Comparative Analysis of Machine Learning Models for Groundwater Level Forecasting: The Impact of Contextual Data

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

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

Abstract

This paper presents a comparative evaluation of three distinct categories of models applied to groundwater level data: traditional batch learning methods, time series deep learning methods, and time series foundation models. By enriching the water level data with weather-related features, we significantly improved the effectiveness of simpler models. The results demonstrate that, despite their state-of-the-art performance on univariate datasets and the corresponding publicity, advanced models without contextual feature support are still surpassed by traditional methods trained on enriched datasets.

Keywords

groundwater level prediction, time series forecasting, deep learning, foundation models, contextual data

1 Introduction

Accurate water level prediction is crucial for mitigating the impacts of climate change on water resources. By forecasting water levels, we can better prepare for potential floods and droughts, and more effectively manage our water supplies. However, predicting water levels presents a significant challenge due to the dynamic nature of the data. As climate change leads to prolonged droughts and increasingly erratic precipitation patterns, the need for reliable forecasting methods becomes even more important [2].

In this paper, we aim to compare the performance of various models in forecasting groundwater levels. Specifically, we focus on the differences between traditional batch learning methods that utilize relevant contextual data and newer univariate time series deep learning and foundation models.

The main contributions of this paper are:

- A comparative analysis of the performance of traditional batch learning methods against state-of-the-art time series deep learning techniques and time series foundation models, particularly in the context of feature vectors enriched with relevant contextual data.
- The application of time series foundation models and deep learning methods to the domain of groundwater level forecasting.

The groundwater dataset used in this study has previously been employed for predictive modeling with traditional batch learning methods [9], where extensive feature engineering was also performed. Our work builds upon and extends this earlier research by incorporating a different set of models.

2 Methods

In our experiments, we employed three categories of methods: traditional batch learning techniques, time series deep learning models, and time series foundation models.

2.1 Traditional Batch Learning Methods

In the context of data-driven modelling of environmental issues, traditional batch learning methods have historically demonstrated significant success [5]. In this study, we employed linear regression alongside two tree-based approaches: random forest and gradient boosting [7] as baselines to evaluate whether the newer, more prominent techniques, which have recently gathered a considerable amount of attention, can perform competitively in this specific setting.

All of the chosen batch learning techniques are regression-based and are valued for their simplicity, speed, and ease of use. However, they often lack the complexity necessary to fully capture intricate patterns in the data. To mitigate this limitation, we incorporated contextual features, such as weather data and forecasts (e.g., precipitation, cloud cover, temperature). While the data fusion problem is solved [8], this approach raises concerns about the availability and relevance of the contextual data.

2.2 Time Series Deep Learning Methods

Time series deep learning models are explicitly designed for forecasting time-dependent data. In our study, we employed N-BEATS [12] and PatchTST [10], both of which have architectures tailored to capture trends and seasonalities inherent in time series data. Despite their advanced capabilities, these models have drawbacks, including longer training and inference times, the necessity for extensive hyperparameter tuning to achieve optimal performance, and limited support for incorporating additional features. Although certain models support multivariate time series, they were not utilized in our experiments.

2.3 Time Series Foundation Models

While deep learning methods require separate training and prediction phases, time series foundation models aim to eliminate the training step. Inspired by large language models, these models are pretrained on extensive time series datasets, enabling zero-shot predictions on new time series without additional training. We used CHRONOS [1], an open source foundation model. The advantages of this approach include ease of use with minimal parameter adjustments and no need for training. However, similar to deep learning models, they lack support for multivariate time series.

Several studies have already evaluated the performance of various deep learning and foundation models for time series forecasting [1] [13]. However, this research extends the application of these forecasting models to groundwater level data, therefore contributing to the better understanding of their effectiveness in this domain.

