})$ contains different relations $\mathcal{L}_i$ (sets of links) over the same set of nodes. Also, the weights from $\mathcal{W}$ are defined on different relations or their union.

In a *linked* or *multimodal* network $\mathcal{N} = ((\mathcal{V}_1, \mathcal{V}_2, \ldots, \mathcal{V}_j), (\mathcal{L}_1, \mathcal{L}_2, \ldots, \mathcal{L}_k), \mathcal{P}, \mathcal{W})$ the set of nodes $\mathcal{V}$ is partitioned into subsets (*modes*) $\mathcal{V}_i$, $\mathcal{L}_s \subseteq \mathcal{V}_p \times \mathcal{V}_q$, and properties and weights are usually partial functions.

A set of networks $\{\mathcal{N}_1, \mathcal{N}_2, \ldots, \mathcal{N}_k\}$ in which each network shares a (sub)set of nodes with some other network is called a *collection* of networks.

A linked network can be transformed into a collection of networks and vice versa.

Bibliographical information is usually represented as a collection of bibliographical networks {**Cite**, **WA**, **WK**, **WC**, **WI**, ...} (W – works, A – authors, K – keywords, C – countries, I – institutions) [7].

Another example of multimodal multirelational networks are knowledge graphs. They can have a very diverse structure (a large number of types of units (modes) and predicates (relations)), which allows for a fairly accurate description of facts from a selected field and solving problems about it. Network analysis methods are particularly useful in analyzing one or a few relational (sub)networks of a knowledge graph.

In a *temporal network*, the presence/activity of a node/link can change through time $\mathcal{T}$. The basic division of temporal network descriptions is into cross-sectional and longitudinal. A cross-sectional description usually consists of a sequence of time slices – ordinary networks that describe the state at a selected moment or time interval. A longitudinal description is based on *temporal quantities* [12, 9] or on a sequence of events.

4 Description of traditional networks

How to describe a network $\mathcal{N} = (\mathcal{V}, \mathcal{L}, \mathcal{P}, \mathcal{W})$? In principle the answer is simple – we list its components $\mathcal{V}$, $\mathcal{L}$, $\mathcal{P}$, and $\mathcal{W}$.

The simplest way is to describe a network $\mathcal{N}$ by providing $(\mathcal{V}, \mathcal{P})$ and $(\mathcal{L}, \mathcal{W})$ in a form of two tables. Both tables are often maintained in some spreadsheet program. They can be exported as text in CSV (Comma Separated Values) format. In large networks, we split a network into some subnetworks – a collection, to avoid the empty cells.

To save space and improve computing efficiency, we often replace values of categorical variables with integers. In R, this

encoding is called a *factorization*. We enumerate all possible values of a given categorical variable (coding table) and afterward replace each value with the corresponding index in the coding table. Since node labels/IDs can be considered a categorical variable, factorization is also usually applied to them. In data analysis, indices start with 1, but real computer scientists start counting from 0. Therefore, it is desirable to include information about the minimal index value in the description.

This approach is used in most programs dealing with large networks. Unfortunately, the coding table is often considered as a kind of metadata and is omitted from the description.

In Pajek [15], node property can be represented in the associated file as a vector (numbers, **.vec**), a partition (nominal, **.clu**), or a permutation (order, **.per**). All network files can be combined into a single project file (**.paj**). Metadata can be added as comments written on lines starting with the **%**. An example of transforming CSV tables into Pajek files is available at GitHub/bavla/netsJSON/example/bib [4].

5 Nets and NetsJSON

We were satisfied with the ”traditional” network description, as implemented in Pajek [15], until we became interested in networks with node/link properties that are not measured in standard scales (ratio, interval, ordinal, nominal), but have structured values (text, subset, interval, distribution, time series, temporal quantity, function, etc.). In topological graph theory, an embedding is described by assigning a rotation to each node [23]. For describing temporal networks, we initially extended the Pajek format, defined and used the Ianus format [12].

For a format supporting structured values, there were two obvious choices for its base – XML and JSON. They are both widely known and suitable as structured data formats. However, JSON can represent the same data as XML more concisely. We chose JSON and in 2015 started developing and using the NetJSON format and the Nets Python package to handle networks with structured-valued properties or weights [5, 4, 3]. On February 26, 2019, the format was renamed to NetsJSON because of the collision with http://netjson.org/rfc.html. NetsJSON has two versions: a *basic* and a *general* version. The current implementation of the Nets library supports only the basic version.

In addition to describing networks with structured values, NetsJSON is expected to offer the capabilities of (most) existing network description formats [13, 25] (archiving, conversion) and provide input data for D3.js visualizations.

A network description in NetsJSON follows the JSON (JavaScript) syntax and consists of five main fields (`netsJSON`, `info`, `nodes`, `links`, `data`).

```
{
"netsJSON": "basic",
"info": {
   "org":1, "nNodes":n, "nArcs":mA, "nEdges":mE,
   "simple":TF, "directed":TF, "multirel":TF, "mode":m,
   "network":fName, "title":title,
   "time": { "Tmin":tm, "Tmax":tM, "Tlabs": {labs} },
   "meta": [events], ...
   },
"nodes": [
   { "id":nodeId, "lab":label, "x":x, "y":y, ... },
   ***
   ],
"links": [
   { "type":arc/edge, "n1":nodeID1, "n2":nodeID2, "rel":r, ... },
   ***
   ],
"data": {
   "data1":description1,
   ***
   }
}
```

where … are user-defined properties and *** is a sequence of such elements.

The `netsJSON` field identifies the format, the `info` field contains metadata, the `nodes` field contains a table $(\mathcal{V}, \mathcal{P})$ and the `links` field contains a table $(\mathcal{L}, \mathcal{W})$. In recent years, we also analyzed bike systems (link weight is a daily number of trips distribution), bibliographies (yearly distributions of publications or citations), and multiway networks [8, 9, 1]. It turned out that it was necessary to add another main field, `data`, to the basic NetsJSON format, in which we provide additional data about the properties of values (translations of labels in selected languages, algebraic structure, etc. [6]).

An event description can contain the following fields: type, date, title, author, desc, url, cite, copyright, etc. It is intended to provide information about the "life" of the dataset – collection/creation, changes, releases, uses, publications, etc.

For describing temporal networks, a node element and a link element have an additional required property `tq` – a temporal quantity. For example, see `violenceU.json` at `GitHub/bavla/Graph/JSON` describing the Franzosi's violence network.

The general NetsJSON format is also expected to support the description of network collections.

6 Elements of a common network format

Our experience with network analysis to date is summarized in the following recommendations on the elements of a common format for describing networks.

Combining data and its metadata into a single file is a robust approach for ensuring data integrity. A JSON-based format is particularly well-suited for this purpose, as it fits well with the data structures of modern programming languages. JSON also supports Unicode.

We would also encourage the provision, as metadata, of information about the context of the network, additional knowledge about it, articles or notebooks on its analysis, comments of users, etc. Kaggle is a good example. An improved ICON repository or Network Repository (we disagree with their "citation request") could be the way to go. Existing metadata standards should be taken into account (Dublin Core, FAIR, Schema). Data has a "life". When selecting data, its age is often important. Metadata should include at least the collection/creation date and the last modification date.

By FAIR principles, the format should support: **F**indability: Globally unique and persistent identifier, rich metadata. **A**ccessibility: Open, free, and universally implementable standardized communication protocol. **I**nteroperability: Formal, accessible, shared, and proudly applicable language for knowledge representations. **R**eusable: Metadata are richly described and associated with detailed provenance.

The format must support all types of networks (simple, 2-mode, linked, multi-relational, multi-level, temporal). The network can contain both arcs and edges, as well as parallel links. To describe some knowledge graphs, it would be necessary to allow other links to act as the end nodes of a link [18].

As mentioned earlier, using factorization produces a more concise description of the network. In cases where the node names are not too long and are readable, we sometimes want to avoid factorization. This can be achieved by using a switch that indicates whether factorization is used. We can also shorten the description length by introducing default values of selected

properties. If we also allow counting from 0, it makes sense to add information about the smallest index.

Long labels cause problems when printing/visualizing (parts of) networks and results. Therefore, it is useful to have abbreviated versions of labels available. For language-based labels, it is sometimes useful to offer additional versions in selected other languages, which increases the accessibility of the data and the understandability of the results.

Most of the network datasets produced by network science have no node labels. Node labels are not needed if you study distributions, but they are essential in the interpretation of the obtained "important substructures". We would encourage providing node labels, or at least some typological information, in cases where privacy issues arise.

The common format should support descriptions of networks specific to specialized fields of application, such as molecular graphs, genealogies (p-graphs) [29], and topological graph embeddings [23, 24], among others. The format must be extensible. In addition to the agreed-upon fields, the users can add their own, allowing for a comprehensive description of their data.

It is also interesting to ask whether and, if so, how to include descriptions of its displays in the network description. Perhaps it would be worth relying on VEGA-lite [26, 28] and D3.js [14]. Some ideas can also be taken from the section on "defining visualization parameters in the input file" in the Pajek manual / 5.3 [19, p. 89].

Although we are committed to a single-file approach, there may be times when external files are needed (for example, images to display nodes). Consideration should be given to how to support this option. Given the basic purpose of the common format, standard tools (ZIP) can be used to compress large networks.

We have not yet started working on a general format. It is supposed to enable descriptions of collections of networks. The question arises about the scope of validity of IDs – does the same ID in different networks represent the same or other units? This is important for operations such as the union or intersection of networks. Which way to go – introducing contexts or using matchings? Maybe some ideas from the Open Archives Initiative Object Reuse and Exchange (OAI-ORE) and GraphX could be used [22, 17]. An interesting option is the constructive network description – building a network from smaller components [10] or describing a network by its construction sequence [2].

Additional ideas may be found on the page "*A Python Graph API?*" [27]. For now, we would leave aside descriptions of generalizations of networks (multiway networks and hypernets), but we must not forget about them.

The agreed format must be well documented and supported by examples of the use of supported options.

7 Conclusions

Yet another format only makes sense as a project of a larger community of users in the field of network analysis.

Acknowledgements

The computational work reported in this paper was performed using programs R and Pajek [15]. The code and data are available at Github/Bavla [4].

V. Batagelj is supported in part by the Slovenian Research Agency (research program P1-0294 and research project J5-4596) and prepared within the framework of the COST action CA21163 (HiTEc).

T. Pisanski is supported in part by the Slovenian Research Agency (research program P1-0294 and research projects BI-HR/23-24-012, J1-4351, and J5-4596).

N. Bašić is supported in part by the Slovenian Research Agency (research program P1-0294 and research project J5-4596).

References

[1] Vladimir Batagelj. 2024. Cores in multiway networks. *Social Network Analysis and Mining*, 14, 1, 122.

[2] Vladimir Batagelj. 1985. Inductive classes of graphs. In *Proceedings of the 6th Yugoslav Seminar on Graph Theory*, 43–56.

[3] Vladimir Batagelj. 2016. Nets – Python package for network analysis. accessed: 2025-03-18. (2016). https://github.com/bavla/Nets.

[4] Vladimir Batagelj. 2016. NetsJSON – a JSON format for network analysis. accessed: 2025-03-18. (2016). https://github.com/bavla/netsJSON.

[5] Vladimir Batagelj. 2016. Network visualization based on JSON and D3.js. slides. (2016). https://github.com/bavla/netsJSON/blob/master/doc/netVis.pdf.

[6] Vladimir Batagelj. 2021. Semirings in network data analysis / an overview. slides. (2021). https://github.com/bavla/semirings/blob/master/docs/semirings.pdf.

[7] Vladimir Batagelj and Monika Cerinšek. 2013. On bibliographic networks. *Scientometrics*, 96, 3, 845–864.

[8] Vladimir Batagelj and Anuška Ferligoj. 2016. Symbolic network analysis of bike sharing data / Citi Bike. accessed: 2025-03-18. (2016). https://github.com/bavla/Bikes/blob/master/bikes.pdf.

[9] Vladimir Batagelj and Daria Maltseva. 2020. Temporal bibliographic networks. *Journal of Informetrics*, 14, 1, 101006.

[10] Vladimir Batagelj and Andrej Mrvar. 1995. NetML. accessed: 2025-03-18. (1995). https://github.com/bavla/netsJSON/blob/master/doc/snetml.pdf.

[11] Vladimir Batagelj and Andrej Mrvar. 2018. Pajek and pajekxxl. In *Encyclopedia of Social Network Analysis and Mining*. Springer, 1–13.

[12] Vladimir Batagelj and Selena Praprotnik. 2016. An algebraic approach to temporal network analysis based on temporal quantities. *Social Network Analysis and Mining*, 6, 1–22.

[13] Jernej Bodlaj and Monika Cerinšek. 2014, 2017. Network data file formats. In *Encyclopedia of Social Network Analysis and Mining*. Reda Alhajj and Jon Rokne, editors. Springer New York, New York, NY, 1076–1091. ISBN: 978-1-4614-7163-9. DOI: 10.1007/978-1-4614-7163-9_298-1.

[14] Bostock, Mike and Davies, Jason and Heer, Jeffrey and Ogievetsky, Vadim. [n. d.] The JavaScript library for bespoke data visualization. accessed: 2025-08-29. (). https://d3js.org/.

[15] Wouter De Nooy, Andrej Mrvar, and Vladimir Batagelj. 2018. *Exploratory social network analysis with Pajek: Revised and expanded edition for updated software*. Vol. 46. Cambridge university press.

[16] Gephi. 2022. Supported graph formats. accessed: 2025-03-18. (2022). https://gephi.org/users/supported-graph-formats/.

[17] Joseph E Gonzalez, Reynold S Xin, Ankur Dave, Daniel Crankshaw, Michael J Franklin, and Ion Stoica. 2014. {Graphx}: graph processing in a distributed dataflow framework. In *11th USENIX symposium on operating systems design and implementation (OSDI 14)*, 599–613.

[18] Aidan Hogan et al. 2021. Knowledge graphs. *ACM Computing Surveys (Csur)*, 54, 4, 1–37.

