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.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
Information Society 2025, 6–10 October 2025, Ljubljana, Slovenia
© 2024 Copyright held by the owner/author(s).
https://doi.org/10.70314/is.2025.sikdd.6

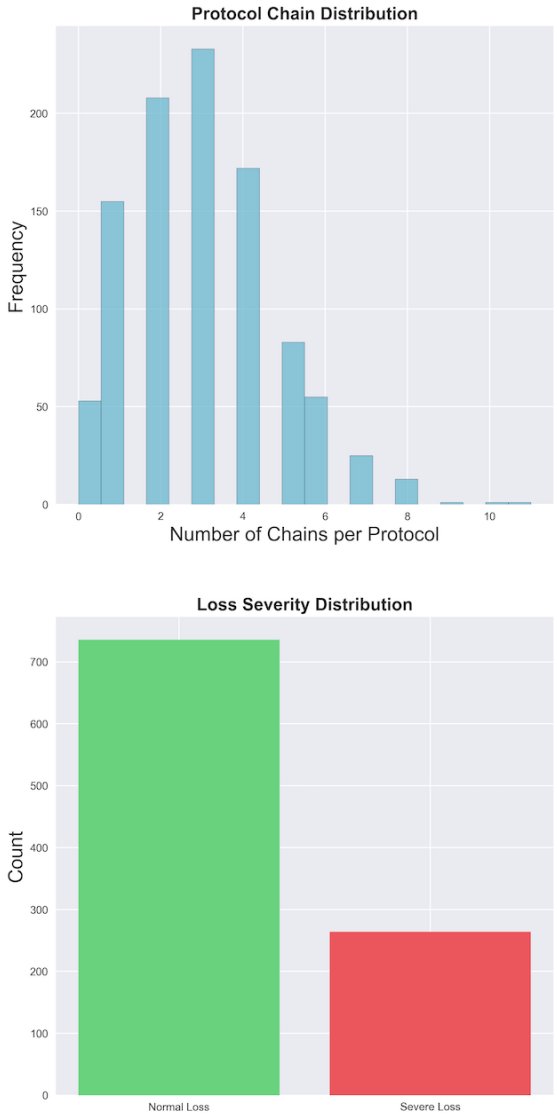


Figure 1: DeFi knowledge graph overview: protocols, blockchains, and incidents with relations (forked-from, deployed-on, involves).



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.

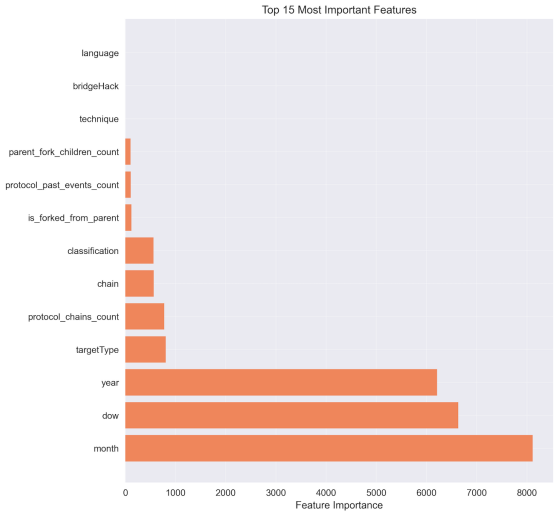


Figure 3: Workflow: derive graph-based features from the DeFi knowledge graph and combine with conventional incident features for classification.

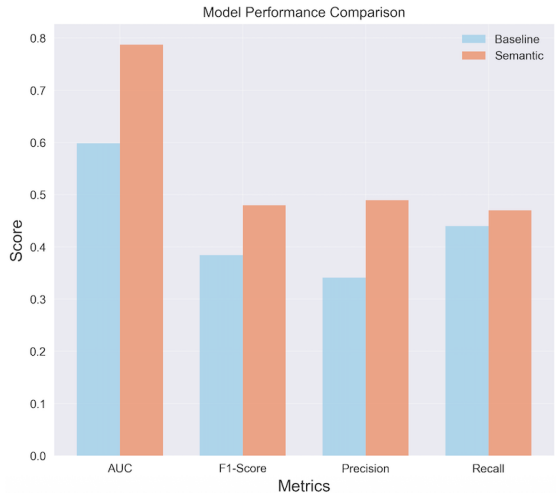


Figure 4: Performance comparison. Bar chart for AUC, F1, precision, recall.

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.

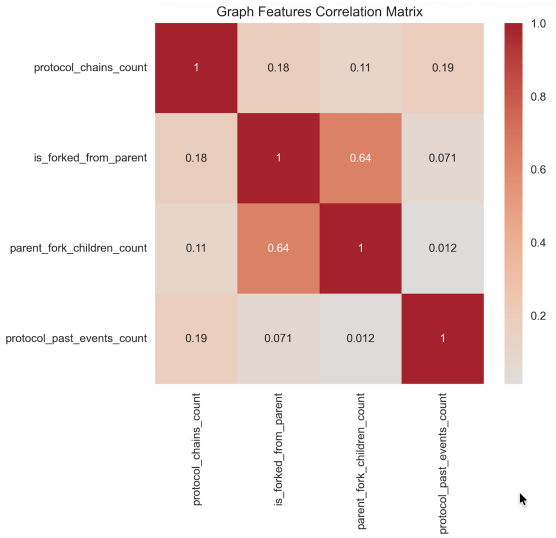


Figure 5: Top 15 most important features ranked by LightGBM gain. Values on the x-axis represent LightGBM's internal feature importance scores (dimensionless, aggregated across all trees in the ensemble). Both temporal features (year, month, day-of-week) and graph-based features (e.g., *protocol_chains_count*, *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).