

A Comparative Analysis of Time Series Forecasting Models for Predicting Stock Prices in the Vietnamese Real Estate Sector

Nhung Tran Thi Hong

University of Information Technology
Ho Chi Minh City, Viet Nam
21522438@gm.uit.edu.vn

Quynh Do Mai Nhu

University of Information Technology
Ho Chi Minh City, Viet Nam
21520429@gm.uit.edu.vn

Thuc Nguyen Huy

University of Information Technology
Ho Chi Minh City, Viet Nam
21521505@gm.uit.edu.vn

Thang Hoang Manh

University of Information Technology
Ho Chi Minh City, Viet Nam
21521428@gm.uit.edu.vn

Dat Nguyen Duy

University of Information Technology
Ho Chi Minh City, Viet Nam
21521936@gm.uit.edu.vn

Abstract—Vietnam’s real estate market serves as a critical pillar of the nation’s economy. Understanding stock price movements of leading companies within this sector is therefore crucial for informed investment decisions. This study applies several time series analysis techniques to explore forecasting models, including Linear Regression, ARIMA, RNN, GRU, LSTM, VARMA, Fuzzy time series, XGBoost, FEDformer, and DLinear. By training forecasting models on historical stock price data, this study compares their effectiveness in predicting the stock prices of Vietnamese real estate corporations. The evaluation will utilize three key metrics: Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE).

Index Terms—Stock prices prediction, Linear Regression, ARIMA, RNN, GRU, LSTM, VARMA, Fuzzy time series, XGBoost, FEDformer, DLinear.

I. INTRODUCTION

Vietnam’s real estate market has experienced phenomenal growth in recent years. Fueled by robust economic growth and favorable investment conditions, the sector has attracted substantial foreign investment. The stock market offers an alternative approach to participate in the real estate market’s growth. Through shares of publicly traded real estate companies, investors can gain exposure to the sector without the complexities of direct property ownership.

Time series is a method which is used for the prediction of share prices. Time series analysis deals with analyzing a series of data gathered during time, which tries to forecast future by assuming that the previously observed patterns can be considered as the foundation to extract future behavior.

Stock market analysis is a challenging domain, characterized by a complex multivariate and time-evolving nature, with high volatility, and multiple correlations with exogenous factors. This study focuses on understanding and comparing the performance of several models for forecasting Vietnamese real estate stock prices using time series analysis techniques,

including Linear Regression, ARIMA, RNN, GRU, LSTM, VARMA, Fuzzy time series, XGBoost, FEDformer and DLinear.

To provide a comprehensive comparison, this study will use historical stock price data starting from March 1st 2017 of three prominent Vietnamese real estate corporations: NVL (Novaland Group), KDH (Khang Dien House) and DXG (Dat Xanh Group).

II. RELATED WORKS

Recently, there has been a significant surge in research focusing on forecasting stock prices through the application of diverse machine learning, deep learning and statistical models.

C.Ebenesh et al. [1] attempted to develop two models, Linear Regression and LSTM. Proposed framework has been tested for the prediction of three stocks and it has shown that LR model depicted promising performance compared to the LSTM model.

J. A. Rusman et al. [2] implemented Machine Learning and Deep Learning models such as RNN models (LSTM, Bi-LSTM, GRU, and Multi-LSTM) and ARIMA models. The results show that LSTM and ARIMA stood out within other models.

C. James et al. [3] investigated simultaneously the relations among stock returns, real activity, inflation, and money supply changes using a vector autoregressive moving average (VARMA) model.

K. Senthamarai Kannan et al. [4] compared Fuzzy time series and ARIMA model when used to predict fuel prices in India. It is concluded that fuzzy time series model gives better results than other time series model.

Y. Zhang et al. [5] applied XGBoost algorithm model to forecast stock prices based on the daily time series characteristics of stocks. The results demonstrate that the model is able to capture the high-frequency time series fluctuation trend of

the stock and obtain more accurate prediction results about the stock price.

T. Zhou et al. [6] proposed the combination of Transformer with the seasonal-trend decomposition method. They also developed a frequency enhanced Transformer termed as Frequency Enhanced Decomposed Transformer (FEDformer).