Information Society 2024, 7–11 October 2024, Ljubljana, Slovenia

https://doi.org/10.70314/is.2024.sikdd.6

3 Experiment Setting

The experiments were conducted on a dataset of groundwater levels in Slovenia. Due to the cumulative nature of water levels and to facilitate comparison with the original study [9], predictions were made on daily changes in water levels rather than on absolute values.

3.1 Dataset

The groundwater dataset is a subset of the larger dataset used in the study [9]. It consists of groundwater level measurements taken daily from multiple stations across Slovenia. To apply traditional batch learning methods, we enriched the dataset with weather data, associating each water measurement station with the nearest weather station. Due to the availability of weather data, only data from the years 2010 to 2017 was included in our study. For consistency and ease of comparison with previous study [9], we focused on data from two water measurement stations located in Ljubljana.

In traditional batch learning within the environmental domain, it is essential to not only use the raw data but also to engineer relevant features. Initially, we removed the pressure and dew point features, as they were either unrelated to the target variable or highly correlated with other features [9]. We then created additional features by shifting the data from 1 to 10 days, making historical values available, and by computing the averages of features over a 2- to 10-day window. This process resulted in approximately 2,000 features. Given the excessive number of features, which could degrade model performance, we employed a feature selection algorithm to identify the most informative subset.

We used a genetic feature selection algorithm from scikit-learn, evaluated on 365-day part of training dataset, with the maximum number of features set to 40. The algorithm was executed separately for each model, focusing on one station and a prediction horizon of three days, resulting in distinct feature vectors. Subsequently, weather forecast features with longer offsets were manually added to the selected feature set.

3.2 Evaluation Metrics

The dataset was split into a training set (approx. 2,500 days), a validation set (100 days), and a test set (365 days) for model evaluation. Model performance was evaluated using the R^2 score, averaged across all tested stations. Although alternative metrics such as root-mean-squared error (RMSE), and mean absolute percentage error (MAPE) were considered, they, for this dataset, produce results that are closely related to the R^2. This metric was selected due to its robustness against variations in data offset and amplitude, and for direct comparability with the results in the original study [9]. The R^2 score is defined as:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2},$$

where y_i is the i-th true value, $\hat{y}_i$ is the i-th predicted value and $\bar{y}$ is the average of true values.

3.3 Baseline Methods

The primary objective of our research was to compare the performance of traditional batch learning methods, enriched with relevant contextual features, against that of modern deep learning techniques and foundation models for time series forecasting. Therefore, we selected linear regression, random forest regressor, and gradient boosting regressor as our baseline methods. These models were previously applied to the groundwater dataset [9], necessitating a reproduction of the results as a benchmark.

3.4 Implementation Details

The prediction pipelines varied slightly between the different types of models:

- For **CHRONOS**, we utilized the dataset without weather features, as it only supports univariate time series. Since no hyperparameter tuning was required, the data was divided into training and test sets, omitting the validation set. The model generated the predictions directly from the water level data. We used the chronos-t5-large model from the chronos library.
- For **N-BEATS** and **PatchTST**, the same dataset was used, given the same limitation as mentioned previously. However, a validation set was required for hyperparameter tuning. After selecting appropriate hyperparameters, the models were trained on the training set and evaluated on the test set. Implementations from the NeuralForecast library were used for both models.
- For the **linear regression**, **random forest regressor**, and **gradient boosting regressor** models, we included both water level and weather data. Feature selection was conducted to reduce the number of features, resulting in 42 features for linear regression, 30 for random forest, and 36 for gradient boosting. After feature selection, hyperparameters for the random forest and gradient boosting models were tuned, and the data for linear regression was normalized. The models were then trained on the training set and evaluated on the test set using scikit-learn's implementations.

The hyperparameters used for training are listed in Appendix A, while a description of the selected features is provided in Appendix B.

4 Results

The results for all tested models across various prediction horizons are presented in Table 1. The reported R^2 scores were calculated based on the differences in water levels; if absolute water levels had been used, the R^2 scores would have been significantly higher. For example, in the case of CHRONOS with 1-day ahead predictions, the R^2 score is 0.725 for relative level differences and 0.998 for absolute water levels.

