

Time Series Forecasting Transformer and Prophet Model

Sibel YILMAZ

Fırat University / M.Sc

sibelyilmaz021@gmail.com

Orcid No: 0000-0002-0599-7263

Seçil YALAZ

Dicle University / Assoc.Prof.

syalaz@dicle.edu.tr

Orcid No: 0000-0001-7283-9225

Sinan ÇALIK

Fırat University / Prof.

scalik@firat.edu.tr

Orcid No: 0000-0002-4258-1662

Abstract

A time series is a sequence of temporal data obtained at specific intervals. These intervals can represent equal time periods such as daily, weekly, monthly, or yearly, or irregular and unequal time periods. Time series forecasting involves predicting future values of the target variable using its past values. This process involves identifying temporal variations such as trend, seasonality, and noise to make predictions about future values. In time series analysis, machine learning involves data analysis, preprocessing, normalization, transformations, time based features, error optimization, and model accuracy optimization. Machine learning methods provide greater efficiency compared to traditional methods, especially when the volume and complexity of data and variables increase. Deep learning, in particular, offers advantages in time series forecasting by capturing long-term dependencies and handling large and complex datasets. Machine learning and deep learning models can achieve high accuracy, interpretability, and fast results for multivariate time series forecasting.

This study focuses on multivariate time series forecasting by leveraging advanced machine learning models. Specifically, it employs the Prophet model and the Transformer architecture, an emerging deep learning method known for its ability to model long-term dependencies. Unlike traditional approaches, the proposed framework integrates both endogenous and exogenous variables, applies error-weighted variable importance calculations, and evaluates model performance through variable augmentation. By comparing and combining the Prophet and Transformer models, the study aims to enhance forecasting accuracy and interpretability for complex temporal datasets.

Keywords: Time Series Forecasting, Machine Learning, Deep Learning, Prophet Model, Transformer Model

Corresponding Author / Sorumlu Yazar: Seçil YALAZ, Dicle University, Statistics Department.

Citation / Atıf: YILMAZ S., YALAZ S., ÇALIK S. (2025). Time Series Forecasting Transformer and Prophet Model. *İstatistik Araştırma Dergisi*, 15 (1), 9-16.

Zaman Serileri Tahmini: Transformer ve Prophet Modeli

Özet

Zaman serisi, belirli aralıklarla elde edilen zamansal veri dizisidir. Bu aralıklar, günlük, haftalık, aylık veya yıllık gibi eşit zaman periyotlarını temsil edebileceği gibi, düzensiz ve eşit olmayan zaman periyotlarını da ifade edebilir. Zaman serisi tahmini, hedef değişkenin geçmiş değerlerini kullanarak gelecekteki değerlerini tahmin etmeyi içerir. Bu süreçte, eğilim (trend), mevsimsellik (seasonality) ve gürültü (noise) gibi zamansal değişimlerin belirlenmesi, gelecekteki değerler hakkında öngörülerde bulunmak için önemlidir. Zaman serisi analizinde makine öğrenmesi, veri analizi, ön işleme, normalizasyon, dönüşümler, zamana dayalı özellikler, hata optimizasyonu ve model doğruluğu optimizasyonunu içerir. Makine öğrenmesi yöntemleri, özellikle veri ve değişkenlerin hacmi ile karmaşıklığı arttığında, geleneksel yöntemlere kıyasla daha yüksek verimlilik sağlar. Özellikle derin öğrenme, uzun vadeli bağımlılıkları yakalayarak ve büyük, karmaşık veri kümelerini işleyerek zaman serisi tahmininde önemli avantajlar sunar. Makine öğrenmesi ve derin öğrenme modelleri, çok değişkenli zaman serisi tahmininde yüksek doğruluk, yorumlanabilirlik ve hızlı sonuçlar elde etmeye olanak tanır.