L. Jialin et al. [7] proposed a hybrid model (HI-IVMD-DLinear) comprising the Hampel identifier (HI), the improved variational mode decomposition (IVMD) optimized by gray wolf optimization (GWO), and DLinear for wind speed forecasting (WSF). The experimental results reveal the model is suited to WSF and DLinear has great prediction performance.

III. MATERIALS

A. Datasets

This study examines the historical stock prices of three leading real estate companies in Vietnam: Novaland Group (NVL), Khang Dien House (KDH), and Dat Xanh Group (DXG), taken from Investing.com. The data will cover the period from March 1, 2017, to March 20, 2024.

Table I describes the columns in the chosen datasets.

TABLE I
DATASETS DESCRIPTION

Column	Description
Date	The date of the data point
Price	The closing price of the stocks on the given date
Open	The opening price of the stocks on the given date
High	The highest price reached by the stocks during the trading session on the given date
Low	The lowest price reached by the stocks during the trading session on the given date
Vol.	The trading volume for the day, denoted in million of shares traded
Change %	The percentage change in the closing price compared to the previous day's closing price

Since this study explores datasets containing historical closing prices, only descriptive statistical data relating to column *Price* will be listed.

B. Descriptive statistics

TABLE II
NVL, KDH, DXG'S DESCRIPTIVE STATISTICS

	NVL	KDH	DXG
Count	1800	1800	1800
Mean	39806.3506	24523.0734	16803.1879
Median	32738.5	20300	15391.4
Mode	34491	12910.1	18950
Std	22351.3231	9700.0272	7150.8501
Min	10250	8944.8	6066.5
25%	26584	17752	11735.825
50%	32738.5	20300	15391.4
75%	42752	31200	19965.9
Max	92366	51636	46750
Kurtosis	-0.3526	-0.3415	2.9124
Skewness	0.9943	0.7184	1.5023

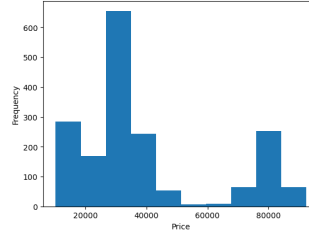


Fig. 1. NVL stock price's histogram

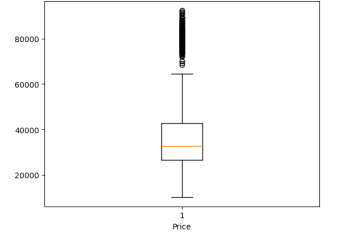


Fig. 2. NVL stock price's boxplot

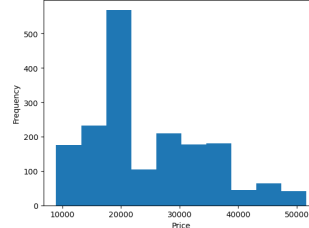


Fig. 3. KDH stock price's histogram

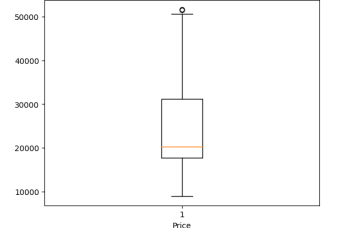


Fig. 4. KDH stock price's boxplot

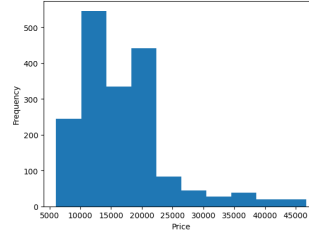


Fig. 5. DXG stock price's histogram

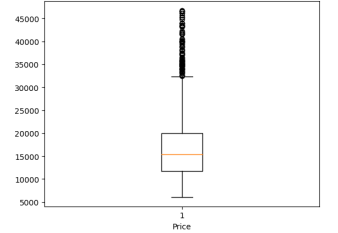


Fig. 6. DXG stock price's boxplot

C. Data split ratio

It was emphatically put that the size of datasets would be the determinant of split ratios that best fit the datasets so as to improve the model performance. Thus, determining the optimal allocation ratio between training and test sets is crucial. A commonly used ratio is 80:20, which means 80% of the data is for training and 20% for testing. Other ratios such as 70:30, 60:40, and even 50:50 are also used in practice. There does not seem to be clear guidance on what ratio is best or optimal for a given dataset [8].