[19] Andrej Mrvar and Vladimir Batagelj. 2025. Pajek reference manual. *Version 6.01*. http://mrvar.fdv.uni-lj.si/pajek/pajekman.pdf.

[20] Zachary P Neal and et al. 2024. Recommendations for sharing network data and materials – endorsement page. accessed: 2025-03-18. (2024). https://www.zacharyneal.com/datasharing.

[21] Zachary P Neal et al. 2024. Recommendations for sharing network data and materials. *Network Science*, 12, 4, 404–417.

[22] Open Archives Initiative. 2014. Object Reuse and Exchange (OAI-ORE). 2014-08-14. accessed: 2025-03-18. (2014). https://www.openarchives.org/ore/.

[23] Tomaž Pisanski. 1980. Genus of cartesian products of regular bipartite graphs. *Journal of Graph Theory*, 4, 1, 31–42.

[24] Tomaž Pisanski and Arjana Žitnik. 2004. Representations of graphs and maps. In *26th International Conference on Information Technology Interfaces, 2004.* IEEE, 19–25.

[25] Matthew Roughan and Jonathan Tuke. 2015. Unravelling graph-exchange file formats. *arXiv preprint arXiv:1503.02781*.

[26] Arvind Satyanarayan, Dominik Moritz, Kanit Wongsuphasawat, and Jeffrey Heer. 2016. Vega-lite: a grammar of interactive graphics. *IEEE transactions on visualization and computer graphics*, 23, 1, 341–350.

[27] The Python Wiki. 2011. A Python Graph API? accessed: 2025-03-18. (2011). https://wiki.python.org/moin/PythonGraphApi.

[28] University of Washington Interactive Data Lab. [n. d.] Vega-Lite – A Grammar of Interactive Graphics. accessed: 2025-08-29. (). https://vega.github.io/vega-lite/.

[29] Douglas R White, Vladimir Batagelj, and Andrej Mrvar. 1999. Anthropology: analyzing large kinship and marriage networks with pgraph and pajek. *Social Science Computer Review*, 17, 3, 245–274.

Automating Numba Optimization with Large Language Models: A Case Study on Mutual Information

Lučka Kozamernik
Teads
lucka.kozamernik@teads.com

Blaž Škrlj
Teads
blaz.skrlj@teads.com

Martin Jakomin
Teads
martin.jakomin@teads.com

Jasna Urbančič
Teads
jasna.urbancic@teads.com

Abstract

Contemporary large language models (LLMs) enable fast research cycles when developing or optimizing new algorithms. In this work, we investigate whether existing LLMs are sufficient to automatically, under constraints of unit tests, produce implementations of computational extensive algorithms such as the mutual information algorithm that would out-perform existing human-made baselines. We establish an evaluation pipeline where new proposed AI implementations are rigorously tested, evaluated, and benchmarked against existing baselines. We used synthetic numeric datasets of different sizes and results show 10-times speed-up using LLM optimized implementations compared to the naive Numba-based optimization while producing consistently correct mutual information scores.

Keywords

optimization, mutual information, LLM, Numba

1 Introduction

Mutual Information (MI) (detailed overview in, e.g., [4]) stands as a fundamental measure in information theory, quantifying the statistical dependency between two random variables. Its application is widespread and critical across numerous domains, including feature selection in machine learning [8], neuroscience for analyzing neural spike trains [2], and bioinformatics for understanding gene regulatory networks [9]. The versatility of MI lies in its ability to capture arbitrary non-linear relationships, a significant advantage over linear correlation measures like Pearson's coefficient.

However, the computational cost of calculating mutual information, especially for large datasets with continuous variables, presents a substantial bottleneck. The standard approach involves discretizing the data into bins in order to estimate probability distributions, a process whose accuracy and performance are highly sensitive to the chosen binning strategy and the efficiency of the underlying implementation. For practitioners working within the Python ecosystem, libraries like NumPy and SciPy are standard tools, but their performance on MI calculations can be suboptimal for high-throughput screening or large-scale data exploration tasks.

To address this performance gap, Just-In-Time (JIT) compilers like Numba [6] have become indispensable. By translating Python and NumPy code into optimized machine code at runtime, Numba offers C-like performance without sacrificing the flexibility and ease of use of the Python language. A well-written, Numba-accelerated MI function can be orders of magnitude faster than its pure Python equivalent. Despite these gains, achieving optimal performance with Numba is not always straightforward. The efficiency of *Numba-jitted* code is highly dependent on the specific implementation patterns, data access methods, and loop structures used—subtleties that often require significant programmer expertise to navigate.

This paper introduces a novel approach to bridge this gap: the use of Large Language Models (LLMs) to automatically optimize Numba-based mutual information algorithms. We hypothesize that modern LLMs, trained on vast repositories of code, possess the capability to analyze suboptimal Numba implementations and refactor them into more efficient versions. Our work explores whether an LLM can identify and correct common performance anti-patterns in Numba code, such as improper loop organization or inefficient data type usage, to generate an MI implementation that surpasses a naively written Numba function. We present a framework for systematically prompting an LLM with a baseline algorithm and evaluating the performance of its generated optimizations, demonstrating the potential for AI-driven code acceleration in scientific computing.

2 Related work

This research builds upon three principal areas of study: the computation of mutual information, performance optimization with JIT compilers, and the application of Large Language Models to code intelligence tasks.

Mutual Information estimation is the long-standing challenge of accurately and efficiently estimating mutual information from given data. Defined as

$$I(X;Y) = \mathbb{E}_{p(X,Y)}\left[\log\left(\frac{p(X,Y)}{p(X)p(Y)}\right)\right],$$

it measures the pairwise relationships between random variables (continuous or discrete). The most common methods, as reviewed by Fraser and Swinney (1986) [3] and explored in detail by Kraskov, Stögbauer, and Grassberger (2004) [5], are based on data discretization (binning) or k-nearest neighbors (k-NN) estimators. While k-NN methods avoid the issue of bin selection, they typically incur higher computational complexity. Binned methods, though conceptually simpler, depend heavily on the binning strategy for accuracy and performance, a topic extensively studied by Steuer et al. (2002) [11]. Our work focuses on the binned approach, as it is highly amenable to loop-based array computations where Numba excels.

Information Society 2025, Ljubljana, Slovenia

https://doi.org/https://doi.org/10.70314/is.2025.sikdd.22

The performance limitations of Python for numerical computation led to the development of various acceleration tools, specifically JIT compilers for Scientific Python. Numba, introduced by Lam, Pitrou, and Seibert in 2015 [6], has emerged as a leading solution by providing a decorator-based JIT compiler that integrates seamlessly with NumPy. It allows developers to accelerate functions containing Python and NumPy syntax, often achieving performance comparable to compiled languages. Research and community best practices have established a set of optimization techniques for Numba, such as managing memory layout, ensuring type stability, and structuring loops for parallelization and vectorization. This body of knowledge forms the basis against which we evaluate the LLM's optimization capabilities. Our work differs from traditional performance tuning by attempting to automate the discovery and application of these techniques solely through an AI model.

The emergence of robust Large Language Models (LLMs), such as OpenAI's Codex (the technology powering GitHub Copilot), has revolutionized software development. These models have demonstrated remarkable proficiency in code generation, translation, and explanation [1]. More recently, research has shifted towards their application in more nuanced tasks like code refactoring and optimization. For instance, studies have explored using LLMs to suggest improvements for energy efficiency or to refactor code for better readability. However, the specific domain of optimizing numerical algorithms within a JIT compilation framework like Numba remains relatively unexplored. While LLMs are known to generate functional code, their ability to produce code that is performant by adhering to the specific constraints and best practices of a framework like Numba is an open and compelling research question that this paper directly addresses.

3 Using LLMs to optimize existing code

To facilitate systematic experimentation with LLM-optimized code, we set up a novel framework. The workflow consists of the following basic steps:

(1) Prompt the LLM with the task and context.
(2) Test the proposed optimizations against the unit tests.
(3) Benchmark the proposed implementation.

The framework is LLM-agnostic, meaning that any LLM can be used with it. We opt for the latest and most advanced versions of two popular LLMs, namely ChatGPT 5 and Gemini 2.5-Pro. Both are freely available and excel in complex tasks such as reasoning and coding. The architecture of the framework is given in Figure 1.

To ensure a fair comparison between the models, both evaluated LLMs received the same prompt and the same context. The prompt was *"Can you make this code computationally more efficient, this meaning it computes faster?"*, while the context included the code that needed to be optimized. The initial code used in the input already contained some Numba instructions, however those were basic and naive. The tested code is a part of OutRank, an open-source tool for computing cardinality-aware feature ranking [10] and encompasses an implementation of the mutual information estimation.

The LLM output was first tested on unit tests to ensure that the optimizations still produced valid code and did not change any functionalities. By testing the proposed solution before using it for benchmarking, we are guaranteed that the code and its output are correct, consistent, and stable. Although not part of the framework at this stage, the output of the unit tests could serve as additional prompts to the LLM in order to improve itself and the code on the areas where the tests are failing.

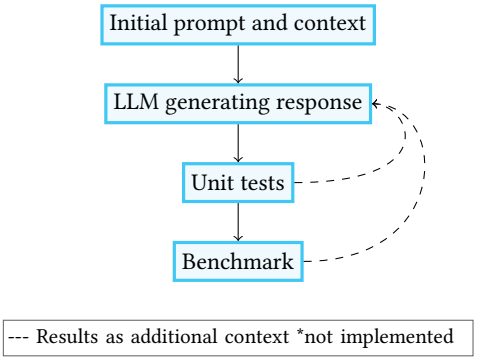


Figure 1: Architectural sketch of the benchmarking framework. Dashed are the feedback loops proposed as future work.

Finally, in the last step of the framework, the resulted implementations were extensively benchmarked. The metric we were most interested in was the time needed to compute the mutual information for a given dataset; however, other metrics, such as memory utilization or GPU utilization, could also be used for a different use case. We further discuss our experimental setup in the results section.

3.1 Reviewing the LLM optimized code

The implementations of mutual information, produced by the selected LLMs, are remarkably similar — both in syntax and in the naming convention. However, there are subtle differences that set them apart, which we will address later. AI-aided implementations have in common that they completely omit error-handling model inherited from NumPy opting for the native Python instead. Moreover, they disregard bound checks for matrix operations beforehand, leaving the code to crash if it goes out of bounds. The latter is, according to the official documentation, advised for debug purposes only and should be turned off for production, as it slows down the code significantly. Having said that, Gemini implemented bound checks using elementary operations. In line with the change in error handling, both implementations prefer elementary operations over native NumPy functions. For example, to find the maximal value in an array, the LLM optimized code goes through all elements in the array by the index and compares to the current maximum instead of calling the built-in NumPy function. There is more evidence for this preference in the code. Such changes make the code appear much more C-like than native Python. Whenever there is the need for typecasting, the optimized code performs it at definition, instead of on return, which is commonly used in the naive implementation. The two types of proposed changes are illustrated with the code samples in Figure 2. Lastly, both LLMs introduced additional function that performs the pre-built grouping to avoid unnecessary allocations and relocations in the loop. While the core techniques used for optimization are the same for both LLMs, Gemini 2.5-Pro used Numba's prange in one of the main computational loops, which adds parallelization, and makes the implementation faster on multicore machines. It also took the use of elementary operations much further than ChatGPT 5 — it replaced nearly all NumPy operations with native operations, increasing the row count twice as much as ChatGPT 5 did. The numbers are reported in Table 1

```python
@njit(
    'Tuple((int32[:], int32[:]))(int32[:])',
    cache=True,
    fastmath=True,
    error_model='numpy',
    boundscheck=True,
)
def numba_unique(a):
    """Identify unique elements in an array, fast"""

    container = np.zeros(np.max(a) + 1, dtype=np.int32)
    for val in a:
        container[val] += 1

    unique_values = np.nonzero(container)[0]
    unique_counts = container[unique_values]
    return unique_values.astype(np.int32),
          unique_counts.astype(np.int32)
```

```python
@njit('Tuple((int32[:], int32[:]))(int32[:])', cache=True,
      fastmath=True)
def numba_unique(a):
    # assumes a >= 0
    maxv = 0
    for i in range(a.size):
        if a[i] > maxv:
            maxv = a[i]
    container = np.zeros(maxv + 1, dtype=np.int32)
    for i in range(a.size):
        container[a[i]] += 1
    unique_values = np.nonzero(container)[0].astype(np.int32)
    unique_counts = container[unique_values].astype(np.int32)
    return unique_values, unique_counts
```

```python
@njit('Tuple((int32[:], int32[:]))(int32[:])', cache=True,
      fastmath=True)
def numba_unique(a):
    """
    Identify unique elements and their counts in a non-negative
          integer array.
    This version finds the max value in one pass to size the
          container.
    """
    # Assumes a >= 0
    maxv = 0
    if a.size > 0:
        for i in range(a.size):
            if a[i] > maxv:
                maxv = a[i]
    container = np.zeros(maxv + 1, dtype=np.int32)
    for i in range(a.size):
        container[a[i]] += 1
    unique_values = np.nonzero(container)[0].astype(np.int32)
    unique_counts = container[unique_values].astype(np.int32)
    return unique_values, unique_counts
```

Figure 2: Examples of proposed code changes. On the top is the initial function, followed by ChatGPTs solution and on the bottom is the code from Gemini 2.5-Pro.

Implementation	Row count	Relative row count change
Baseline	182	0%
ChatGPT5	213	+17%
Gemini 2.5-Pro	262	+43%

Table 1: Row count for each of the implementations. Whitespace and comments are included in the row count.

In addition, Gemini 2.5-Pro implemented its own in-code bounds checks based on elementary operations, while ChatCPT 5 did not. Contributing to the increase in the row count is also the amount of comments. The code review also revealed that Gemini 2.5-Pro was more consistent in code commenting and the comments were much more useful and informative for the developer.