Among the models, linear regression achieved the highest performance, followed by the random forest. In contrast, the more complex methods, including deep learning models and the foundation model, showed generally lower performance, with the exception of the 1-day prediction horizon, where N-BEATS outperformed the tree-based models. Notably, the R^2 scores decrease as the prediction horizon lengthens, with a more pronounced decline observed in the deep learning and the foundation models compared to the traditional batch learning methods.

Figures 2 and 3 display the predictions from CHRONOS, PatchTST, and linear regression compared to the true data for the 1-day and 5-day prediction horizons. It is evident that the predictions from CHRONOS and PatchTST begin to exhibit a rightward shift as the horizon extends. Figure 1 visualizes the R^2 scores for all models across the different prediction horizons.

Table 1: R^2 Scores for Different Prediction Horizons and Models.

Methods	1 day ahead	2 days ahead	3 days ahead	4 days ahead	5 days ahead
Chronos-large	0,725	0,365	0,175	0,04	-0,09
GradientBoostingRegressor	0,640	0,603	0,527	0,556	0,545
RandomForestRegressor	0,726	0,697	0,701	0,706	0,691
N-BEATS	0,742	0,397	0,17	-0,03	-0,143
PatchTST	0,721	0,394	0,215	0,109	-0,02
LinearRegression	**0,792**	**0,781**	**0,785**	**0,784**	**0,780**

The best and second-best results are bolded and underlined respectively.

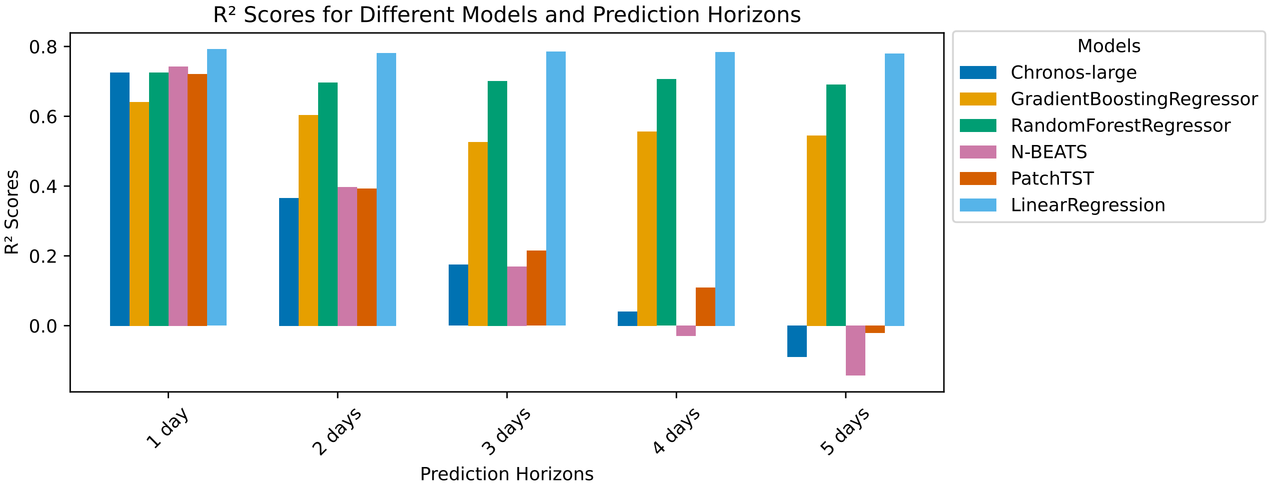

Figure 1: R^2 Scores for All of the Methods and Prediction Horizons.

