University of Information Technology

Faculty of Information System

Predictive Modeling of Vietnamese Bank Stock Prices: Integrating Machine Learning and Statistical Approaches for Enhanced Forecasting

NGUYEN TOAN KHANG[1], LY TUAN KHOA[2], AND LUU MINH CHU[3]

[1]University of Information Technology Ho Chi Minh City, Vietnam(e-mail: 21522195@gm.uit.edu.vn)
[2]University of Information Technology Ho Chi Minh City, Vietnam(e-mail: 21522225@gm.uit.edu.vn)
[3]University of Information Technology Ho Chi Minh City, Vietnam(e-mail: 21520652@gm.uit.edu.vn)

ABSTRACT In response to the growing interest of young investors in Vietnamese bank stocks, this research focuses on optimizing profits through advanced forecasting techniques. Leveraging a comprehensive suite of statistical and machine learning algorithms, including Autoregressive Integrated Moving Average (ARIMA), Support Vector Regression (SVR), Long Short-Term Memory (LSTM), Linear Regression (LN), Seasonal Autoregressive Integrated Moving Average with Exogenous Variables (SARIMAX), Extreme Gradient Boosting (XGBoost), Graph Neural Networks (GNN) and Fully convolutional network (FCN), the study aims to predict time series stock prices. The analysis utilizes evaluation metrics, including Root Mean Square Error (RMSE), Mean Absolute Percentage Error (MAPE%), and Mean Absolute Error (MAE%), Mean Squared Logarithmic Error (MSLE) to rigorously assess the performance of each forecasting model across diverse datasets. The research outcomes highlight the distinctive strengths of ARIMA, SVR, LSTM, Linear Regression, SARIMAX, XGBoost, and GNN, emphasizing their efficacy in achieving superior forecasting accuracy in the context of Vietnamese bank stock prices.

INDEX TERMS Stock Price Forecasting, Time Series Analysis, Machine Learning, Statistical Models, Autoregressive Integrated Moving Average, ARIMA, Support Vector Regression, SVR, Long Short-Term Memory, LSTM, Linear Regression, LN, Autoregressive Integrated Moving Average with Exogenous Variables, SARIMAX, Extreme Gradient Boosting, XGBoost, Graph Neural Networks, GNN, Fully convolutional network, FCN, Evaluation Metrics, Root Mean Square Error, RMSE, Mean Absolute Percentage Error, MAPE, Mean Absolute Error, MAE, Mean Squared Logarithmic Error, MSLE, Vietnamese Bank Stocks, Financial Forecasting, Investment Strategies, Financial Market Trends.

I. INTRODUCTION

In recent years, the securities investment landscape has seen a surge in interest, especially among young investors. This demographic shift has spurred research efforts to help investors optimize profits in dynamic financial markets. Our focus is on Vietnamese bank stocks, a sector gaining attention for lucrative opportunities.

Online trading platforms and social networks have democratized market access, attracting a new generation of investors. Vietnamese bank stocks, crucial to the nation's economy, are particularly appealing. Our study addresses profit maximization through advanced forecasting techniques.

We employ a holistic approach, using statistical and machine learning algorithms like ARIMA, SVR, LSTM, Linear Regression, SARIMAX, XGBoost, GNN and FCN to predict time series stock prices. Rigorous evaluation using metrics like RMSE, MAPE%, and MAE%, MSLE assesses model efficacy.

Our findings highlight strengths of forecasting models, emphasizing ARIMA, SVR, LSTM, Linear Regression, SARIMAX, XGBoost, GNN and FCN. By delineating their unique contributions, our study offers insights for investors and analysts, enhancing forecasting accuracy in the Vietnamese bank stock market. This contributes to the broader discussion on financial forecasting, investment strategies, and identifying trends in the intricate financial market landscape.