Therefore, in this study, we decided to split the dataset into the ratios of 70:30, 80:20 and 90:10. Ultimately, the choice of ratio will depend on the specific characteristics of the dataset and the goals of the analysis.

D. Model evaluation metrics

To assess the performance of the different forecasting models applied, we will utilize three key evaluation metrics: Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE).

Root Mean Squared Error (RMSE): This metric calculates the square root of the average squared difference between the predicted and actual values.

$$RSME = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (1)$$

Mean Absolute Error (MAE): This metric calculates the average of the absolute differences between predicted and actual values.

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (2)$$

Mean Absolute Percentage Error (MAPE): This metric calculates the average error as a percentage relative to the actual values.

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right| \times 100\% \quad (3)$$

Where:

- y_i is the actual value for the i -th data point.
- \hat{y}_i is the predicted value for the i -th data point.
- n is the total number of data points.

IV. METHODOLOGY

A. Linear Regression

Linear Regression is a statistical analysis method that characterizes linear relationships between a dependent variable and one or more independent variables. Eq. (4) describes the general form of a multivariable linear regression function.

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_k X_k + \epsilon \quad (4)$$

where:

- Y is the dependent variable.
- X_1, X_2, \dots, X_k are the independent variables.
- β_0 is the intercept term.
- $\beta_1, \beta_2, \dots, \beta_k$ are the regression coefficients for the independent variables.
- ϵ is the error term.

B. ARIMA

Autoregressive integrated moving average (ARIMA) is a popular statistical method used for time series forecasting. An ARIMA model includes the following components: Auto Regressive (AR), Integration (I) and Moving Average (MA).

An ARIMA model is defined by three parameters:

- p , which is the order of the Auto Regressive part (AR), indicates the number of lagged observations.
- d , which is the degree of differencing or the number of times the data have had past values subtracted (to achieve stationarity).
- d is the order of the Moving Average part (MA), represents the number of lagged forecast errors.

The mathematical formula for an ARIMA(p, d, q) model can be represented as Eq. (5).

$$\hat{y}_t = \mu + \phi_1 y_{t-1} + \dots + \phi_p y_{t-p} - \theta_1 e_{t-1} - \dots - \theta_q e_{t-q} \quad (5)$$

Where:

- μ is the constant term.
- ϕ_1, \dots, ϕ_p are the coefficients of the AR part.
- y_{t-1}, \dots, y_{t-p} are the previous observed values.
- $\theta_1, \dots, \theta_q$ are the coefficients of the MA part.
- e_{t-1}, \dots, e_{t-q} are the residual terms.

C. RNN

Recurrent Neural Networks (RNN) is a type of Neural Network designed to work with sequential data like time series data and text data. RNN is different to other Neural Networks because it is capable of remembering information from previous steps with the help of hidden states.

RNN consists of an input layer, a hidden layer, and an output layer like other Neural Networks. But a hidden layer of RNN has neurons that take input not only from the input layer but also from the previous state due to an internal memory.

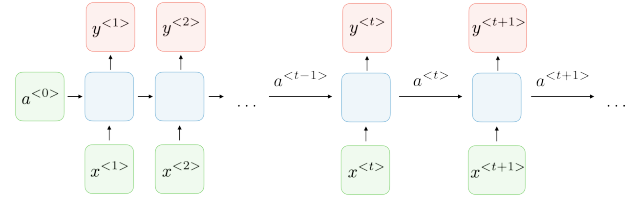


Fig. 7. RNN architecture

For each timestep t , the activation $a^{<t>}$ and the output $y^{<t>}$ are expressed as Eq. (6) and Eq. (7), respectively.

$$a^{<t>} = g_1(W_{aa}a^{<t-1>} + W_{ax}x^{<t>} + b_a) \quad (6)$$

$$y^{<t>} = g_2(W_{ya}a^{<t>} + b_y) \quad (7)$$

Where:

- $W_{ax}, W_{aa}, W_{ya}, b_a, b_y$ are coefficients that are shared temporally.
- g_1, g_2 are activation functions.