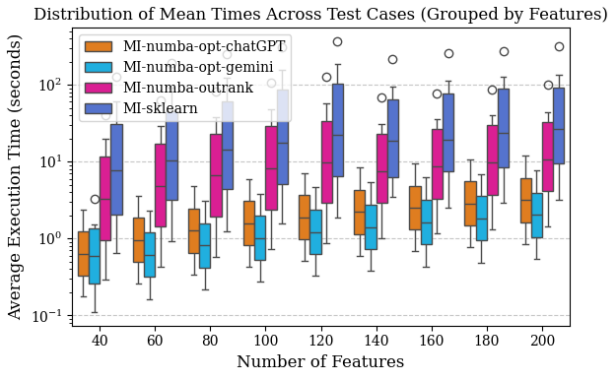


Figure 3: Distribution of Mean Times Across Test Cases (grouped by the number of features) showing the most efficient implementation is the one optimized with Gemini 2.5-Pro.

4 Results

The setup for our benchmark was the following. We evaluated four different implementations of mutual information. For the two baselines, we used the standard and generic Sci-Kit learn mutual information and OutRank's basic MI-numba (that already contains some Numba instructions to optimize the performance). And as discussed before, two LLM optimized implementations were tested— MI-numba-chatgpt5 and MI-numba-gemini, which also support subsampling with a factor in range $(0, 1]$. For the evaluation, the subsampling factor ranges from 0.1 to 1, which means that no subsampling was applied.

To gauge how the performance scales with different parameters of the dataset, namely the number of examples (rows) and number of features (columns), we synthetically generated several datasets, containing raw numerical features with non-negative values (and varied the numbers of examples and features). The number of features ranged from 40 up to 200 in increments of 20, while the number of examples ranged from 200.000 to 20.000.000 in eight logarithmic steps. For each combination, represented by a tuple (algorithm, subsampling factor (where applicable), number of examples, number of features), we made five runs of the code. For each run, we recorded the time to compute mutual information using Python's time function.

The results are shown in Figure 3. The boxes represent 25th percentile in the bottom and 75th percentile on the top. For all test case, the LLM optimized implementations were significantly faster than the baselines (the naive Numba implementation of mutual information from OutRank and the generic Sci-Kit learn mutual information), with Gemini's implementation being the most efficient regardless of the number of features, number of samples or approximation factor. The LLMs sped up the computation of mutual information for approximately 10 times, while the difference between ChatGPT's and Gemini's version was much smaller. This implies that the biggest contribution to the speedup comes from the code changes that the two LLM optimized solutions have in common. Those are primarily the prebuilt grouping, which aims to reduce in-loop allocations, and the heavy use of elementary operations. Although parallelization in the Gemini 2.5-Pro's implementation still plays a role, its effect is less significant.

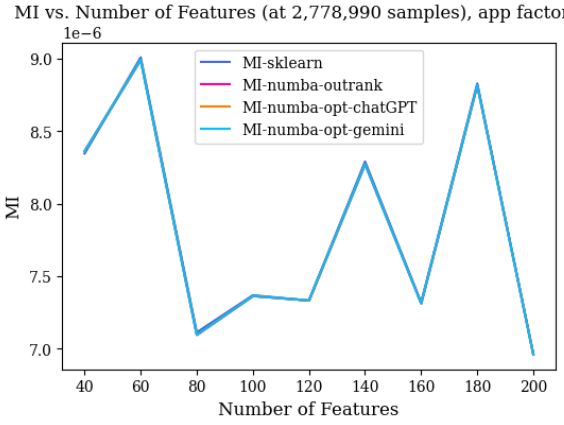


Figure 4: Computed Mutual Information for all tested implementations and for various numbers of feature.

To verify that the computed mutual information is consistent with the generic implementations, namely the Sci-Kit learn implementation, we plotted the mutual information for each number of features. We show the results in Figure 4, where we can observe that the computed mutual information is almost identical for all implementations, regardless of the number of features and different optimizations applied. We conclude that the code optimized by LLMs is valid and correct.

5 Discussion

In our experiment, we used the latest and most advanced versions of two popular LLMs, namely ChatGPT 5 and Gemini 2.5-Pro, with Gemini 2.5-Pro being specifically targeted for coding. While we did put two different LLMs to the test, the goal was not so much to compare them, but to develop a framework that would serve well for evaluating LLM-based optimizations in scientific computing. As the new versions of LLMs and new LLMs are periodically appearing in the market, the framework can serve to keep improving the existing code or, on the other hand, can be used to quantify the improvements (specifically for the coding subdomain) in the LLMs themselves as the new versions are released. Additionally, using the framework in development phase for scientific experiments can reduce the computational time and computational resources needed, leading to a lower cost for the experiments.

Focusing on the LLM aspect of the framework, the question remains what the result of the LLM-based optimization would be, had the context represented by the initial code not used Numba optimizations already. Few additional experiments could be done to explore that:

(1) Use Python code without Numba instructions and explicitly mention Numba in the prompt
(2) Use Python code without Numba instructions and do not mention Numba in the prompt
(3) Task the LLM to prepare the most computationally efficient implementation of mutual information in Python

6 Conclusions

In this work, we presented an initial framework for automatic code optimization via LLM achieving a very impressive 10-fold speedup compared to the naive baseline in the benchmarking experiments while maintaining correctness of the code. We were very impressed by the remarkable similarity of the code produced by two different and independent LLMs. The proposed solutions from both models focused on the same key areas: adding an auxiliary function that creates the pre-built groupings to reduce the in-loop allocations, and shifting the paradigm from native NumPy to C-like Python code relying on elementary operations.

While the optimization process is not yet fully automatic, our contribution outlines a possible direction for efficient use of LLMs in scientific computing. To reach the fully automatic stage when referring to Numba optimization, we propose the following steps are incorporated in the framework:

(1) Use unit test output in case of failure as the next prompt for the LLM to give it a chance to correct the code.
(2) Use the result of the benchmarking experiments as feedback to the LLM and iterate on the proposed optimization.

Both of these suggestions create feedback loops back to the LLMs, enabling an iterative process like the one proposed in Novikov et al. [7]. By comparing the outputs with the existing solutions, we have shown that the LLMs maintained the correctness when introducing optimizations.

References

[1] Mark Chen et al. 2021. Evaluating large language models trained on code. *CoRR*, abs/2107.03374. https://arxiv.org/abs/2107.03374 arXiv: 2107.03374.
[2] Ahmad El Ferdaoussi, Eric Plourde, and Jean Rouat. 2025. Maximizing information in neuron populations for neuromorphic spike encoding. *Neuromorphic Computing and Engineering*, 5, 1, 014002.
[3] Andrew M Fraser and Harry L Swinney. 1986. Independent coordinates for strange attractors from mutual information. *Physical review A*, 33, 2, 1134.
[4] Alexander Kraskov, Harald Stögbauer, and Peter Grassberger. 2011. Erratum: estimating mutual information [phys. rev. e 69, 066138 (2004)]. *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics*, 83, 1, 019903.
[5] Alexander Kraskov, Harald Stögbauer, and Peter Grassberger. 2004. Estimating mutual information. *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics*, 69, 6, 066138.
[6] Siu Kwan Lam, Antoine Pitrou, and Stanley Seibert. 2015. Numba: a llvm-based python jit compiler. In *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC*, 1–6.
[7] Alexander Novikov et al. 2025. Alphaevolve: a coding agent for scientific and algorithmic discovery. *arXiv preprint arXiv:2506.13131*.
[8] Hanchuan Peng, Fuhui Long, and Chris Ding. 2005. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on pattern analysis and machine intelligence*, 27, 8, 1226–1238.
[9] Lior I Shachaf, Elijah Roberts, Patrick Cahan, and Jie Xiao. 2023. Gene regulation network inference using k-nearest neighbor-based mutual information estimation: revisiting an old dream. *BMC bioinformatics*, 24, 1, 84.
[10] Blaz Skrlj and Blaž Mramor. 2023. Outrank: speeding up automl-based model search for large sparse data sets with cardinality-aware feature ranking. In *Proceedings of the 17th ACM Conference on Recommender Systems*, 1078–1083.
[11] Ralf Steuer, Jürgen Kurths, Carsten O Daub, Janko Weise, and Joachim Selbig. 2002. The mutual information: detecting and evaluating dependencies between variables. *Bioinformatics*, 18, suppl_2, S231–S240.

Topological Structure in GitHub Repository Embeddings Using Mapper

Ivo Hrib
ivo.hrib@gmail.com
Jožef Stefan Institute
Ljubljana, Slovenia

Patrik Zajec
patrik.zajec@ijs.si
Jožef Stefan Institute
Ljubljana, Slovenia

Abstract

We present a preliminary framework for the topological analysis of GitHub repository embeddings using the Mapper algorithm. Applied to 10,000 repositories embedded in 768-dimensional space, our approach currently provides visual representations of Mapper graphs, offering a first view into potential topological structures such as branching patterns and cycles. While these initial results are exploratory, they establish a foundation for rigorous statistical testing of topological features. Future work will incorporate persistent homology–based significance testing to distinguish genuine structural patterns from noise, with the ultimate goal of interpreting these features in terms of repository characteristics.

Keywords

topological data analysis, Mapper, GitHub, embeddings, significance testing, software repositories, persistent homology

1 Introduction

We present a preliminary framework for the topological analysis of GitHub repository embeddings using the Mapper algorithm. Applied to 10,000 repositories embedded in 768-dimensional space, our approach provides visual representations of Mapper graphs that reveal branching structures and cycles as potential organizational patterns in the data. These results are exploratory and serve as a foundation for future work, where statistical significance testing will be applied to rigorously validate which features represent genuine topological structure rather than noise. Our framework thus establishes an initial step toward understanding the topology of repository embeddings and motivates further methodological development.

1.1 Research Questions

This work addresses the following specific question:

(1) Do GitHub repository embeddings contain significant topological structures beyond simple clustering?

1.2 Contributions

Our main contributions are:

- A preliminary framework for constructing and visualizing Mapper graphs of GitHub repository embeddings.
- A systematic comparison of Mapper graphs across multiple parameter settings, highlighting sensitivity and recurring structural patterns.
- A discussion of how these preliminary results can guide future work, in particular the application of statistical testing methods to validate topological features and their interpretation in terms of repository characteristics.

Information Society 2025, Ljubljana, Slovenia

https://doi.org/10.70314/is.2025.sikdd.27

2 Background and Related Work

2.1 The Mapper Algorithm

The Mapper algorithm [6] constructs a graph representation of a topological space by combining a filter function, overlapping covers, and clustering. Given a point cloud P embedded in $\mathbb{R}^d$ and a continuous function $f : P \to \mathbb{R}$ refered to as a filter function, the algorithm:

(1) Constructs a cover $\mathcal{U} = \{U_1, \ldots, U_n\}$ of the range $f(P)$ using overlapping intervals
(2) For each interval U_i, computes the preimage $P_{U_i} = f^{-1}(U_i)$
(3) Clusters each preimage into connected components using a clustering algorithm
(4) Creates vertices for each cluster and edges between clusters whose point sets intersect

Common practice uses the first PCA component [4] as the filter and density based clustering methods, such as DBSCAN [2], unless specific domain knowledge is provided. The resulting graph $G = (V, E)$ provides a combinatorial description with mapping $\phi : V \to \mathcal{P}(P)$ associating each vertex with a subset of points.

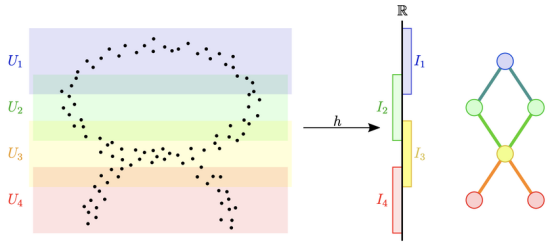


Figure 1: Visual demonstration of mapper algorithm for a projection filter and a simple pointcloud.

2.1.1 Parameter Selection and Sensitivity. Mapper results are sensitive to three main parameters:

- **Resolution** (n): Number of intervals in the cover
- **Overlap** (p): Percentage overlap between consecutive intervals
- **Clustering threshold** (ϵ): Distance parameter for the clustering algorithm

Taking this into account, we devised the following methodology for parameter selection. We define a discrete grid of candidate values, for which the mapper graph is reasonably computable, for each of the previously mentioned parameters and the minimum number of points per cluster. For each point in this grid

we applied the Mapper algorithm to the dataset and computed a collection of quality measures.

Three main criteria were used to assess the quality of each Mapper graph:

- **Coverage:** the proportion of data points captured by the nodes of the Mapper graph, measuring how well the graph represents the entire dataset.
- **Modularity:** a measure of the strength of community structure in the resulting graph, reflecting the presence of well-defined clusters or substructures.
- **Stability:** the reproducibility of the graph under sampling noise, estimated by a bootstrap procedure in which multiple resampled datasets were processed and the resulting node assignments compared for consistency.

For each parameter combination, we computed stability, coverage, and modularity. To aggregate these into a single composite score, we used a weighted sum that places the highest emphasis on stability (0.5), followed by coverage (0.3) and modularity (0.2). These weights were chosen to reflect our prioritization of reproducibility and representativeness over community structure.

n_{cubes}	Overlap	ε	MinPts	Coverage	Stability	Modularity	Score
12	0.70	0.7	3	0.966	0.948	0.785	0.921
12	0.70	0.7	5	0.933	0.924	0.745	0.891
16	0.70	0.7	3	0.952	0.872	0.791	0.880
10	0.70	0.7	3	0.966	0.847	0.765	0.866
16	0.70	0.7	5	0.915	0.852	0.771	0.855

Table 1: Top 5 Mapper parameter settings ranked by score.

For each parameter layout, we employed the first PCA component as our chosen filter and DBSCAN as our chosen clustering algorithm.

2.1.2 Adaptive Mapper and Learnable Filter Functions. Recent advances in Mapper methodology include adaptive approaches in which filter functions are learned from data rather than manually specified. Such approaches could potentially optimize for statistically significant topological features[3]. These methods were, however, not utilized in our case due to computational complexity and remain to be explored in the future.