Bu çalışma, çok değişkenli zaman serisi tahmini üzerine odaklanarak gelişmiş makine öğrenmesi modellerinden Prophet ile uzun dönemli bağımlılıkları modellemedeki başarısıyla öne çıkan Transformer mimarisini kullanmaktadır. Geleneksel yöntemlerden farklı olarak, modelleme sürecine içsel (endojen) ve dışsal (eksojen) değişkenler dahil edilmiştir. Her bir değişken için hata ağırlıklı önem dereceleri hesaplanmış, değişken artırımı uygulanmış ve iki modelin performansı karşılaştırılmıştır. Prophet ve Transformer modellerinin birlikte kullanılmasıyla, karmaşık zaman serilerinde yüksek doğruluk ve yorumlanabilirlik hedeflenmiştir.

Anahtar sözcükler: Zaman serisi tahmini, Makine öğrenmesi, Derin öğrenme, Prophet modeli, Transformer modeli

1. Introduction

Forecasting is the process of making inferences, identifying relationships, or predicting future states based on existing data or information. The forecasting process is analyzed and modeled according to the knowledge of the data and its structure (Huang et al., 2022). A time series is a sequence of temporal data obtained at regular intervals. These data intervals can represent equal time intervals such as daily, weekly, monthly, or yearly, or irregular and unequal time intervals. Time series forecasting is a field of research widely used in many common areas such as business, economics, and energy. The process of time series forecasting involves predicting future values using past values of the dependent variable. It provides foresight by identifying temporal variations such as trends and seasonality in the process of predicting future values (Liu et al., 2023). Time series forecasting models are widely implemented using traditional, machine learning, and deep learning algorithms. High-accuracy prediction models are preferred for univariate or multivariate datasets (Sun et al., 2022).

This study includes multivariate time series models. The impact and accuracy of many endogenous and exogenous variables such as seasonal effects, trends, periodic differences, weather conditions, and macroeconomic data on the time series forecasting model have been addressed (Jha and Pande, 2021). Among the many forecasting models with multiple variables, the Prophet model offers several advantages over many machine learning models in terms of interpretability, obtaining and controlling seasonal variables, considering change intervals, and including variables as regressors (Riyantoko et al., 2021). The Transformer model, a deep neural network, provides ease in

long-term dependencies and produces high-accuracy predictions in time series forecasting by focusing on variables through the attention mechanism (Vaswani et al., 2017).

Unlike the existing literature, this study proposes a method for model prediction by incorporating the regressor effect into the multivariate Prophet model and adding trend and seasonality variables to the Transformer model. This method aims to boost accuracy, learning efficiency, generalization, and interpretability, leading to highly precise and reliable time series forecasts.

2. Time Series Forecasting Models

Time series are divided into two groups according to the types of variables. These are univariate and multivariate time series. Univariate time series contain a single variable dependent on time, while multivariate time series include multiple variables dependent on time (Riyantoko et al., 2021). In this study, the Prophet model used in multivariate time series models and the deep learning method known as the Transformer model are examined.

2.1 Prophet Model

The Prophet model is a time series forecasting model developed by Facebook as an open-source software in 2017. It enables the modeling of nonlinear daily, weekly, monthly, and yearly data, incorporating seasonality and the holiday effects specific to different countries. Seasonality represents the variations occurring within a time unit. The Prophet model offers advantages over other models by addressing missing data in the dataset and automatically correcting outliers to a certain extent.

The Prophet model is given as,

$$y_t = g(t) + s(t) + h(t) + \varepsilon_t. \quad (1)$$

The time series model given in Eq. (1), $g(t)$ represents the trend function modeling nonperiodic changes, $s(t)$ represents the time-dependent seasonal periodic changes, $h(t)$ denotes the holiday and country-specific effects component and ε_t signifies the error or noise component in the time series model (Taylor and Letham, 2018).

The Prophet model provides forecasts by constructing time series models with flexible structures that capture trend and seasonal patterns. It supports modeling with a univariate numerical variable (y) and a time column (d_s). Advanced hyperparameters and the multivariate advanced prophet model are employed for datasets involving multiple variables (Taylor and Letham, 2018).

2.2 Advanced Multivariate Prophet Model

The Prophet model, developed by Facebook, provides more accurate forecasts by incorporating components such as seasonality and holiday effects, along with flexibility, uncertainty, and the inclusion of external factors. In this study, to reduce uncertainty and errors and enhance reliability, several hyperparameter tuning and model capability procedures have been included, which are novel and additional to the literature. Some of the techniques used in this study include forecasting growth types, trend and seasonality changepoint detection, dataset-specific multiple regression additions, Markov Chain Monte Carlo (MCMC) parameter distributions and uncertainty detection parameters, and interval width (Hasnain et al., 2022).