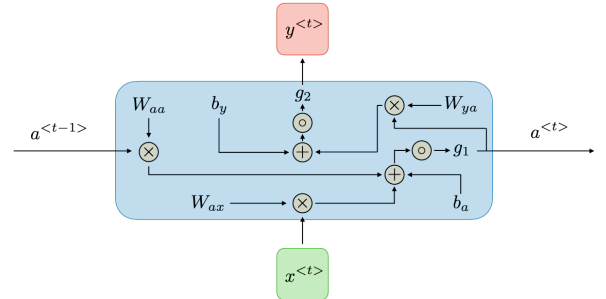


Fig. 8. RNN block description

D. GRU

A gated recurrent unit (GRU) was proposed by Cho et al. [9] to make each recurrent unit to adaptively capture dependencies of different time scales. Similarly to the LSTM unit, the GRU has gating units that modulate the flow of information inside the unit, however, without having a separate memory cells [10]. The structure of GRU is illustrated in Fig. 9

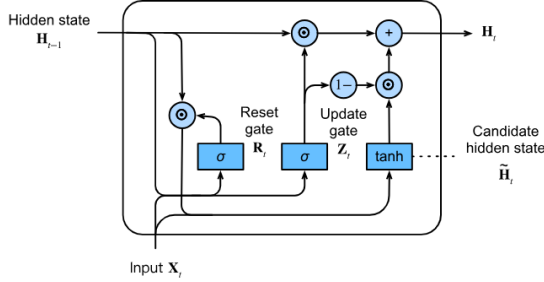


Fig. 9. The structure of GRU

The activation h_t^i of the GRU at time t is a linear interpolation between the previous activation h_{t-1}^i and the candidate activation \tilde{h}_t^j , written as Eq. (8).

$$h_t^j = (1 - z_t^j)h_{t-1}^j + z_t^j\tilde{h}_t^j \quad (8)$$

where an update gate z_t^j , written as Eq. (9) decides how much the unit updates its activation, or content.

$$z_t^j = \sigma(W_z x_t + U_z h_{t-1})^j \quad (9)$$

The candidate activation \tilde{h}_t^j is computed as Eq. (10).

$$\tilde{h}_t^j = \tanh(W x_t + U(r_t \odot h_{t-1}))^j \quad (10)$$

where r_t is a set of reset gates and \odot is an element-wise multiplication.

The reset gate r_t^j is computed similarly to the update gate, described in Eq. (11).

$$r_t^j = \sigma(W_r x_t + U_r h_{t-1})^j \quad (11)$$

E. LSTM

The Long Short-Term Memory (LSTM) model is a recurrent neural network (RNN) which was designed as a solution to the dissatisfaction with RNN's ability to capture long-term dependencies in sequential data.

The key feature of the LSTM model is the use of memory cells that can maintain information over long periods of time, allowing the network to learn and remember context from earlier in the sequence. LSTMs are composed of layers of memory blocks, each of which contains: cell state, forget gate, input gate, output gate. The structure of an LSTM model is illustrated in Fig. 10.

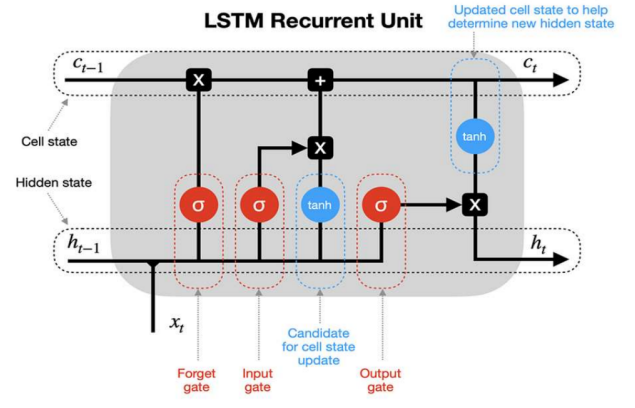


Fig. 10. LSTM architecture

Mathematically, the nodes can be represented as:

$$i_g = \delta(W_{xi}X_{tt} + W_{hi}h_{tt-1} + W_{ci}C_{tt-1} + b_i) \quad (12)$$

$$f_g = \delta(W_{xf}X_{tt} + W_{hf}h_{tt-1} + W_{cf}C_{tt-1} + b_f) \quad (13)$$

$$c_g = f_t c_{t-1} + i_t \tanh(W_{xc}X_{tt} + W_{hc}h_{tt-1} + b_c) \quad (14)$$

$$o_g = \delta(W_{xo}X_{tt} + W_{ho}h_{tt-1} + W_{co}c_t + b_o) \quad (15)$$

$$h_t = o_g \tanh(c_t) \quad (16)$$

Eqs. (12) - (16) show the formation of various nodes, here i_g , f_g and o_g being the input gate, forget gate and output gate respectively. The input of the LSTM is twofold process, where the current sample X_{tt} is the previous hidden layer sample, h_{tt-1} being the cell state and c_{t-1} gives the internal source of each gate. While the activation function is given as σ and \tanh . W_{xf} is the weight matrix for signal a and f [11].

F. VARMA

The Vector Autoregression Moving-Average (VARMA) is a type of time series model used in econometrics and statistics to analyze and forecast the behavior of multiple time series variables.

1) VAR model:

A Vector Autoregression or VAR model of order p (denoted as $VAR(p)$) is a system of equations where each variable in the system is regressed on its own lagged values and the lagged values of all other variables in the system. Eq. (17) describes the VAR model of k variables.

$$Y_t = c + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \dots + \phi_p Y_{t-p} + \epsilon_t \quad (17)$$

Where:

- Y_t is a $k \times 1$ vector of variables at time t .
- c is a $k \times 1$ vector of constants.
- $\phi_1, \phi_2, \dots, \phi_k$ are $k \times k$ coefficient matrices.
- ϵ_t is a $k \times 1$ vector of error terms, assumed to be multivariate normally distributed with mean zero and covariance matrix Σ .

2) MA model:

A moving-average model of order q (denoted as $MA(q)$) is a model that regresses the variable on past forecast errors. Eq. (18) describes the MA model of k variables.

$$Y_t = \mu + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + \dots + \theta_p \epsilon_{t-p} + a_t \quad (18)$$

Where:

- μ is a $k \times 1$ vector of means.
- $\theta_1, \theta_2, \dots, \theta_k$ are $k \times k$ coefficient matrices.
- $\epsilon_{t-1}, \epsilon_{t-2}, \dots, \epsilon_{t-q}$ are the lagged forecast errors.
- a_t is a $k \times 1$ vector of residuals, assumed to be multivariate normally distributed with mean zero and covariance matrix Σ .

3) Combining VAR and MA into VARMA model:

A $VARMA(p, q)$ model is a combination of the $VAR(p)$ and $MA(q)$ models. Eq. (19) expresses a VARMA model.

$$Y_t = c + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \dots + \phi_p Y_{t-p} + \epsilon_t + \mu + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + \dots + \theta_p \epsilon_{t-p} + a_t \quad (19)$$

Estimating the parameters (coefficient matrices $\phi_1, \phi_2, \phi_p, \theta_1, \theta_2, \theta_q$) of a VARMA model is typically done using maximum likelihood estimation or other methods like least squares.

G. Fuzzy time series

Fuzzy prediction for time series is an approach that applies the theory of fuzzy logic to make predictions about future data points in a time series. This approach is particularly useful when dealing with data that is uncertain, imprecise, or subject to vagueness.

Fuzzy time series models have been developed along two main directions: *a)* Fuzzification of raw data to establish its association, followed by the application of a known model for future forecasting, and *(b)* Direct construction of a forecasting model for the future. Compared to *(b)*, the research direction *(a)* has garnered more interest from statisticians.