2.2 Related Work in Software Repository Analysis

Several recent studies have explored software repository embeddings and clustering. For example, Rokon et al. introduced *Repo2Vec*, which combines metadata, source code, and structural signals into repository embeddings for similarity search and clustering [5]. Lherondelle et al. proposed an attention-based model that learns repository embeddings from code and metadata to support auto-tagging and recommendation tasks [**lherondelle2022topical**]. Zhang et al. developed *HiGitClass*, a hierarchical classification framework for GitHub repositories using embedding-based methods [8]. Other work has examined clustering repositories with software metrics [**repo_metrics2023**] or analyzing the characteristics of repositories in specific domains such as embedded systems [**polaczek2021embedded**].

While these approaches demonstrate that embeddings and clustering can yield useful insights about software repositories, they focus primarily on supervised tasks (classification, tagging) or flat similarity clustering. In contrast, our work explores the topological features of the high-dimensional space in which repositories reside. By applying the Mapper algorithm to repository embeddings, we intend to characterize how repositories are organized in terms of branching patterns, hubs, and cycles. This perspective emphasizes the geometry and connectivity of the embedding space itself, offering potential insights that complement more conventional similarity- or classification-based analyses of repositories.

3 Dataset and Methodology

3.1 Dataset Description

The raw dataset comprised approximately 500,000 GitHub repositories, each annotated with a range of metadata fields. These can be grouped into three broad categories:

- **Textual features:** free-form text fields such as description, readme, requirements, and packages, which capture natural-language documentation and dependency declarations.
- **Categorical features:** attributes such as language, topic, and visibility, which provide discrete labels describing repository characteristics.
- **Contextual metadata:** fields such as name, bio, website, company, location, and date of creation, which provide identifying information and organizational context.

3.1.1 Repository Selection Criteria. In the interest of computational feasibility, this dataset was then sampled to 10,000 repositories. Repositories were chosen via simple random sampling from the full dataset, as many repositories contained incomplete or inconsistent categorical and contextual metadata; therefore, stratified sampling was not appropriate.

3.1.2 Embedding Process. Each sampled repository was converted into a structured dictionary combining the available metadata fields. These dictionaries were embedded using the *nomic-embed-text* model. The model accepts long-context inputs (up to approximately 8,000 tokens), which makes it suitable for processing repository documentation such as README files.

The resulting embeddings are 768-dimensional vectors. Together, the 10,000 sampled repositories form a point cloud in $\mathbb{R}^{768}$. Because the embeddings primarily reflect textual and documentation content (e.g., README and description fields), the analysis in this study centers on topological structure in the documentation space rather than source code semantics. These embeddings serve as the basis for the Mapper-based topological data analysis described in the following sections.

3.2 Mapper Implementation

For our purposes we used Kepler-Mapper to get the mapper graphs which scored highest, previously mentioned in **??**.

- Graph 1:
  Resolution = 12, Overlap = 0.7, eps = 0.7, min_samples = 3
- Graph 2:
  Resolution = 12, Overlap = 0.7, eps = 0.7, min_samples = 5
- Graph 3:
  Resolution = 16, Overlap = 0.7, eps = 0.7, min_samples = 3
- Graph 4:
  Resolution = 10, Overlap = 0.7, eps = 0.7, min_samples = 3
- Graph 5:
  Resolution = 16, Overlap = 0.7, eps = 0.7, min_samples = 5

The filter function is, as before, projection onto the first principal component. Clustering using DBSCAN with minimum cluster size parameter adjusted per graph.

4 Results

Table 2 reports the structural properties of the selected Mapper graphs, while Table 3 summarizes degree distributions.

Graph 1 (Resolution = 12, MinPts = 3) produced 207 nodes and 368 edges across 36 components, with 197 cycles. Graph 3 (Resolution = 16, MinPts = 3) was even larger (232 nodes, 421 edges, 229 cycles), reflecting the finer subdivisions introduced by higher resolution.

Graphs 2 and 5 (MinPts = 5) were smaller, around 100 nodes each, as stricter clustering merged many small clusters. Graph 4 (Resolution = 10, MinPts = 3) fell between these extremes (194 nodes, 337 edges).

Degree distributions confirm these patterns: Graphs 1 and 3 contain many nodes of degree 3–5 with some higher-degree hubs, while Graphs 2 and 5 are simpler and tree-like. Overall, higher resolution and lower MinPts yield more fragmented, cycle-rich graphs, while stricter clustering produces fewer, larger components. These trends highlight the need for statistical testing to separate genuine topological signals from parameter effects.

As for the visual representations of the graphs, see 2a and 2c, as well as the bar plots of their respective node sizes . Note That many of the nodes are relatively small most likely due to the reasons mentioned previously.

Graph	Nodes	Edges	Conn. comps.	Cycles (len)
Graph 1	207	368	36	197
Graph 2	101	187	18	104
Graph 3	232	421	40	229
Graph 4	194	337	36	179
Graph 5	108	200	19	111

Table 2: Comparison of structural properties across Mapper graphs.

Graph	1–2	3–5	6–10	11–20	21+
Graph 1	9	181	9	2	6
Graph 2	10	82	2	6	1
Graph 3	20	182	20	6	4
Graph 4	10	171	5	3	5
Graph 5	9	84	7	8	0

Table 3: Binned degree distributions across graphs.

5 Figures and Results Visualization

6 Discussion

The consistent branching patterns across multiple Mapper graphs suggest genuine topological structure in the repository embedding space rather than parameter artifacts.

The large presence of cycles indicates more complex topological relationships beyond simple clustering, possibly representing repositories that share multiple characteristics or form transition regions between different project types. Although most of these may be attributed to noise. We aim to further explore those that are relevant using the techniques from [7] and [1].

6.1 Limitations and Error Analysis

Several limitations must be acknowledged:

6.1.1 Parameter Sensitivity. While we observe some consistent patterns across parameter choices, a more systematic exploration of parameter space is needed than a pure grid search.

6.1.2 Computational Constraints. Full significance testing of all features is computationally expensive, limiting the scale of analysis possible.

6.1.3 Interpretation Challenges. The semantic meaning of topological features requires domain expertise and may not generalize across different types of software projects.

6.1.4 Embedding Model Dependence. Results depend on the quality and characteristics of the embedding model used.

7 Future Work and Conclusions

7.1 Immediate Extensions

- Complete statistical validation of all observed topological features
- Systematic parameter sensitivity analysis
- Comprehensive repository characteristic analysis for interpretation
- Cross-validation with different embedding models and data subsets

7.2 Methodological Advances

- Adaptive Mapper guided by significance testing to optimize filter functions
- Validation on simple synthetic datasets to confirm methodology effectiveness
- Development of Mapper quality metrics and automated parameter selection
- Hybrid approaches combining Mapper with other dimensionality reduction techniques

7.3 Applications and Validation

- Predictive modeling using topological features for repository characteristics
- Integration with software engineering workflows and tools
- Evaluation by domain experts for practical relevance
- Extension to other software engineering datasets and problems

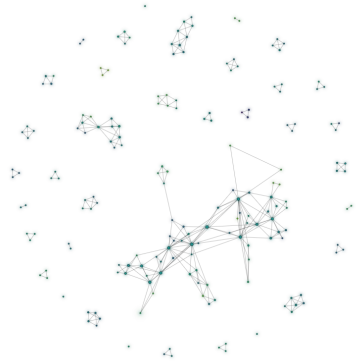
7.4 Conclusions

While computational constraints limit the scope of current analysis, the framework establishes a foundation for rigorous topological analysis of software engineering data. The combination of visualization, statistical validation, and manual interpretation provides a comprehensive approach to understanding high-dimensional repository relationships.

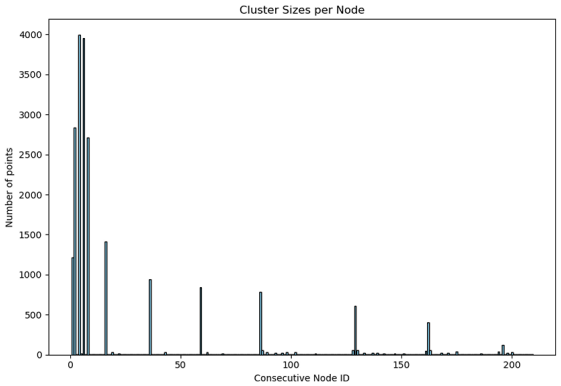
The observed topological structure suggests that repository embeddings capture meaningful relationships beyond simple clustering, opening possibilities for novel applications in software engineering and repository analysis.

Acknowledgements

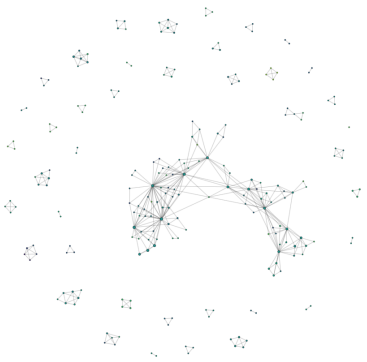
This research was supported by the *EnrichMyData* project, which provided financial support for the work presented in this paper.



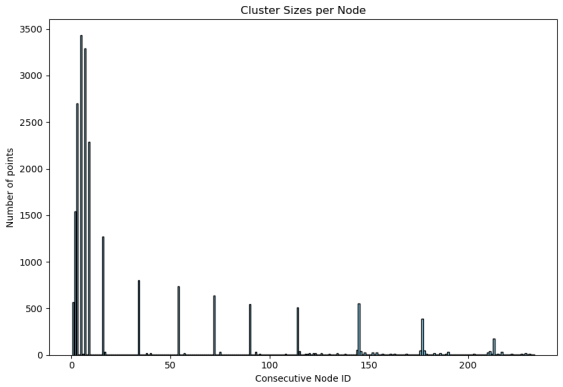
(a) Mapper graph (Graph 1).



(b) Node count distribution (Graph 1).



(c) Mapper graph (Graph 3).



(d) Node count distribution (Graph 3).

Figure 2: Representative Mapper graphs (Graph 1 and Graph 3) with corresponding node count barplots. Both 2a and 2c show a significant central connected component with some branching, however the boundary of the largest connected component seems to be quite noisy. Further statistical testing will aim to improve upon pruning the noisy artifacts.

References

[1] Omer Bobrowski and Primož Skraba. [n. d.] A universal null-distribution for topological data analysis. (). https://www.nature.com/articles/s41598-023-37842-2.

[2] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD'96)*. AAAI Press, 226–231.

[3] Ziyad Oulhaj, Mathieu Carrière, and Bertrand Michel. 2024. Differentiable mapper for topological optimization of data representation. In *Proceedings of the 41st International Conference on Machine Learning* (Proceedings of Machine Learning Research). Vol. 235. PMLR, 38919–38936. doi:10.48550/arXiv.2402.12854.

[4] Karl Pearson. 1901. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2, 11, 559–572. doi:10.1080/14786440109462720.

[5] Md Rafsan Jani Rokon, Panagiotis Kallis, Michele Castronovo, Alexander Serebrenik, and Alberto Bacchelli. 2021. Repo2vec: repository embeddings for effective similarity search and recommendation. In *Proceedings of the 18th International Conference on Mining Software Repositories (MSR 2021)*, 384–394.

[6] Gurjeet Singh, Facundo Mémoli, and Gunnar E. Carlsson. 2007. Topological methods for the analysis of high dimensional data sets and 3d object recognition. In *Eurographics Symposium on Point-Based Graphics*. Eurographics Association, 91–100. doi:10.2312/SPBG/SPBG07/091-100.

[7] Patrik Zajec. 2023. Towards testing the significance of branching points and cycles in mapper graphs. (2023).

[8] Yu Zhang, Frank F. Xu, Sha Li, Yu Meng, Xuan Wang, Qi Li, and Jiawei Han. 2019. Higitclass: keyword-driven hierarchical classification of github repositories. In *ICDM '19*, 876–885. doi:10.1109/ICDM.2019.00098.

CO_2 Monitoring for Energy-Efficient Workloads in Kubernetes: A Data Provider for CO_2-Aware Migration

Ivo Hrib
ivo.hrib@gmail.com
Jožef Stefan Institute
Ljubljana, Slovenia

Jan Šturm
jan.sturm@ijs.s
Jožef Stefan Institute
Ljubljana, Slovenia

Oleksandra Topal
oleksandra.topal@ijs.si
Jožef Stefan Institute
Ljubljana, Slovenia

Maja Škrjanc
maja.skrjanc@ijs.si
Jožef Stefan Institute
Ljubljana, Slovenia

Abstract

We present a CO_2 monitoring component developed within the FAME project's Energy Efficient Analytics Toolbox. The service continuously collects power usage for containerized workloads in Kubernetes via Kepler and fuses it with regional electricity-grid carbon intensity (e.g., ElectricityMaps) to compute per-workload CO_2 emission rates in $g\,s^{-1}$. Its primary role is to *store* accurate, timestamped emission values and expose them through lightweight APIs and an optional time-series database (TimescaleDB). It acts as a data provider consumed by external orchestration services, enabling CO_2-aware migration strategies across clusters and regions.

Keywords

CO2 monitoring, Kubernetes, energy efficiency, carbon-aware computing, time-series storage, ElectricityMaps, Kepler

1 Introduction

Data centers are a significant contributor to global electricity demand. Beyond advances in hardware efficiency and renewable energy procurement, intelligent orchestration of workloads can reduce emissions by aligning computation with cleaner energy availability. A prerequisite for such carbon-aware orchestration is *reliable and accessible measurements* of workload-level emissions.

This paper introduces a CO_2 monitoring and storage service designed for Kubernetes environments. The service ingests pod/-container power data from Kepler [5], combines it with regional grid carbon intensity from ElectricityMaps [2], computes instantaneous emission rates, and *persists* the resulting time series. Unlike optimization or migration tools, this component deliberately restricts its scope: it provides measurements and exposes them via stable APIs for later consumption.

By decoupling measurement from decision-making, we ensure modularity and interoperability. External orchestrators such as the ATOS migration service in FAME D5.4 [3] can consume these metrics to implement CO_2-aware migration strategies without needing to handle the intricacies of measurement or data storage.

Our contributions are threefold: (i) a minimal but complete architecture for per-workload CO_2 measurement and storage in Kubernetes; (ii) a schema and REST API design that facilitates external consumption; and (iii) scenario-based evaluations demonstrating the potential of CO_2-aware workload migration.