2.2.1 Forecasting Growth Types

To describe long-term or short-term changes (trends) in a time series, linear and logistic growth models are used. The logistic growth model represents models characterized by curves that rapidly decrease as the variable approaches a certain saturation point. Linear growth, on the other hand, represents a model where the variable in the time series typically shows increasing or decreasing trends (Setianingrum et al., 2022).

2.2.1.1 Nonlinear saturating growth

The logistic growth model is also known as the nonlinear saturating growth trend. Nonlinear saturating growth trends represent situations where the increase in the series reaches certain limits or saturation. This growth type is a modeling approach where the rate of increase changes over time. The mathematical representation of the logistic growth model (Setianingrum et al., 2022) is given in Eq. (2) as:

$$g(t) = \frac{C}{1 + \exp(-k(t-m))} \quad (2)$$

In the logistic growth model, C represents the carrying capacity, k denotes the growth rate, and m is the shift parameter. In the Prophet model, the classic logistic growth model is extended to include a time-dependent carrying capacity. Constant C , becomes $C(t)$ signifying that it is dependant upon time, rather than a constant. The vector $a(t)$ represents binary outputs that signify which growth rate adjustments are valid for that point in time. It is an ‘‘on/off’’ switch for growth rate (and offset parameter) adjustments. It is transposed to allow for valid vector multiplication with the aforementioned vectors. The vector δ represents vector of growth rate adjustments, if we have 5 different linear growth rates, this will simply contain the 5 different values for growth rate adjustments (k gets adjusted by the value in δ). Multiplying by (transposed) $a(t)$ vector simply yields the correct growth rate adjustment for that point in time. The vector γ represents adjustments to the offset parameter (m). In the same way as δ vector, when multiplied by (transposed) $a(t)$ vector yields the correct growth rate adjustment, this yields the correct offset parameter adjustment to connect the endpoints of the linear ‘‘pieces’’ that form piecewise function (URL 1, 2021). The logistic growth model in the Prophet model (Setianingrum et al., 2022) is given in Eq. (3) as:

$$g(t) = \frac{C(t)}{1 + \exp(k + a(t)^T \delta)(t - (m + a(t)^T \gamma))} \quad (3)$$

2.2.1.2 Linear trend with changepoints

The linear trend with changepoints is used to model linear slopes in a time series and changes where a certain saturation point is not present. The nonlinear trend model addresses situations where the trend in the time series exhibits nonlinear changes under specific conditions. These situations are referred to as changepoints or breakpoints. The equation for the linear trend with changepoints (Huang et al., 2022) is given as:

$$g(t) = (k + a(t)^T \delta)(t - (m + a(t)^T \gamma)). \quad (4)$$

In the Eq. (4), $g(t)$ represents the time-dependent trend changepoint, k denotes the growth rate, δ is the adjustment rate, m is the offset parameter that shifts a time point to a specific time, and γ represents the trend changepoint. The changepoint parameter has become advantageous with the Prophet model. With the Prophet model, automatic changepoint selection can be computed and adjusted based on the dataset. Changepoints can be adjusted according to annual, monthly, or seasonal periods. The Prophet model can detect numerous changepoints that might cause changes in the rate. The changepoints (δ_j) are modeled using a Laplace distribution $(0, \tau)$ (Van den Burg and Williams, 2020):

$$\delta_j \sim \text{Laplace}(0, \tau), \quad (5)$$

where j represents the changepoint at time j . The Laplace $(0, \tau)$ distribution indicates the flexibility of the model at changepoints, where the variable has a mean of 0 and a scale parameter τ . As τ approaches 0, the model becomes more flexible and converges to either logistic or linear behavior (Van den Burg and Williams, 2020).