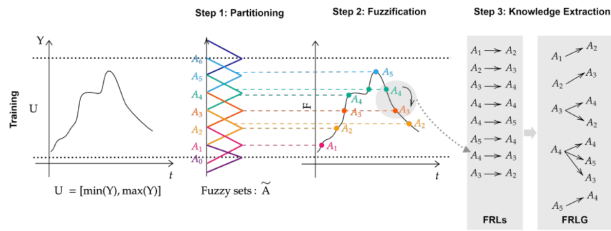


Fig. 11. Training procedure of fuzzy time series model

The training phase, illustrated in Fig. 11, consists of stages 1 to 3 described below.

Step 1 - Universe of Discourse partitioning: First, define the universe of discourse. The universe of discourse U is defined as follows: take the maximum value f_{max} and the minimum value f_{min} of the time series, and $U = [f_{max}, f_{min}]$. Sometimes, this range can be extended by adding an additional

value for ease of computation. Then, divide the interval U into m equal sub-intervals u_1, u_2, \dots, u_m .

Step 2 - Fuzzification: Determine the fuzzy sets A_i and fuzzify the values. Each A_i set is assigned a linguistic variable and determined on the identified segments u_1, u_2, \dots, u_m . Then the fuzzy sets A can be represented as Eq. (20):

$$A_i = \frac{\mu_{A_i}(u_1)}{u_1} + \frac{\mu_{A_i}(u_2)}{u_2} + \dots + \frac{\mu_{A_i}(u_m)}{u_m} \quad (20)$$

Step 3 - Knowledge or Temporal Pattern Extraction: Establish fuzzy relationships and group them. As defined above, for a fuzzy time series, fuzzy relationships can be identified at each time t , thereby determining groups of fuzzy relationships.

For the forecasting procedure, it is illustrated in Fig. 12.

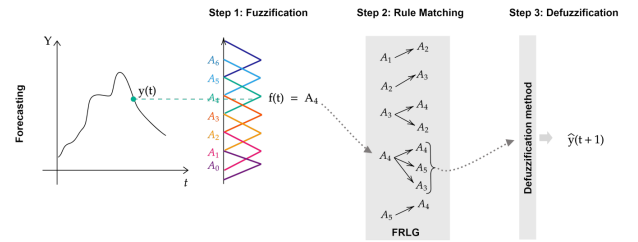


Fig. 12. Forecasting procedure of fuzzy time series model

Step 1 - Fuzzification: the fuzzification step is the same for both forecasting process and training procedure. It means that it translates the numeric sample $y(t) \in Y$ into a fuzzy value $f(t)$, where $f(t) \in \tilde{A}$.

Step 2 - Rule Matching: find the rules that contain the fuzzy set $f(t)$ in its precedent. The activation of the rules are given by the membership values in $f(t)$.

Step 3 - Defuzzification: the aim of this process is transforming the fuzzy value $f(t+1)$ to a crisp value $\hat{y}(t+1)$. In additions point forecasting methods, other types of defuzzification have been proposed for intervals and probabilistic distributions forecasting in the literature, see for instance.

H. XGBoost

Extreme Gradient Boosting (XGBoost) is a machine learning tool based on massively parallel Gradient Boosting (GB) as an enhanced version of the GB method, which is an algorithm based on residual optimization designed to achieve high efficiency, flexibility and portability.

XGBoost provides parallel boosting trees and establishes K regression trees, so that the predicted value of the tree group is close to the true value. It has strong generalization ability, and can quickly and accurately solve many scientific data problems. XGBoost is an improved algorithm of Gradient-boosted decision trees, and its core is the value of optimizing the objective function [12].

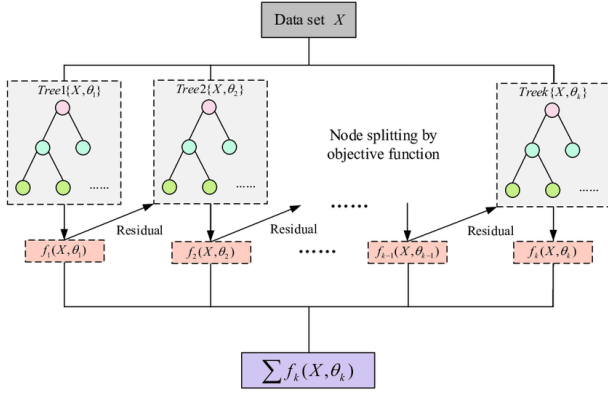


Fig. 13. XGBoost architecture

The objective function of XGBoost is typically formulated as a sum of a differentiable loss function $l(y_i, \hat{y}_i)$ and a regularization term $\Omega(f)$ to prevent overfitting. The goal is to minimize this objective function. The objective function can be written as Eq. (21).