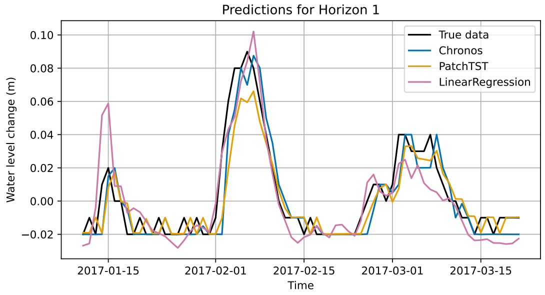

Figure 2: Example Predictions for Three Models for 1-Day Prediction Horizon.

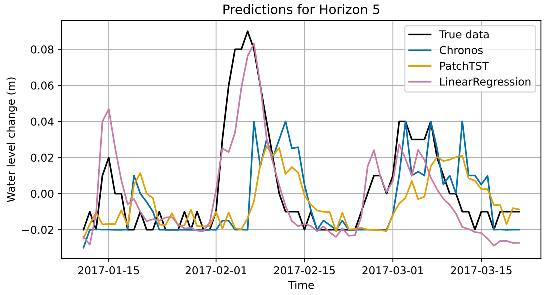

Figure 3: Example Predictions for Three Models for 5-Day Prediction Horizon.

The results indicate that traditional methods, when supplemented with relevant contextual features, outperform more complex models that do not incorporate such data. While the 1-day ahead predictions show comparable performance across all methods, as the prediction horizon extends, the accuracy of CHRONOS, PatchTST, and N-BEATS declines sharply. In contrast, the traditional models, supported by contextual features, maintain their predictive accuracy much more effectively, as shown in Figure 1.

A closer examination of the predictions in Figures 2 and 3 reveals that for 1-day ahead predictions, all models track the true data closely. However, in the 5-day ahead predictions, models lacking contextual data begin to exhibit a rightward shift in their predictions. This likely occurs due to the absence of contextual information, causing these models to lag in capturing the true trajectory of water levels. In contrast, models with access to weather data can predict further ahead by accounting for factors such as the impact of rainfall patterns on water levels.

An unexpected finding is that among the baseline models, linear regression outperforms the more sophisticated methods. For instance, in the article [9], while linear regression produced strong results, it did not surpass the performance of the other two methods.

5 Conclusion and Future Work

After evaluating all models on the groundwater level dataset, we observed that traditional methods, when equipped with relevant features, consistently outperformed newer and more sophisticated techniques, particularly as the prediction horizon lengthened. This suggests that the emphasis on developing the most powerful deep learning or foundation models for time series predictions may be overstated. With thoughtful selection of contextual features, even the simplest models can outperform modern approaches, which is a significant finding for fields with sufficient contextual data, such as data-driven environmental modelling.

To enhance the robustness of our evaluation, future work could involve testing additional methods, expanding the analysis to include more measurement stations and surface water level data, and incorporating deep learning models that support multivariate time series, such as N-BEATSx [11] and N-HiTS [3]. Further insights could be gained by exploring foundation models with multivariate support, such as TimesFM [4], as well as some more univariate models, like TimeGPT-1 [6]. Future research could also compare the inference times of various models and assess performance across different time series lengths.

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.

Furthermore, we would like to thank Erik Novak for his assistance in completing this research.

References

[1] Abdul Fatir Ansari et al. 2024. Chronos: learning the language of time series. *arXiv preprint arXiv:2403.07815*.

[2] ARSO. 2009. Freshwater. Retrieved August 27, 2024 from https://www.arso.gov.si/en/soer/freshwater.html.

[3] Cristian Challu, Kin G Olivares, Boris N Oreshkin, Federico Garza Ramirez, Max Mergenthaler Canseco, and Artur Dubrawski. 2023. NHiTs: neural hierarchical interpolation for time series forecasting. In *Proceedings of the AAAI conference on artificial intelligence* number 6. Vol. 37, 6989–6997.

[4] Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. 2023. A decoder-only foundation model for time-series forecasting. *arXiv preprint arXiv:2310.10688*.