2.3 Transformers

Recurrent Neural Networks (RNNs), including Long Short-Term Memory (LSTM) networks, are among the most commonly used methods for sequential modeling and transformation tasks, such as language models (Feng et al., 2022). With the growth of data structures and the need for improved performance, transformer architectures based on attention mechanisms have emerged. Attention mechanisms have provided higher performance and parallelizability compared to RNN and LSTM models. Machine learning models often result in poor outcomes in cases requiring long-term dependencies and efficiency. The Transformer model, however, addresses these limitations through its attention mechanisms, offering a novel approach (Wu et al., 2020).

2.3.1 Transformer model architecture

The Transformer model, which consists of encoder and decoder structures, transforms an input sequence of symbols $(x_1, x_2, x_3, \dots, x_n)$ into a sequence of continuous symbols $z = (z_1, z_2, z_3, \dots, z_n)$. Using z , outputs $(y_1, y_2, y_3, \dots, y_m)$ are generated. The Transformer model uses the produced inputs as input to generate subsequent outputs, thus follows an autoregressive structure. The model architecture depicted in Figure (1) illustrates the Transformer's encoder-decoder structure. The encoder consists of a stack of layers, each containing a multi-head attention mechanism followed by a feed-forward network, both wrapped with residual connections and layer normalization (add and norm). The input embeddings are first passed into the encoder stack. The decoder mirrors the encoder's design but includes an additional masked multihead attention layer to ensure autoregressive property during training. The decoder receives the output embeddings and processes them through masked multihead attention, multihead attention with encoder outputs, and a feed-forward network, each with residual connections and normalization. Finally, the decoder's output is transformed through a linear layer followed by a softmax activation to generate the final predictions.

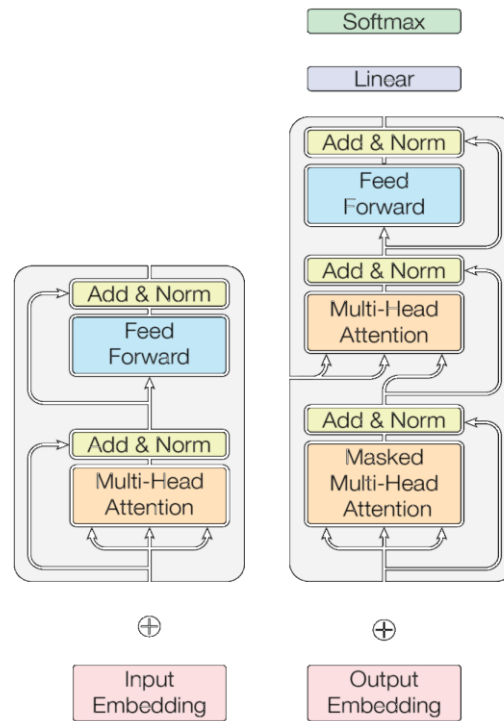


Figure 1. Transformer model architecture (Vaswani et al., 2017).

2.3.1.1 Encoder - Decoder stacks

Comparative neural networks, such as the Transformer model, have encoder and decoder structures. The encoder generates a hidden vector at each layer using the input sequence within the neural network structure. The encoder consists of $N = 6$ identical layers, each with two sub-layers. The first sub-layer is a multi-head self-attention mechanism, while the second sub-layer is a position-wise fully connected feedforward network. Residual connections are employed around the sub-layers, followed by layer normalization. This approach facilitates the training of deep networks and addresses the vanishing gradient problem. With residual connections, the input data is directly linked to the output data. Layer normalization ensures that activations are balanced across layers and reduces the risk of overfitting (Zhou et al., 2021).

The decoder stack receives the representation vectors from the encoder and generates the output sequence. It also consists of $N = 6$ identical layers. Unlike the encoder stack, the decoder includes a third sub-layer that implements multi-head attention on the encoder outputs. Additionally, the attention sub-layer in the decoder is designed to prevent and isolate attention to subsequent positions, ensuring that the model focuses only on previous positions. This mechanism is referred to as masking. Masking eliminates the influence of future positions, allowing the model to make predictions based solely on past data. This feature provides significant advantages in both time series and language models. Since the Transformer processes elements of the input sequence in parallel rather than sequentially, it uses positional encoding to represent each position and facilitate learning (Zhou et al., 2021).

2.3.1.2 Attention mechanisms

The attention function transforms a query vector and key-value pairs into an output vector. Each of these elements query, key, value, and output is a vector. The function calculates weights by determining the compatibility between the query and the key, aligning and weighting the query and key vectors accordingly.