$$Obj = \sum_{i=1}^n l(y_i, \hat{y}_i) + \sum_{k=1}^K \Omega(f_k) \quad (21)$$

Where:

- $l(y_i, \hat{y}_i)$ is the training error of sample x_i .
- $\Omega(f_k)$ is the regular term of the first tree.
- K is the total number of trees.
- f_k represents the k first trees.
- \hat{y}_i is the prediction result of sample x_i .

I. FEDFormer

The FEDformer, also known as *Frequency Enhanced Decomposed Transformer*, is a method for short-term and long-term time series prediction. It combines the Transformer model with a seasonal decomposition approach to address the high computational and memory requirements of Transformers for long time series.

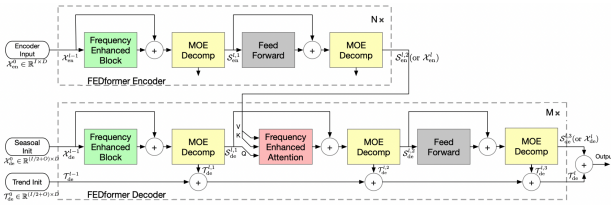


Fig. 14. Overall structure of FEDFormer

Structure (illustrated in Fig. 14): FEDFormer comprises of N encoders and M decoders. It integrates Frequency Enhanced Blocks (FEB) and Frequency Enhanced Attention (FEA) for representation learning in the frequency domain. These blocks come in two versions: FEB/FEA with Fourier basis (-f) and FEB/FEA with Wavelet basis (-w). The Mixture

Of Expert Decomposition Blocks (MOEDecom) are used to extract seasonal-trend patterns from the input data.

Mixture of Experts for Seasonal-Trend Decomposition: It contains a set of average filters with different sizes to extract multiple trend components from the input signal and a set of data-dependent weights for combining them as the final trend [6]. Its formula can be written as Eq. (22).

$$X_{trend} = Softmax(L(x)) \times (F(x)) \quad (22)$$

Where:

- $F(\cdot)$ is a set of average pooling filters.
- $Softmax(L(x))$ is the weights for mixing these extracted trends.

J. DLinear

DLinear is a novel high-precision time-series forecasting model proposed by A. Zeng et al. [13] in 2022. Despite its simple structure (illustrated in Fig. 15), consisting solely of a decomposition scheme and two linear networks, its predictive accuracy exceeds that of the transformer model. It first decomposes a raw data input into a trend component X_t by a moving average kernel and a remainder component $X_s = X - X_t$. Subsequently, two single-layer linear networks are utilized to forecast each of these decomposed components. The output results of the two single-layer linear networks are combined to yield the final predicted outcome \hat{X} .

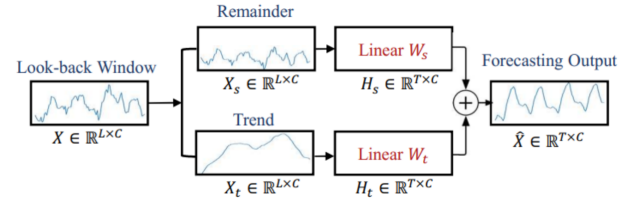


Fig. 15. The whole structure of DLinear

The whole process is $\hat{X} = H_s + H_t$, where $H_s = W_s X_s$ and $H_t = W_t X_t$ are the decomposed trend and remainder figures. $W_t \in \mathbb{R}^{T \times L}$ and $W_s \in \mathbb{R}^{T \times L}$ are two linear layers, as illustrated in Fig. 16.