Further testing will take place, utilizing real measurements and migrations from within the FAME framework, as to showcase the service's precise final capabilities, as opposed to benchmark tests.

1.1 Key-Idea

The key idea of our approach is to compute container-level CO_2 emissions by combining two complementary data sources: (i) instantaneous power consumption estimates from *Kepler*, and (ii) regional grid carbon intensity values from *ElectricityMaps*.

First, Kepler provides pod- and container-level telemetry in the form of estimated power usage $P(t)$, expressed in watts. This power signal is derived from eBPF-based kernel observations and model-based inference all provided by Keplers data source. Second, ElectricityMaps exposes a carbon intensity factor $I(t)$, expressed in gCO_2/kWh, corresponding to the bidding zone of the node on which the container executes.

We align these two signals in time and compute instantaneous emission rates by:

$$E(t) = P(t) \cdot \frac{I(t)}{3600}$$

where $E(t)$ is the CO_2 emission rate in $g\,s^{-1}$, $P(t)$ is container power in watts ($J\,s^{-1}$), and the division by 3600 converts the intensity factor from per-kWh to per-second units.

These per-container emission rates are then aggregated into a time series, optionally persisted in TimescaleDB, and exposed via a REST API. This composition allows downstream orchestration services to reason about the carbon impact of workloads at fine temporal and spatial granularity, enabling CO_2-aware migration strategies.

2 Background and Related Work

Components of our approach. Our service integrates two external data sources to produce fine-grained CO_2 emission signals. *Kepler* is an open-source project that estimates the energy consumption of containerized workloads in Kubernetes by leveraging eBPF-based telemetry and machine learning models. It exposes per-container power and energy metrics that can be consumed by higher-level services. *ElectricityMaps* provides real-time and historical carbon intensity data for electricity grids, expressed in gCO_2/kWh. By fusing Kepler's workload-level power estimates with regional carbon intensity factors from ElectricityMaps, our system produces a continuous stream of container-level CO_2

Information Society 2025, Ljubljana, Slovenia

https://doi.org/10.70314/is.2025.sikdd.24

emission data. This stream can then be consumed by orchestration or scheduling components for migration and placement decisions.

Carbon-aware computing. Prior research demonstrates the potential of carbon-aware strategies, such as shifting workloads across time or regions to align with lower-carbon electricity supplies [4]. Such approaches rely on access to reliable, fine-grained emission signals to inform scheduling policies.

Existing monitoring tools. Several open-source frameworks exist for energy and carbon monitoring. For example, *CodeCarbon* [1] and *Scaphandre* [6] estimate workload emissions, but they rely on hardware-specific telemetry, such as Intel's Running Average Power Limit (RAPL) counters. This dependence limits portability to Intel CPUs and makes integration across heterogeneous infrastructures challenging. In contrast, our design—built on Kepler and ElectricityMaps—remains hardware-agnostic: eBPF enables container-level monitoring without vendor-specific counters, while ElectricityMaps provides global coverage of carbon intensity signals. This combination makes our service applicable in diverse Kubernetes environments and datacenter setups.

Time-series storage. Finally, for persistence, we optionally employ TimescaleDB, which extends PostgreSQL with hypertables and compression optimized for telemetry data [7]. Nevertheless, the service can also operate in buffer-only mode when persistent storage is not required.

Positioning. This paper positions our monitoring service as a foundational measurement substrate for carbon-aware orchestration in Kubernetes environments. By combining hardware-agnostic energy estimates with real-time grid carbon data, it extends the applicability of carbon-aware scheduling beyond the limitations of prior approaches.

3 Design and Implementation

3.1 Architecture

The component runs as a Kubernetes deployment. Workers collect power metrics from Kepler, fetch grid intensity values, compute emissions, and either persist results in TimescaleDB or serve them from memory. A REST API provides read-only access to historical and recent emissions.

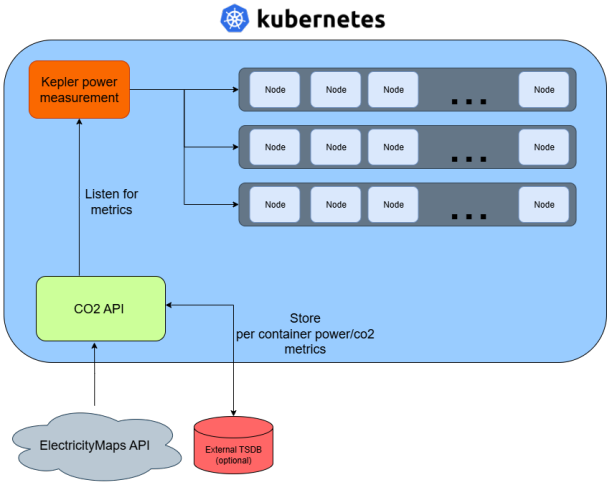


Figure 1: System architecture

3.2 Data Model

Each emission record is structured as a tuple that captures both workload identifiers and measurement values. This schema is designed to balance expressiveness with minimal storage overhead, while ensuring compatibility with external orchestration services.

- `ts` (timestamp, UTC): the precise moment when the measurement was taken, enabling time-series alignment across nodes and regions.
- `namespace`, `pod`, `container`: identifiers for locating the workload within the Kubernetes hierarchy, which is essential for container-level granularity and reproducibility.
- `node`, `region`, `country_iso2`: metadata that ties the container execution to its physical and geographical context. This supports carbon-aware decisions that depend on grid intensity differences across regions.
- `power_w`, `energy_j`: raw telemetry provided by Kepler, describing both instantaneous power and accumulated energy consumption.
- `intensity_g_per_kwh`: regional grid carbon intensity retrieved from ElectricityMaps, serving as the multiplier that translates energy into emissions.
- `co2_g_per_s`: the computed emission signal, representing the core value consumed by orchestrators.
- `source_version`: versioning tag for tracking provenance of measurements and external data dependencies.

This schema ensures that each record is self-contained, interpretable across clusters, and suitable for longitudinal analysis in time-series databases.

3.3 API Endpoints

The service exposes a lightweight REST API, designed to be easily consumed by external orchestrators or monitoring pipelines. The API emphasizes read-only access to maintain reliability and auditability.

- `GET /api/containers`: returns the set of containers currently monitored by the service, allowing orchestrators to discover available emission signals.
- `POST /api/emissions`: fetches recent emission values in bulk. This endpoint is optimized for dashboards or monitoring agents that need timely updates with low overhead. Requires a specified time range to return said emissions.
- `POST /api/emissions/by-container`: queries the emission history of a specific container, Similarly requires a time range, as well as the names of specific containers for which to fetch data.
- `GET /api/schema`: provides the data schema including units and field definitions. This enables clients to validate their assumptions and facilitates long-term interoperability across versions.

By standardizing access patterns, the API makes it possible for external services to reliably retrieve emissions information without depending on internal implementation details.

4 Evaluation

We now present evaluations based on benchmarks and scenario analyses conducted in the FAME project [3]. The goal was to assess whether exposing real-time CO_2 signals can enable meaningful emission reductions when coupled with migration strategies.

4.1 Benchmark Test

In a simple benchmark using busybox, a lightweight Linux container, the optimal CO_2 emissions achieved were significantly lower than the mean observed values. The key performance indicator (KPI) was defined as a 200% improvement, corresponding to a 66.6% reduction compared to baseline. Results show that this threshold can be achieved and often surpassed. The baseline is, for lack of a better, metric defined as the mean of emissions across all tracked countries with available resources for migration.

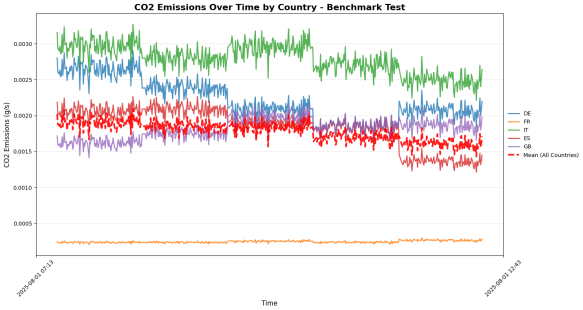


Figure 2: Small timeframe emissions of a benchmark Busybox for testing purposes. We can see noticably low emisions for France, which can be explained by heavy reliance on nuclear power, as can be seen in [2]

4.2 Scenario-Based Evaluation

Scenarios simulate workload migrations across subsets of European countries. Each scenario randomly selects 4–7 countries from a pool of 28, representing constrained deployment options. The system attempts to minimize emissions within these subsets.The abbreviations (e.g., FR, DE, SI) correspond to ISO-3166 country codes representing different electricity regions. We employed random sampling of countries to simulate the heterogeneity faced by cloud and edge providers operating across multiple regions. This choice enables us to reflect migration challenges where workloads are moved not only between datacenters but also across electricity grids with diverse carbon intensities. While random sampling is a simplification, it provides statistically representative insights into the variability of emission factors. We showcase oure results through the following 5 scenarios:

- Scenario 1 (IS, CZ, BG, RO, AT, SE): 88.2% ± 2.1% reduction.
- Scenario 2 (DE, PL, GR, LV): 72.8% ± 5.6% reduction.
- Scenario 3 (GB, LT, SI, DE, AT, GR): 78.0% ± 1.7% reduction.
- Scenario 4 (ES, FR, GB, PL, HU, LT, SE): 89.6% ± 1.1% (best case).
- Scenario 5 (LV, ES, HU, LT): 32.4% ± 12.7% (worst case).
- All Countries: 87.7% ± 1.7% reduction (ideal case).

Across all scenarios, at least one migration was executed per window, with an average emission reduction of 74.8%. These results confirm that even under limited availability, CO_2-aware migration strategies yield substantial benefits.

4.3 Insights

The best-performing scenario demonstrates that careful selection of even a limited number of regions can approach the effectiveness of full global availability. Conversely, the poorest-performing scenario illustrates the dependency on geographic flexibility. Overall, results validate that exposing reliable CO_2 signals through our service empowers orchestration layers to meet or exceed environmental KPIs.

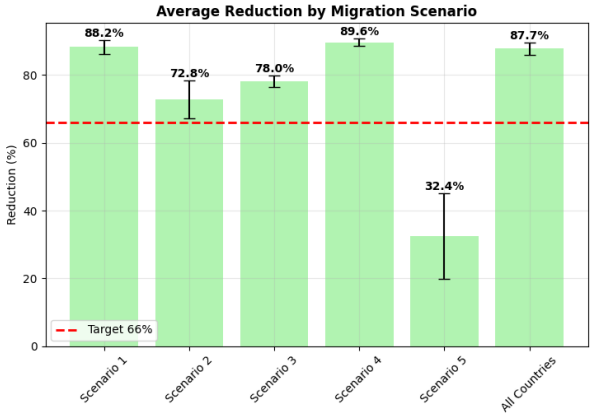


Figure 3: Plot of average reductions per scenario

5 Limitations and Future Work

The reported emissions are estimates subject to the accuracy of both Kepler's models and grid intensity data. As a result, the benchmark tests previously performed may not fully capture all possible scenarios, as grid dependency may sometimes force suboptimal migrations in the CO2-system as per resource availability. Resolution is limited by the update frequency of intensity sources, and storage requirements increase with sampling granularity.

We considered only a single baseline, defined as the mean CO_2 emissions across all tracked countries. While this provides a general reference point, it is not directly comparable to region-specific benchmarks and may obscure finer-grained differences. Future work should therefore incorporate multiple baselines, such as per-country averages or established benchmarks from the literature, and assess statistical significance relative to them.

Our benchmark scenarios were simplified to ensure reproducibility and interpretability. Although random sampling of countries illustrates the variability in energy mixes, it does not fully capture the operational constraints of datacenter migrations or multi-cloud scheduling. More complex benchmarks with realistic workloads and infrastructure heterogeneity would further validate the applicability of our approach.

Finally, while implementation details such as REST endpoints and TimescaleDB integration were reported for transparency, their evaluation was not the main focus of this study. Additional experimentation with scalability and deployment overhead would strengthen the case for adoption in production environments.

Future work will focus on service options to adjust granularity and tackle scalability issues within the service as well as broader evaluation.

6 Conclusion

We presented a Kubernetes-native CO_2 monitoring service that provides real-time emissions data through stable APIs. Evaluations demonstrate that when coupled with migration strategies, these metrics enable significant emission reductions, often surpassing KPI thresholds. Future work will include integration with more compute-intensive workloads, multi-source intensity aggregation, and cryptographic provenance for auditability.

Acknowledgements

This work was supported by the FAME project under the European Union's Horizon Europe programme. We thank the Kepler community and colleagues who contributed feedback during testing.

For all online resources cited, the date of access has been included to ensure reproducibility and traceability.

References

[1] CodeCarbon Project Contributors. 2025. Codecarbon: track and reduce the carbon footprint of your computing. Accessed: 25 September 2025. https://mlco2.github.io/codecarbon/.

[2] Electricity Maps ApS. 2025. Electricity maps: real-time carbon intensity of electricity consumption. Accessed: 25 September 2025. https://app.electricitymaps.com.

[3] European Union Horizon Europe Programme. 2025. Fame project: federated and multicloud enablers for green computing. https://www.fame-horizon.eu/the-project/. Accessed: 25 September 2025. (2025).

[4] Google. 2020. Our data centers now work harder when the sun shines and wind blows. Accessed: 25 September 2025. https://blog.google/inside-google/infrastructure/data-centers-work-harder-sun-shines-wind-blows/.

[5] Kepler Project Contributors. 2025. Kepler: kubernetes-based efficient power level exporter. Accessed: 25 September 2025. https://github.com/sustainable-computing-io/kepler.

[6] Scaphandre Project Contributors. 2025. Scaphandre: energy monitoring agent for linux servers. Accessed: 25 September 2025. https://github.com/hubblo-org/scaphandre.

[7] Timescale Inc. 2025. Timescaledb: an open-source time-series database. Accessed: 25 September 2025. https://www.timescale.com.