[5] Fan Feng, Hamzeh Ghorbani, and Ahmed E. Radwan. 2024. Predicting groundwater level using traditional and deep machine learning algorithms. *Frontiers in Environmental Science*, 12. DOI: 10.3389/fenvs.2024.1291327.

[6] Azul Garza and Max Mergenthaler-Canseco. 2023. TimeGPT-1. *arXiv preprint arXiv:2310.03589*.

[7] Trevor Hastie, Robert Tibshirani, and Jerome H Friedman. 2009. *The elements of statistical learning: data mining, inference, and prediction*. Vol. 2. Springer.

[8] Klemen Kenda, Blaž Kažič, Erik Novak, and Dunja Mladenić. 2019. Streaming data fusion for the internet of things. *Sensors*, 19, 8. DOI: 10.3390/s19081955.

[9] Klemen Kenda, Jože Peternelj, Nikos Mellios, Dimitris Kofinas, Matej Čerin, and Jože Rožanec. 2020. Usage of statistical modeling techniques in surface and groundwater level prediction. *Journal of Water Supply: Research and Technology-Aqua*, 69, 3, (Apr. 2020), 248–265. DOI: 10.2166/aqua.2020.143.

[10] Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. 2022. A time series is worth 64 words: long-term forecasting with transformers. *arXiv preprint arXiv:2211.14730*.

[11] Kin G. Olivares, Cristian Challu, Grzegorz Marcjasz, Rafał Weron, and Artur Dubrawski. 2023. Neural basis expansion analysis with exogenous variables: forecasting electricity prices with nbeatsx. *International Journal of Forecasting*, 39, 2, 884–900. DOI: https://doi.org/10.1016/j.ijforecast.2022.03.001.

[12] Boris N Oreshkin, Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio. 2019. N-BEATS: neural basis expansion analysis for interpretable time series forecasting. *arXiv preprint arXiv:1905.10437*.

[13] Hongwei Ye et al. 2024. A transformer-based forecasting model for f10.7 index and its application study on the chinese langfang dataset. *Advances in Space Research*. DOI: https://doi.org/10.1016/j.asr.2024.08.024.

A Hyperparameters

Table 2: Hyperparameters Used for Gradient Boosting Regressor and Random Forest Regressor.

Hyperparameter	GradientBoosting	RandomForest
n_estimators	28	164
max_features	'log2'	0.5
max_depth	10	20

Table 3: Hyperparameters Used for N-BEATS and PatchTST.

Hyperparameter	N-BEATS	PatchTST
loss	HuberLoss	/
n_harmonics	5	/
n_polynomials	5	/
scaler_type	'robust'	/
n_blocks	[3, 3, 1]	/
mlp_units	[[128, 128]]	/
horizon	5	5
input_size	15	71
learning_rate	0.001	0.001
max_steps	25	1323
encoder_layers	/	12
n_heads	/	16
hidden_size	/	64
linear_hidden_size	/	512
dropout	/	0.2
fc_dropout	/	0.1
head_dropout	/	0.1
attn_dropout	/	0.2
patch_len	/	16
stride	/	8
revin	/	True

B Selected Features

Due to the large number of features selected by the feature selection algorithm, we provide a summarized description of the most frequently chosen features. The features that appeared most often include shifts and averages of precipitation, precipitation forecasts, temperature, altitude difference, cloud cover, humidity, and snow accumulation. Notably, the majority of selected features were derived features we generated, with only approximately one original feature being selected per model.

In Table 4, the most common shifts and averages for each individual model are presented. The table indicates that shifts and averages of varying lengths were selected, with a slight preference for shorter ones.

Table 4: Most Frequently Selected Shifts and Averages for Various Methods.

Method	Shifts (days)	Averages (days)
GradientBoostingRegressor	4, 10	2, 6
RandomForestRegressor	2, 6	3, 9
LinearRegression	2, 10	2, 7
Combined	**2, 10**	**2, 3**