Attention mechanisms are fundamental to the Transformer model. The attention function aligns a query with a key-value pair to generate an output. This output vector is computed as a weighted average of the value vectors, allowing the model to learn which value vectors are more significant and contribute more to the output, thereby facilitating information transfer (Mohammadi Farsani and Pazouki, 2020).

2.3.1.3 Positional encoding

In the Transformer model, since each token or input is processed in a nonsequential manner, its position within the sequence is represented using positional encoding, which allows the model to incorporate this positional information (Van den Burg and Williams, 2020).

3. Conclusion

Time series forecasting is a methodology used to predict future trends. Time series forecasting models, which are employed across various domains, often incorporate multiple external variables for prediction. In addition to traditional methods, performance evaluations are conducted using machine learning and deep learning models. Some of these methods, as discussed in this study, include the Prophet model and the Transformer model.

The Prophet model provides rapid, high-performance forecasts by capturing various effects such as seasonality, trends, holiday effects, weather conditions, and lag variables, and by incorporating external factors. The Transformer model leverages attention mechanisms to capture long-term temporal dependencies, facilitating learning. Integrating external variables into the Transformer model can yield highly accurate predictions. In conclusion, the Prophet and Transformer models offer advantages in speed and interpretability compared to traditional methods in time series forecasting. Transformer models provide detailed pattern recognition capabilities, which are beneficial for large datasets, compared to Prophet models.

This study aims to achieve high-accuracy performance, incorporating seasonal, trend, and tail variables obtained from the Prophet model into the Transformer model. This hybrid combination of the two models can significantly enhance time series forecasting performance. Here are the key benefits and outcomes of this integration:

1. The proposed integration helps to improve accuracy. The seasonal, trend, and holiday variables from the Prophet model provide rich contextual information that can help the Transformer model make more accurate predictions. This additional data allows the Transformer to better understand underlying patterns and anomalies in the time series data.
2. The Transformer model, known for its attention mechanisms, can effectively leverage the detailed patterns captured by the Prophet model. This synergy enables the Transformer to capture long-term dependencies and complex temporal relationships more efficiently. And this brings enhancement to the learning process.
3. By integrating the Prophet model's endogenous variables, the Transformer model can generalize better across different time series datasets, which means better generalization. This is because the Prophet model is excellent at capturing domain-specific effects like seasonality and holidays, which are crucial for making accurate predictions.

4. The additional variables from the Prophet model can help mitigate overfitting in the Transformer model. By providing more informative features, the model is less likely to rely on outlier in the data, leading to more robust predictions.

5. While the Transformer model is powerful, it can be computationally intensive. The Prophet model's ability to rapidly preprocess and capture essential time series components allows the Transformer model to focus on refining its predictions, thus improving both speed and interpretability.

6. The combination of the Prophet model's ability to handle various effects (seasonality, trends, holidays, lag variables, additional and multiplication variables) with the Transformer's capability to model complex temporal dependencies creates a comprehensive approach to time series forecasting. This holistic view can lead to superior performance compared to using either model independently.

In summary, a hybrid model will be achieved by combining the Prophet model and Transformer models. The Prophet model will be used to generate numerous endogenous and exogenous temporal, holiday, lagged, additive, and multiplicative variables for the dataset. The error associated with each variable will be assessed. Subsequently, these variables will be incorporated into Transformer models, which can effectively handle each variable using the attention mechanism and resolve complex dependencies. This approach aims to leverage the strengths of both models to enhance the overall forecasting performance.

Ethical Statement

The authors declare that this document does not require ethics committee approval or any special permission. Our study does not cause any harm to the environment and does not involve the use of animal or human subjects.

Acknowledgment

The authors thank Fırat University for providing the necessary facilities and resources to conduct the research.

Conflict of Interest

The authors declare that this document has not conflict of interest.