Beyond Surveys: Adolescent Profiling via Ecological Momentary Assessment and Mobile Sensing

Jasminka Dobša
University of Zagreb
Faculty of Organization and Informatics
Varaždin, Croatia
jasminka.dobsa@foi.hr

Simona Korenjak-Černe
University of Ljubljana
School of Economics and Business, and IMFM
Ljubljana, Slovenia
simona.cerne@ef.uni-lj.si

Miranda Novak
University of Zagreb
Faculty of Education and Rehabilitation Sciences
Zagreb, Croatia
miranda.novak@erf.unizg.hr

Maja Buhin Pandur
University of Zagreb
Faculty of Organization and Informatics
Varaždin, Croatia
mbuhin@foi.unizg.hr

Lucija Šutić
University of Zagreb
Faculty of Education and Rehabilitation Sciences
Zagreb, Croatia
lucija.sutic@erf.unizg.hr

Abstract

The aim of this study is to identify profiles of adolescents using survey data and data collected via mobile phones, which included ecological momentary assessment (EMA) and passive mobile sensing. EMA involved responses to short questionnaires delivered seven times per day over one week, while mobile sensing captured time spent using different categories of mobile applications. The study was conducted on a sample of 77 secondary school students. Profiling was performed through clustering of EMA data aggregated into six composite variables reflecting confidence, attentiveness, positive and negative emotions related to friends, and overall positive and negative affect. Based on the interpretability of the results, four adolescent profiles were identified. These profiles are further explained using survey data and passive data on mobile application usage patterns.

Keywords

Adolescents, clustering, mobile sensing, ecological momentary assessment, well-being

1 Introduction

This study was conducted using the Effortless Assessment of Risk States (EARS) application developed by Ksana Health in collaboration with the University of Oregon (https://ksanahealth.com/ears/) [6]. The EARS application was originally launched in 2018 to facilitate the collection of high-quality passive mobile sensing data and to support the development of predictive machine learning algorithms capable of identifying risk states for human well-being before they escalate into crises. In 2023 [7], the platform was reintroduced with significant improvements, enabling the collection of behavioral and interpersonal data through natural smartphone use which enabled collection of reported self-ratings known as ecological momentary assessments used in this research.

Previous research using EARS has explored various applications. For instance, one study examined the use of mobile sensing data to assess stress by analyzing affective language captured via smartphone keyboards [4]. Another study investigated the role of friendship quality and well-being in adolescence [9], concluding that adolescents who experienced more positive affect also reported more positive characteristics of close friendships two hours later.

In the present study, profiles of adolescents were identified using EMA variables, resulting in four distinct groups. These profiles were subsequently analyzed with respect to survey data and passive mobile sensing data. The study was guided by the following research questions:

- What distinct adolescent profiles emerge from EMA-based composite variables?
- How are these profiles associated with demographic and psychosocial survey measurements (gender, academic achievement, perceived overuse of social media, level of depression, anxiety, and stress symptoms)?
- What patterns of mobile application use characterize the identified profiles?

The rest of the paper is organized in the following way: in the second section materials and methods are described, the third section presents the results of data analysis, and the fourth section offers a discussion of results and conclusion.

2 Materials and methods

A sample of 77 Croatian high school students participated in this study. We employed three types of data: (1) survey data, (2) EMA data aggregated into six composite variables (confidence, attentiveness, positive and negative emotions related to friends, and overall positive and negative affect), and (3) passive mobile sensing data related to mobile applications usage. The survey

Information Society 2025, 6–10 October 2025, Ljubljana, Slovenia

http://doi.org/10.70314/is.2025.sikdd.29

data included respondents' gender, academic achievement (final grades of 3, 4, or 5), self-reported perceptions of overuse of social media (measured on a scale from 14 to 70), and symptoms of depression, anxiety, and stress (determined by DASS-21 scale, each measured on a scale from 0 to 21 [1]). EMA data and passive mobile data were collected using the EARS application.

Within the framework of ecological momentary assessment (EMA), respondents reported on the quality of their close friendships and their affect, seven times per day over the course of one week (i.e., up to 49 assessments). The assessment schedule followed a semi-random structure: respondents received questions at random intervals within 2-hour windows between 7 a.m. and 9 p.m. Only respondents who completed at least 10 out of 49 assessments were included in the analyses.

Friendship quality was measured with five items rated on a scale from 1 (not at all like me) to 7 (completely like me). All items were adapted from prior studies on close relationships [3, 5, 8]. Two composite variables were derived: *PosFriendEmo*, calculated as the average of three items related to positive friendship-related emotions, and *NegFriendEmo*, calculated as the average of items reflecting negative friendship-related emotions. Items related to positive friendship-related emotions were following:

- "I feel that I can share some worries or secrets with my close friends."
- "I enjoy being with my close friends."
- "I have fun with my close friends."

Items related to negative friendship-related emotions included following statements:

- "I feel that my close friends criticize me."
- "My close friends get on my nerves."

Affect was measured with ten items on the same 7-point scale, adapted from [3]. Two composite variables were created: *PosAffects* (joyful, cheerful, happy, lively, proud) and *NegAffects* (guilty, angry, insecure, scared, sad, worried, ashamed), representing the mean values of the respective items. In addition, a composite variable *Confident* was formed from three items related to peer popularity, self-satisfaction, and body satisfaction, while a composite variable *Attentive* was formed from five items reflecting responsibility, caring for others, perceived adult support, readiness for schoolwork, and perceived teacher support.

Regarding passive data, respondents used a total of 927 applications, which were categorized into 16 groups. Of these, 11 categories were included in the analysis, while the remaining five were excluded due to their negligible usage time. Initial categorization was performed using generative AI tools (Google Bard and ChatGPT) based on app functionality. Each app's classification was then manually verified through its official website to confirm its primary function. Beside variables related to usage of 11 observed categories of mobile apps, variable reflecting the total time spent on the mobile phone (*Total passive*) was also included into the analysis. The analyzed categories included: *Tools and productivity*, *Social media*, *Music and audio*, *Games*, *Communication*, *Multimedia*, *Education and learning*, *Online shopping and services*, *Travel*, *Device management*, and *Entertainment*.

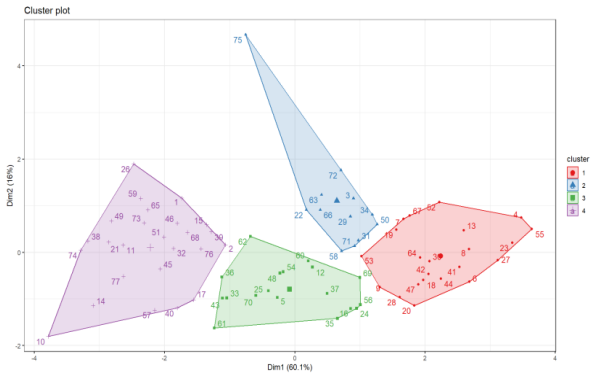


Figure 1. Groups obtained by k-means algorithm projected to the first two principal components of composite EMA variables.

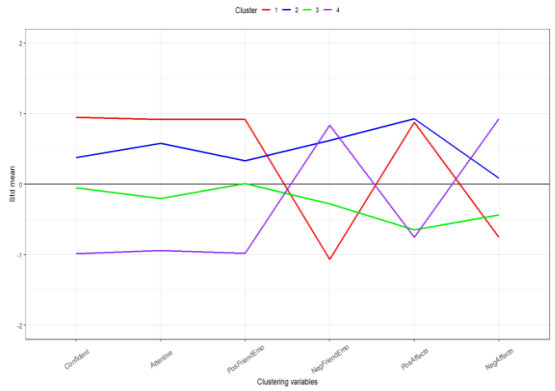


Figure 2. Mean values of standardized composite variables by groups.

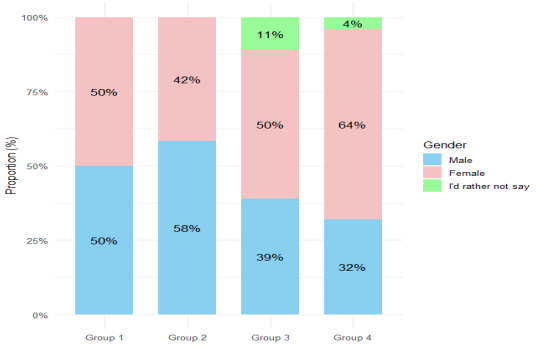


Figure 3. Proportion of respondents by group and gender (male, female, I'd rather not say).

Profiles of adolescents were identified using k-means clustering applied to standardized composite EMA variables. Based on the interpretability of the resulting clusters, the model with four groups was selected.

Data analysis was conducted using **R** statistical software. Group differences were tested using the non-parametric Kruskal–Wallis test, followed by Dunn's post hoc test. Non-parametric tests were applied because analyzed variables were not normally distributed. For the analysis of dependency between groups and their school success it was used chi-square test.

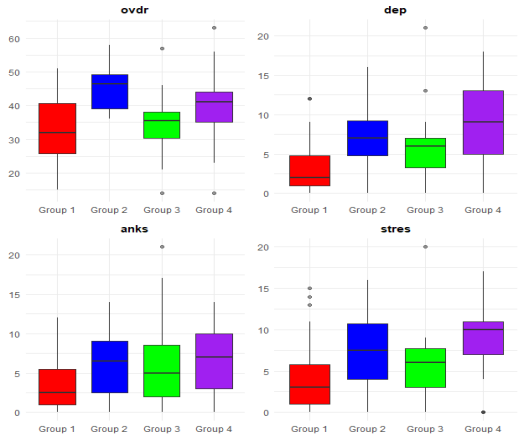


Figure 4. Box-plots for variables of self-assessment of overuse of social media (*ovdr*, 14-70), level of symptoms of depression (*dep*, 0-21), level of symptoms of anxiety (*anks*, 0-21), and level of symptoms of stress (*stres*, 0-21).

3 Results

Figure 1 shows groups of respondents obtained by k-means algorithm projected to the first two principal components of composite EMA variables. Figure 2 illustrates the mean values of the composite variables across groups. Two related pairs of groups can be observed: Groups 1 and 4, and Groups 2 and 3. Groups 1 and 4 display nearly mirror-image profiles with respect to the x-axis. For Group 1, the composite variables *Confident*, *Attentive*, *PosFriendEmo*, and *PosAffects* are above average, whereas in Group 4, these same variables fall below average. Conversely, *NegFriendEmo* and *NegAffects* are below average for Group 1 but above average for Group 4. A similar pattern emerges for Groups 2 and 3, which also show mirror-image profiles, though shifted slightly toward above-average values. Group 3 is characterized by nearly average levels of *Confident*, *Attentive*, and *PosFriendEmo*, while *NegFriendEmo*, *PosAffects*, and *NegAffects* are slightly below average. In contrast, Group 2 demonstrates above-average mean values across all variables. Overall, emotions related to friendships and affective states are less pronounced in Groups 2 and 3 compared to Groups 1 and 4.

Figure 3 shows that female respondents predominate in Groups 3 and 4, in Group 1 there is approximately an equal proportion of male and female respondents, while in Group 2 predominate male respondents. Figure 4 presents the distribution of survey-based variables: self-assessment of overuse of social media (*ovdr*, 14-70), level of symptoms of depression (*dep*, 0-21), anxiety (*anks*, 0-21), and stress (*stres*, 0-21). Group 4 exhibits the highest levels of symptoms of depression, anxiety, and stress. According to the non-parametric Kruskal-Wallis test, there is a significant difference between the groups in symptoms of depression (p=0.0045) and stress (p=0.0162). The Dunn's post hoc test indicated that Group 4 has statistically significant higher levels of symptoms of depression (p=0.0015) and stress (p=0.0090) compared to Group 1. The Kruskal-Wallis test shows that there is a difference in the perception of overuse of social media between the groups (p=0.0024). The highest perceived overuse was reported by Group 2, with a significant difference compared to Group 3 (p=0.0021) and Group 1 (p=0.0040). Results indicate that respondents' perceptions of their social media use did not correspond to the actual time spent on social media (r = 0.0741).

Figure 5 presents the distribution of daily time (in seconds) that respondents spent using different categories of mobile applications across groups. No statistically significant differences were found in the median time spent on social media or in the total time spent across all application categories. Group 1, which showed the highest median values for the composite variables *Confidence*, *Attentiveness*, and positive friendship-related emotions, also reported spending the most time on social media; however, their perception of social media overuse was the lowest among all groups. Group 3, characterized by near-average median values of *Confidence*, *Attentiveness*, positive and negative friendship-related emotions, and affect, demonstrated the highest median usage across most application categories (*Tools and productivity*, *Music and audio*, *Games*, *Communication*, *Education and learning*, *Travel*, *Device management*, and *Entertainment).* The Kruskal-Wallis test revealed a significant difference in application use only for the *Education and learning* category, although Dunn's post hoc test did not confirm differences between specific group pairs. Respondents in Group 4 had the highest median usage of *Multimedia* applications, while those in Group 2 spent the most time on applications related to *Shopping and services*. Notably, respondents in Group 2 were predominantly male and reported the highest perceived overuse of social media among all groups. School success was measured by average grade point, which was 4.05 for Group 1, 4.33 for Group 2, 4.61 for Group 3, and 4.20 for Group 4. The chi-square test indicated a borderline non-significant difference in school success across the groups (p=0.0501). Group 3, which showed the highest median time of application use across most categories, also achieved the highest average grade point (4.61). In contrast, Group 1, which reported the highest levels of confidence and attentiveness in EMA (including perceived readiness for school tasks), had the lowest average grade point.

4 Discussion and conclusion

This study identified four adolescent profiles based on data collected from 77 Croatian high-school students using EMA. Data collected from EMA was aggregated across respondents in the form of 6 composite variables representing their self-reported confidence, attentiveness, positive and negative friendship-related emotions, and positive and negative affect. Two pairs of mirror-image profiles were observed: Groups 2 and 3, and Groups 1 and 4. Emotional states related to friendships and affective states are less pronounced in Groups 2 and 3 compared to other pair of groups, and these groups are characterized by better academic success.