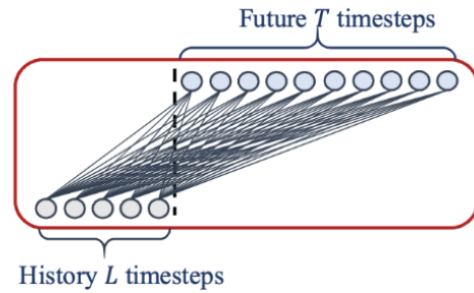


Fig. 16. One linear layer

V. RESULTS

A. NVL dataset

B. KDH dataset

C. DXG dataset

VI. PREDICTIONS

VII. CONCLUSION

A. Summary

B. Future considerations

ACKNOWLEDGMENT

REFERENCES

- [1] C. Ebenesh and K. Anitha, "A Novel Approach to Minimize the Mean Square Error in Predicting Stock Price Index using Linear Regression in Comparison with LSTM Model," in *2022 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS)*, Erode, India, 2022, pp. 1365-1370. doi: 10.1109/ICSCDS53736.2022.9760764.
- [2] J. A. Rusman, K. Chunady, S. T. Makmud, K. E. Setiawan, and M. F. Hasani, "Crude Oil Price Forecasting: A Comparative Analysis of ARIMA, GRU, and LSTM Models," in *2023 IEEE 9th International Conference on Computing, Engineering and Design (ICCED)*, Kuala Lumpur, Malaysia, 2023, pp. 1-6. doi: 10.1109/ICCED60214.2023.10425576.
- [3] C. James, S. Koreisha, and M. Partch, "A VARMA Analysis of the Causal Relations Among Stock Returns, Real Output, and Nominal Interest Rates," *The Journal of Finance*, vol. 40, pp. 1375-1384, 2019. doi: 10.1111/j.1540-6261.1985.tb02389.x.
- [4] K. Senthamarai Kannan, M. SulaigaBeevi, and S. Syed Ali Fathima, "A Comparison of Fuzzy Time Series and ARIMA Model," *International Journal of Scientific & Technology Research*, 2019, vol. 8, no. 8, pp. 1872-1876.
- [5] Y. Zhang, "Stock Price Prediction Method Based on XGboost Algorithm," in *Proceedings of the 2022 International Conference on Bigdata Blockchain and Economy Management (ICBBEM 2022)*, 2022, pp. 595-603. doi: 10.2991/978-94-6463-030-5_60.
- [6] T. Zhou, Z. Ma, Q. Wen, X. Wang, L. Sun, R. Jin, and R. "FEDformer: Frequency Enhanced Decomposed Transformer for Long-term Series Forecasting," in *Proceedings of the 39th International Conference on Machine Learning*, 2022, PMLR 162:27268-27286.
- [7] L. Jialin, C. Gong, S. Chen, and N. Zhou. 2023. "Multi-Step-Ahead Wind Speed Forecast Method Based on Outlier Correction, Optimized Decomposition, and DLinear Model", *Mathematics* 11, no. 12: 2746. doi: 10.3390/math11122746.
- [8] V. R. Joseph, "Optimal ratio for data splitting", *Statistical Analysis and Data Mining: The ASA Data Science Journal*. 15 (2022), pp. 531-538. <https://doi.org/10.1002/sam.11583>.
- [9] K. Cho, B. van Merriënboer, D. Bahdanau, and Y. Bengio. "On the properties of neural machine translation: Encoder-decoder approaches", 2014. *arXiv preprint arXiv:1409.1259*.
- [10] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling", 2014. *arXiv preprint arXiv:1412.3555*.
- [11] . "Gradient Boosting and LSTM Based Hybrid Ensemble Learning for Two Step Prediction of Stock Market". *Journal of Advances in Information Technology*, 2023, vol. 14, no. 6. doi: 10.12720/jait.14.6.1254-1260.
- [12] T. Liwei, F. Li, S. Yu, and G. Yuankai. "Forecast of LSTM-XGBoost in Stock Price Based on Bayesian Optimization," *Intelligent Automation & Soft Computing*, 2021, vol. 29, no. 3, pp. 855-868. doi: /10.32604/iasc.2021.016805.
- [13] A. Zeng, M. Chen, L. Zhang, and Q. Xu. "Are Transformers Effective for Time Series Forecasting?", 2022. *arXiv preprint arXiv:2205.13504*.