Authors' Contributions

All authors contributed to the study conception and design. Material preparation, data collection and analysis were performed by Sibel Yılmaz (%40). The first draft of the manuscript was written by Seçil Yalaz (%40) and Sinan Çalık (%20), and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

References

- Feng, T., Zheng, Z., Xu, J., Liu, M., Li, M., Jia, H., & Yu, X. (2022). The comparative analysis of SARIMA, Facebook Prophet, and LSTM for road traffic injury prediction in Northeast China. *Frontiers in Public Health*, 10, 946563. <https://doi.org/10.3389/fpubh.2022.946563>
- Hasnain, A., Sheng, Y., Hashmi, M. Z., Bhatti, U. A., Hussain, A., Hameed, M., & Zha, Y. (2022). Time series analysis and forecasting of air pollutants based on prophet forecasting model in Jiangsu province, China. *Frontiers in Environmental Science*, 10, 945628. <https://doi.org/10.3389/fenvs.2022.945628>
- Huang, Y. T., Bai, Y. L., Yu, Q. H., Ding, L., & Ma, Y. J. (2022). Application of a hybrid model based on the Prophet model, ICEEMDAN and multi-model optimization error correction in metal price prediction. *Resources Policy*, 79, 102969. <https://doi.org/10.1016/j.resourpol.2022.102969>
- Jha, B. K., & Pande, S. (2021). Time series forecasting model for supermarket sales using FB-prophet. In *2021 5th International Conference on Computing Methodologies and Communication (ICCMC)* (pp. 547–554). IEEE. <https://doi.org/10.1109/ICCMC51019.2021.9418205>
- Liu, W., Yu, X., Zhao, Q., Cheng, G., Hou, X., & He, S. (2023). Time series forecasting fusion network model based on prophet and improved LSTM. *Computational Materials Continuum*, 74(2), 3200–3219. <https://doi.org/10.32604/cmc.2023.030971>
- Mohammadi Farsani, R., & Pazouki, E. (2020). A transformer self-attention model for time series forecasting. *Journal of Electrical and Computer Engineering Innovations (JECEI)*, 9(1), 1–10. <https://doi.org/10.22061/JECEI.2020.6672.348>
- Riyantoko, P. A., Fahrudin, T. M., Hindrayani, K. M., & Muhaimin, A. (2021). Water availability forecasting using univariate and multivariate Prophet time series model for ACEA (European Automobile Manufacturers Association). *International Journal of Data Science, Engineering, and Analytics*, 1(2), 43–54. <https://doi.org/10.53894/ijdsea.v1i2.19>
- Setianingrum, A. H., Anggraini, N., & Ikram, M. F. D. (2022). Prophet model performance analysis for Jakarta air quality forecasting. In *2022 10th International Conference on Cyber and IT Service Management (CITSM)* (pp. 1–7). IEEE. <https://doi.org/10.1109/CITSM55714.2022.9910907>
- Sun, S., Zhou, H., Ji, J., & Liu, S. (2022). Research on residual value prediction of new energy second-hand cars based on prophet multivariate time series model. In *International Conference on Computer, Artificial Intelligence, and Control Engineering (CAICE 2022)* (Vol. 12288, pp. 146–151). SPIE. <https://doi.org/10.1117/12.2648570>
- Taylor, S. J., & Letham, B. (2018). Forecasting at scale. *The American Statistician*, 72(1), 37–45. <https://doi.org/10.1080/00031305.2017.1380080>
- URL1 (Advancing Analytics). (2021). *Facebook Prophet and the Stock Market (Part 2)*. <https://www.advancinganalytics.co.uk/blog/2021/7/26/facebook-prophet-and-the-stock-market-part-2>
- Van den Burg, G. J., & Williams, C. K. (2020). An evaluation of change point detection algorithms. *arXiv Preprint*, arXiv:2003.06222. <https://arxiv.org/abs/2003.06222>
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N. Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 5998–6008.
- Wu, S., Xiao, X., Ding, Q., Zhao, P., Wei, Y., & Huang, J. (2020). Adversarial sparse transformer for time series forecasting. *Advances in Neural Information Processing Systems*, 33, 17105–17115.
- Zhou, T., Pan, S., Wang, J., Vasilakos, A. V., & Liu, H. (2021). Edge intelligence: Paving the last mile of artificial intelligence with edge computing. *Proceedings of the IEEE*, 107(8), 1738–1762. <https://doi.org/10.1109/JPROC.2019.2958030>