Mobile sensing revealed that respondents used a total of 927 apps, which were categorized into 16 categories, out of which 11 were analyzed in this study. Although social media accounted for the largest share of usage time, no significant group differences were found either in social media use or in total application use. Group 1, according to self-perception, exhibited the most confident and attentive and has lowest median levels of depression, anxiety and stress, spent the most time on social media, but perceived its overuse the least. This group contains approximately an equal proportion of male and female respondents. Group 2, which was predominantly male, spent the most time on *Online shopping and services* and reported the highest perceived overuse of social media, with significant differences compared to Group 1 (p=0.0040) and Group 3 (p=0.0021).

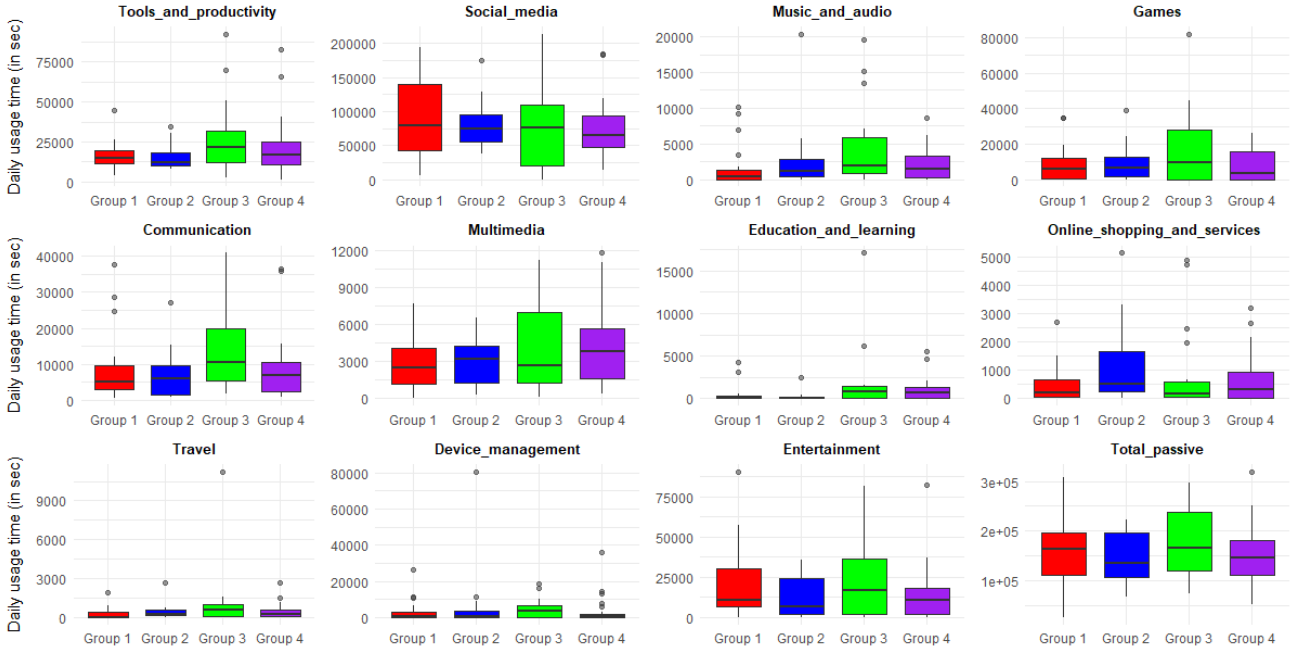


Figure 5. Box-plots for variables of daily usage of categories of mobile applications by groups (in seconds). Note the different ordinal scales due to the large differences in the use of apps.

Group 3, which had the highest academic achievements and the majority of female respondents, had the highest usage of applications in categories *Tools and productivity*, *Music and audio*, *Games*, *Communication*, *Education and learning*, *Travel*, *Device management*, and *Entertainment.* Group 4, also predominantly female, exhibited the highest levels of depression, anxiety, and stress symptoms, spent the least time on social media, used *Multimedia* applications more than other groups, and ranked second in the use of *Education and learning* applications. Importantly, there was no significant correlation between perceived overuse of social media by respondents and their actual time spent using it, as measured by passive sensing. This finding highlights the added value of combining mobile sensing with survey data, as it provides insights that would not be captured through self-report alone. While symptoms of depression, anxiety, and stress were assessed on a 0-21 scale, all median values were below 10, reflecting the general population sample in which the prevalence of psychological problems is low. Future research could therefore focus on adolescents with higher levels of depression, anxiety, and stress symptoms.

In addition, future work will explore the application of symbolic data analysis for clustering based on both EMA and mobile sensing data. Symbolic data analysis, developed for the study of complex and large-scale datasets, incorporates variability directly into the aggregation process [2]. This approach would allow us to account for the stability of emotional states and behavioral patterns at the individual level, potentially offering more refined indicators for defining adolescent profiles.

Acknowledgments

This study was conducted as a part of the Testing the 5C framework of positive youth development: traditional and digital mobile assessment (P.R.O.T.E.C.T.) research project, founded by the Croatian Science Foundation (UIP-2020-02-2852).

References

[1] Antony, M. M., Bieling, P. J., Cox, B. J., Enns, M. W., & Swinson, R. P. 1998. Psychometric properties of the 42-item and 21-item versions of the Depression Anxiety Stress Scales in clinical groups and a community sample. *Psychological Assessment*, 10(2), 176–181. https://doi.org/10.1037/1040-3590.10.2.176

[2] Billard, L., Diday, E. 2007. Symbolic Data Analysis: Conceptual Statistics and Data Mining, 1st edition, Wiley

[3] Bülow, A., van Roekel, E., Boele, S., Denissen, J.J.A. and Keijsers, L.. 2022. Parent –adolescent interaction quality and adolescent affect: An experience sampling study on effect heterogeneity. *Child Development*, 93(3), 315-331, DOI: https://doi.org/10.1111/cdev.13733

[4] Byrne, M.L., Lind, M.N., Horn, S.R., Mills, K.L., Nelson, B.W., Barnes, M.L., Slavich, G.M. and Allen, N.B. 2012. Using mobile sensing data to assess stress: Associations with perceived and lifetime stress, mental health, sleep, and inflammation, *Digital Health*. 2021:7, DOI: 10.1177/20552076211037227

[5] Li, L.M.W., Chen, Q., Gao, H., Li, W.Q. and Ito, K.2021. Online/offline self-disclosure to offline friends and relational outcomes in a diary school: The moderating role of self-esteem and relational closeness. *International Journal of Psychology*, 56(1), 129-137, DOI: https://doi.org/10.1002/ijop.12684

[6] Lind, M.N., Byrne, M.L., Wicks, G., Smidt, A.M., Allen, N.B., 2018. The Effortless Assessment of Risk States (EARS) Tool: An Interpersonal Approach to Mobile Sensing, *JMIR Ment. Health*, 2018; 5(3):e10334, DOI: 10.2196/10334.

[7] Lind, M. N., Kahn, L. E., Crowley, R., Read, W., Wicks, G., Allen, N. B., 2023. Reintroducing the Effortless Assessment Research System (EARS), *JMIR Ment. Health*, 2023; 10:e38920, DOI: 10.21196/38920.

[8] Ng, Y.T., Huo, M., Gleason, M.E., Neff, L.A., Charles, S.T. and Fingerman, K.L. 2021. Friendship in old age: Daily encounters and emotional well-being. *The Journals of Gerontology: Series B*, 76(3), 551-562, DOI: https://doi.org/10.1093/geronb/gbaa007

[9] Šutić, L., van Roekel, E. and Novak, M. 2025. Quality of friendships and well-being in adolescence: daily life study. *International Journal of Adolescence and Youth*, 30(1), DOI: https://doi.org/10.1080/02673843.2025.2467112

Brazil's First AI Regulatory Sandbox: Towards Responsible Innovation

Cristina Godoy Oliveira†
CIAAM, C4AI, Univ. of São Paulo
São Paulo, Brazil
cristinagodoy@usp.br

Joao Paulo Candia Veiga
CIAAM, C4AI, Univ. of São Paulo
São Paulo, Brazil
candia@usp.br

Vasilka Sancin
Faculty of Law, Univ. of Ljubljana
Ljubljana, Slovenia
vasilka.sancin@pf.uni-lj.si

Joao Pita Costa
IRCAI, Quintelligence
Ljubljana, Slovenia
joao.pitacosta@ircai.org

Rafael Meira Silva
CIAAM, C4AI, Univ. of São Paulo
São Paulo, Brazil

Maša Kovič Dine
Faculty of Law, Univ. of Ljubljana
Ljubljana, Slovenia
masa.kovic-dine@pf.uni-lj.si

Lucas Costa dos Anjos
Faculty of Law, Univ. of Juiz de Fora
Juiz de Fora, Brazil

Thiago Gomes Marcilio, Anthony C. de Novaes Silva
CIAAM, C4AI, Univ. of São Paulo
São Paulo, Brazil
tgm.marcilio@gmail.com
anthonycharles.silva@outlook.com

Abstract / Povzetek

As artificial intelligence technologies rapidly evolve, regulatory sandbox initiatives have emerged as crucial tools for promoting responsible AI development, enabling innovation while safeguarding fundamental rights and public interests. This paper analyzes the development and implications of Brazil's first AI regulatory sandbox, with a particular focus on the model established by SUSEP (Superintendence of Private Insurance). Designed as a controlled environment for testing innovative products and services in the insurance sector, the SUSEP sandbox illustrates how regulatory flexibility can foster technological advancement, financial inclusion, and market efficiency while maintaining consumer protection and risk oversight. Being developed under Brazil's Economic Freedom Law, the sandbox has evolved through three editions (2020, 2021, and 2024), prioritizing both sustainable and technological projects. This study explores the sandbox's structure, eligibility criteria, business plan requirements, operational limitations, and transition mechanisms for companies seeking permanent licensure. It also identifies actionable insights for future regulatory frameworks, particularly for the National Data Protection Authority (ANPD) as Brazil advances toward AI-specific governance. By comparing the sandbox's legal foundations, selection processes, and risk mitigation protocols with international best practices, this paper underscores the sandbox's role as a blueprint for responsible AI regulation in emerging markets.

Keywords / Ključne besede

Regulatory Sandboxes, Artificial Intelligence Governance, Data Protection, Innovation Policy, Brazilian AI Regulation

†Corresponding author

Information Society 2025, 6–10 October 2025, Ljubljana, Slovenia

http://doi.org/10.70314/is.2025.sikdd.13

1 Introduction

The rapid evolution of artificial intelligence (AI) has prompted urgent global discussions about governance frameworks that can both stimulate innovation and mitigate potential risks. Around the world, regulators are grappling with how to manage AI systems that are increasingly impacting critical sectors, such as finance, healthcare, education, and public administration. While countries in Europe have taken the lead in formalizing AI-specific legislation—most notably through the European Union's AI Act—many nations in the Global South, including those in South America, are only beginning to articulate coherent regulatory approaches. In Europe, the EU AI Act represents the first comprehensive legal framework for AI, categorizing applications by risk level and imposing strict requirements for high-risk systems. It introduces transparency, accountability, and human oversight obligations, while also fostering innovation through mechanisms such as regulatory sandboxes. This structured and anticipatory approach reflects Europe's long-standing tradition of precautionary regulation and data protection, rooted in the General Data Protection Regulation (GDPR), and with succeeding regulations and standards such as the upcoming European AI Sandbox Act that will further extend Article 57 of the European AI Act, focusing on AI Sandboxes in Europe.

By contrast, AI regulation in Brazil and South America remains fragmented, preliminary, and largely reactive. In Brazil, multiple legislative proposals have been introduced in Congress, but no comprehensive AI law has yet been enacted. The country's current approach relies on a patchwork of sectoral regulations, soft law instruments, and the foundational framework provided by the General Data Protection Law (Lei Geral de Proteção de Dados - LGPD). While the LGPD is a significant step forward in regulating personal data and algorithmic decision-making, it does not address the broader ethical, operational, and societal challenges posed by AI systems. Regionally, South American countries exhibit a similar lack of uniformity. Argentina, Chile, and Colombia have published national AI strategies or draft policy guidelines, yet most remain in early implementation phases. Regulatory oversight is often spread across multiple

agencies, and few jurisdictions have adopted binding legal norms for AI beyond data protection. In this landscape, Brazil stands out as a potential regional leader, particularly through initiatives such as the National Artificial Intelligence Strategy (Estratégia Brasileira de Inteligência Artificial – EBIA) [1], National Artificial Intelligence Plan (Plano Nacional de Inteligência Artificial – PBIA) [2], and the growing role of ANPD.

This paper argues that regulatory sandboxes − flexible, supervised environments for testing innovative solutions −offer a pragmatic and context-sensitive tool for advancing AI governance in Brazil and Latin America. In particular, the experience of the SUSEP Regulatory Sandbox, an experimental regulatory environment created by the Superintendence of Private Insurance (SUSEP) [3] designed for the insurance market, provides a valuable model for structuring oversight of emerging technologies. Through an in-depth analysis of the SUSEP sandbox, this research explores how key regulatory principles−such as proportionality, transparency, risk management, and sustainability−can inform the development of Brazil's first AI sandbox. In doing so, this study contributes to ongoing policy debates about how developing economies can chart their own paths in AI governance, drawing lessons from both global benchmarks and local regulatory experiments. Moreover, it feeds the ongoing collaboration with the different stakeholders in the development of the Slovenian AI Sandbox initiative, hoping for a constructive exchange based of good practices and AI regulation perspectives.

2 Methodology

The SUSEP Regulatory Sandbox is an experimental regulatory environment established to enable the implementation of innovative projects that offer products and/or services in the insurance market. These innovations are developed or offered using new methodologies, processes, procedures, or by applying existing technologies in a novel way. Companies participating in the sandbox can test − under supervision − new products, services, or new ways of providing traditional services. SUSEP assesses the benefits and risks associated with each innovation and determines whether adjustments are needed, either to the business model or to existing regulations.

When the SUSEP Sandbox was launched, it was part of a joint initiative involving the financial, insurance, and capital markets, led by the Central Bank of Brazil (BCB), the Securities and Exchange Commission (CVM), and SUSEP. The SUSEP Sandbox was established during the Bolsonaro administration, in alignment with the Economic Freedom Law (Law No. 13,874/2019) and broader deregulation efforts. There have been three editions so far: in 2020, 2021, and 2024 [4] − with the 2024 edition currently open for an indefinite period. The SUSEP Sandbox is governed by CNSP Resolution No. 381/2020, as amended, along with SUSEP Circular No. 598/2020, and by specific public notices for each edition. The National Private Insurance Council (CNSP) sets the rules for the insurance market, and SUSEP ensures compliance.

ANPD's Regulatory Sandbox, on the other hand, is structured to comprehensively evaluate the technical, legal, ethical, and social dimensions of AI-based projects involving personal data. It adopts a multidisciplinary approach encompassing organizational, technological, and regulatory aspects. Participants are required to present a detailed description of the problem or opportunity addressed by their project, highlighting the current context, challenges, and expected benefits, such as innovation and efficiency. The methodology emphasizes the innovative aspects of the solution, the processing of personal data in AI systems, the social impact, and the intended outcomes.

A core component of the methodology is the implementation of algorithmic transparency measures. Applicants must describe how their systems will make algorithmic logic, decisions, and criteria understandable to end users. This includes the use of explainable AI (XAI) tools, audit reports, documentation, and dashboards, as well as practices for data traceability and decision accountability. The methodology also requires information on compliance with the LGPD, such as data minimization, risk management, mitigation of algorithmic bias, governance mechanisms, and respect for data subject rights. Projects must show alignment with ethical and legal standards to ensure responsible AI development.

In terms of data methodology, applicants must describe the lifecycle of the personal data used, including its origin, collection, processing, storage, and disposal. In addition, the quality of data is crucial, and applicants must describe it to demonstrate that they are in a good phase to participate in the regulatory sandbox. A preliminary impact assessment on data protection must be included, along with a risk matrix that identifies potential harms to data subjects and proposes mitigation strategies. The form also assesses the technical feasibility of the project by requiring information on the IT infrastructure (cloud, hybrid, on-premises), API data flows, outsourcing arrangements, LLM usage, and cybersecurity controls. Financial planning (FINOPS), scalability, social impact assessment, and performance metrics are also critical elements of the methodology.

Finally, organizations must consolidate their identified risks and mitigation measures into a summary framework, ensuring transparency and accountability throughout the project lifecycle.

3 Legislation, Regulation, and Ethical Use: Objectives and Priorities

In the 2024 edition of the SUSEP Regulatory Sandbox, participating companies were required to submit detailed information and upload relevant documents through Brazil's Electronic Information System (SEI)**.** The sandbox was designed to: (i) stimulate competition to improve efficiency; (ii) promote financial inclusion; (iii) encourage capital formation and efficient resource allocation; and (iv) develop and deepen the Brazilian insurance market.

SUSEP prioritized proposals classified by the applicants themselves as either Sustainable or Technological projects:

- **Sustainable Projects**: Aligned with SUSEP and CNSP rules, as well as the Federal Government's Ecological Transformation Plan. These initiatives must deliver climate, environmental, or social benefits to policyholders, beneficiaries, or society as a whole.
- **Technological Projects**: Promote the development of innovative technology by introducing technological

novelties or enhancements to products, services, business models, or processes, thereby adding functionality or quality improvements.

Regarding the eligibility criteria for startups (insurtechs), applicants were required to offer an innovative product or service and operate via remote/digital platforms. They should demonstrate the novelty of their technology or its creative application and present the solution in a development stage suitable for temporary authorization. Moreover, they had to submit a business plan, which included a risk assessment, specifically addressing cybersecurity, and a damage mitigation plan. Besides the typical proposed and current legal/trade names, or organizational structure and director profiles, the business plan had to include strategic objectives, and company history, mission, and vision, along with a problem statement and market/consumer benefits, proof of concept of product or service and demonstration of potential cost reduction for consumers, if any. It also described a comparative analysis with existing offerings, target market, and geographic scope, along with risk factors and mitigation strategies, the technical architecture and operational model, the justification for the Priority Project classification, and the sustainability policy. The selection process involved two stages: (i) a Selection Phase with a video interview with SUSEP; and a (ii) Temporary Authorization Phase**,** with a follow-up interview and submission of evidence proving compliance with normative requirements and completion of corporate formalities, as well as appointment of a director responsible for sandbox participation and documentary evidence attesting to the lawful origin of funds contributed by investors.

4 Discussion of initial results

The 2024 edition of this initiative included four companies that were granted permanent licenses (by September), while 32 projects were selected, amongst which 21 received temporary authorization (by April). Authorized companies were required to transmit operational data to SUSEP via API. While in the sandbox, companies:

- can only sell approved types of insurance,
- operate under capped risk exposure, and
- face limits on claims payouts.

Given the similarities between insurance regulation and data protection governance, several SUSEP sandbox practices could inform the design of an AI sandbox under Brazil’s National Data Protection Authority (ANPD), such as:

1. Innovation focus – Projects must demonstrate clear novelty or novel applications of technologies, methods and procedures.
2. Sustainability integration – For AI, this could include energy, water and natural resources efficiency, environmental impact, and ethical safeguards.
3. Defined operational boundaries – Limitations on AI use cases, affected populations, and permitted risk categories.
4. Mandatory submissions – Risk analysis and mitigation plan, business plan, and funding source verification.
5. AI registry – Formal registration with ANPD, with authorizations subject to revocation.
6. Virtual interviews – Ensuring nationwide accessibility.
7. Exit Strategy – A clear post-sandbox transition plan for continued compliance.

In Phase 1 of the ANPD’s regulatory sandbox selection process, whose application period closed on August 25, 2025, additional points will be awarded to startups, public sector organizations, and companies developing generative AI solutions. These categories were identified as strategic priorities for Brazil: startups are legally recognized in the Brazilian Innovation Framework [5] as key beneficiaries of sandbox initiatives; public sector organizations often develop socially impactful solutions and are expected to sustain participation without financial or technical aid from ANPD; and Brazil has an explicit national interest in fostering large language models (LLMs) in Portuguese as part of its broader AI sovereignty strategy.

As part of the application process, the ANPD’s form required that any confidential or sensitive business information be clearly marked as such by the applicants. This provision is necessary due to Brazil’s Freedom of Information Law (Lei de Acesso à Informação – LAI), which mandates public disclosure unless a legal exception is claimed. Without this explicit classification, all submitted materials may be treated as public, potentially exposing strategic or proprietary information from participating firms.

To enhance visibility and inclusiveness, the ANPD also adopted a multi-channel outreach strategy, disseminating the call for applications through official platforms and with the support of civil society organizations. To maximize participation, the deadline for applications was extended by an additional 15 days, although the overall schedule for evaluation and publication of results remained unchanged. The final list of selected participants is scheduled to be released on October 2, 2025, as originally planned.

Finally, there is also another point of flexibility, not expressly codified, which is the absence of a fixed taxonomy of sandboxes. For example, the SUSEP sandbox has an innovative character, seeking to make regulations more flexible. At the same time, the service is being used in the market. In contrast, the ANPD sandbox aims to provide the regulator with knowledge that enables the preventive updating of market rules, rather than a reactive one. Oversight may be distributed among agencies like SUSEP, yet the regulatory status of AI companies post-sandbox remains unclear. For this reason, ANPD must establish both sandbox-specific rules and post-sandbox AI regulations, ensuring long-term supervision and market stability.

The importance of embedding responsible and ethical principles in AI governance is particularly acute in Brazil and across South America, where technological innovation intersects with social inequality, fragile institutions, and diverse regulatory capacities. By prioritizing transparency, accountability, and fairness in AI systems, these countries can foster public trust while mitigating risks of discrimination, exclusion, or misuse of personal data. Brazil’s initiatives—such as its National AI Strategy (EBIA), the forthcoming AI legal framework, and the regulatory sandbox programs led by SUSEP and the ANPD—illustrate how

developing nations can create adaptive governance models that balance innovation with fundamental rights. Moreover, as the largest economy in Latin America, Brazil is well-positioned to serve as a regional benchmark, showing how ethical AI practices can promote financial inclusion, strengthen democratic values, and encourage sustainable development. In this sense, South America's experience underscores that responsible AI is not a luxury for advanced economies but a prerequisite for equitable technological progress in the Global South.

5 Conclusions and further work

The ANPD's regulatory sandbox demonstrates Brazil's commitment to experimental and responsible governance of AI. By ensuring transparency through a public information portal, addressing confidentiality in accordance with the Access to Information Law, and promoting inclusive engagement, the initiative aligns with international standards. Drawing on frameworks such as the OECD's recommendations and the EU's AI Act, which formally includes regulatory sandboxes, the Brazilian approach reinforces the importance of embedding such mechanisms into national legislation. In the context of Bill 2338/2023 (under debate in the Deputy Chamber to regulate AI in Brazil) [6], regulatory sandboxes emerge as strategic tools to enable adaptive, participatory, and context-aware AI regulation.

The Brazilian AI sandbox experience also carries significant relevance beyond Brazil and South America, offering valuable insights for other developing countries and even jurisdictions with more advanced regulatory frameworks, such as Europe. While the European Union has already institutionalized AI sandboxes within the AI Act, the Brazilian model demonstrates how experimental, flexible, and context-sensitive approaches can be adapted to environments where regulatory structures are less consolidated. Its emphasis on transparency, proportionality, and multi-stakeholder participation shows that effective governance does not require fully mature institutions but rather innovative mechanisms that align local priorities with global best practices. By proving that responsible innovation can be pursued within resource-constrained and diverse legal settings, the Brazilian sandbox contributes to a global dialogue on AI governance, helping countries at different stages of regulatory development to tailor sandbox initiatives to their specific socio-economic and institutional realities.

Acknowledgments / Zahvala

Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here.

References / Literatura

[1] MCTI (2021). Brazilian Strategy of Artificial Intelligence. [Online]. Available: ebia-documento_referencia_4-979_2021.pdf (www.gov.br)

[2] PBIA (2024). Brazilian Artificial Intelligence Plan . [Online]. Available: https://www.gov.br/mcti/pt-br/acompanhe-o-mcti/noticias/2024/07/plano-brasileiro-de-ia-tera-supercomputador-e-investimento-de-r-23-bilhoes-em-quatro-anos/ia_para_o_bem_de_todos.pdf/view

[3] Brazilian Government Portal (2025). About SUSEP. [Online]. Available: https://www.gov.br/susep/pt-br/acesso-a-informacao/institucional/sobre-a-susep

[4] Brazil (2019). JOINT STATEMENT: COORDINATED ACTION TO IMPLEMENT A REGULATORY SANDBOX REGIME IN THE BRAZILIAN FINANCIAL, SECURITIES, AND CAPITAL MARKETS. [Online]. Available: https://www.gov.br/susep/pt-br/central-de-conteudos/noticias/2022/noticia

[5] Brazil (2021). Complementary Law No. 182, of June 1, 2021. Establishes the Legal Framework for Startups and Innovative Entrepreneurship. [Online]. Available: planalto.gov.br/ccivil_03/leis/lcp/lcp182.htm

[6] Brazil (2023). Bill No. 2338, of 2023. Establishes the legal framework for artificial intelligence in Brazil. [Online]. Available: https://www.camara.leg.br/proposicoesWeb/prop_mostrarintegra?codteor=2868197&filename=PL%202338/2023

Indeks avtorjev / Author index

Anjos Lucas Costa dos 122
Barrionuevo Leonardo 98
Bašić Nino 102
Batagelj Vladimir 102
Brank Janez 11
Calcina Erik 7
Camlek Neca 29
Caporusso Jaya 19
Cek Rok 82
Ćetković Marija 65
Čibej Jaka 53
Costa João Pita 45, 98, 122
Debeljak Žiga 57
Dine Masa Kovic 122
Dobša Jasminka 118
Forcolin Marherita 82
Frattini Matteo 49
Grobelnik Adrian Mladenić 15, 25
Grobelnik Marko 11, 15, 25, 29, 73
Guček Alenka 25, 69
Guo Zhenyu 33
Hegler Živa 29
Hosseini Seyed Iman 86
Hrib Ivo 110, 114
Jakomin Martin 106
Jelenčič Jakob 29
Jeršek Domen 49
Kassis Rayan 45
Kavšek Branko 94
Kenda Klemen 49, 57, 77, 82, 86
Kladnik Matic 41
Klančič Rok 49
Kochovska Sofija 94
Kocjančič Oskar 37
Korenjak-Černe Simona 118
Kozamernik Lučka 106
Krumpak Roy 33
Lamgari Asmai 45
Leonardi Linda 82
Leskovec Gašper 77
Ma Xiang 33
Marcilio Thiago Gomes 122
Mladenić Dunja 7, 11, 29, 33, 41, 57, 61, 73, 86
Mochariq Ouidad 45
Mylonas Costas 77
Novak Erik 7
Novak Miranda 118
Novalija Inna 11, 33, 61, 73
Oliveira Cristina Godoy 122
Pandur Maja Buhin 118
Pavlova Daria 61, 90
Pisanski Tomaž 102
Pollak Senja 19
Polzer Mirozlav 98
Purver Matthew 19

Rahmani Yousef 45
Roman Dumitru 33
Rožanec Jože M. 33
Sancin Vasilka 122
Savnik Iztok 102
Silva Anthony Novaes 122
Silva Rafael Meira 122
Sittar Abdul 69
Škrjanc Maja 73, 114
Škrlj Blaž 106
Slavec Ana 102
Smiljanic Mateja 69
Song Tao 33
Souss Sohaib 45
Stopar Luka 45
Šturm Jan 73, 114
Šutić Lucija 118
Topal Oleksandra 73, 82, 114
Tošić Aleksandar 65
Trajkov Georgi 15
Urbančič Jasna 106
Vake Domen 65
Veiga João Cândia 98, 122
Vičič Jernej 94
Zajec Patrik 110
Zaouini Mustafa 45
Žnidaršič Martin 37
Žust Martin 25