II. RELATED WORKS

A. LINEAR REGRESSION (LN)

Sonali Antad and et al. [1] from the Vishwakarma Institute of Technology, Pune, present a project focused on developing a stock price prediction website using the linear regression algorithm as a machine learning tool. Recognizing the importance of accurate stock market forecasting, the authors introduce the "J3 predictor" website, designed to predict stock prices over various durations. The methodology involves implementing linear regression in Python, specifically utilizing the scikit-learn library and the Django framework. The authors emphasize the simplicity and effectiveness of the linear regression model, achieving an accuracy rate between 75% to 85%. The future scope of the project suggests improvements in prediction accuracy, integration of natural language processing, forecasting the impact of non-financial events, and examining the influence of climate change on the stock market. In conclusion, the paper underscores the significance of accurate stock market predictions, demonstrating the efficacy of linear regression in forecasting, and proposes avenues for future research in the field.

B. AUTOREGRESSIVE INTEGRATED MOVING AVERAGE (ARIMA)

Adem Üntez et al. [2]focused on the analysis and prediction of exchange rates using algorithms such as ARIMA, Neural Network, and Fuzzy Neuron. The ARIMA model was employed to forecast exchange rates and compared with two other algorithms. The results revealed that while ARIMA provided good outcomes, it did not outperform the neural network in exchange rate prediction. According to the evaluations, Fuzzy Neuron demonstrated the best forecasting performance, achieving an accuracy rate of 95.39% using the RMSE metric and 7.89% using the MAPE metric.The author introduced these findings and applied algorithms in the context of exchange rate prediction. The study highlights the superiority of Fuzzy Neuron in terms of forecasting accuracy.

C. SUPPORT VECTOR REGRESSION (SVR)

Guo et al. [3] presents a hybrid model for enhancing energy consumption prediction in the context of energy management systems. The research integrates feature selection (FS) algorithms, including stepwise, Lars, and Boruta, with machine learning methods, specifically Random Forest (RF), Gradient Boosting Regressor (GBR), and Support Vector Regression (SVR). The paper focuses on parameter optimization for each model and employs four evaluation indicators (MAE, MSE, RMSE, and R2) to assess their performance.

The key findings reveal that the SVR-Boruta hybrid model outperforms others, achieving an accuracy of 90.585%. SVR consistently demonstrates superior performance compared to GBR and RF, and Boruta is identified as the most effective FS algorithm.

D. LONG SHORT-TERM MEMORY (LSTM)

Rahmi Yunida and et al. [4] from Lambung Mangkurat University explores the effectiveness of Long Short-Term Memory (LSTM) and Bidirectional LSTM (Bi-LSTM) in identifying natural disaster reports from social media. Using word2vec for text-to-vector transformation, the combination of word2vec and Bi-LSTM achieved an improved accuracy of 72.17% compared to LSTM's 70.67%. The study suggests the potential for further enhancement by adjusting parameters, offering avenues for future research to explore improved model performance.

E. GRAPH NEURAL NETWORKS (GNN)

Zexi Huang and et al [5] study about link prediction, a crucial task in various graph applications, focusing on the limitations of Graph Neural Networks (GNNs) in handling class imbalance. The authors introduce Gelato, a novel framework that combines topological and attribute information for link prediction without relying on GNNs. Gelato applies topology-centric graph learning and Autocovariance, a topological heuristic, achieving superior accuracy, faster training, and fewer parameters compared to state-of-the-art GNNs. The paper emphasizes the importance of unbiased testing and proposes the use of the N-pair loss for link prediction training. The contributions include scrutinizing link prediction evaluation and training, proposing an effective alternative to GNNs, and introducing unbiased training with the N-pair loss.

F. EXTREME GRADIENT BOOSTING (XGBOOST)

Qingwen Jin et al. [6] established predictive models using the Best Track TC dataset to anticipate Tropical Cyclone (TC) intensity in the Western North Pacific (WNP). Employing the XGBOOST model, we conducted predictions for 6, 12, 18, and 24-hour intensities across six scenarios. The feature set was meticulously designed through brainstorming and CLIPER methods. Testing the model on recent TCs (Hato, Rammasum, Mujiage, and Hagupit) produced compelling outcomes:

The XGBOOST model's accuracy significantly improved by integrating climatology and persistence factors, environmental factors, brainstorm features, intensity category, and TC month. Across all six scenarios, the model achieved a mean absolute error (MAE) < 4.50 m/s, correlation coefficient (CC) > 0.89, and normalized root mean square error (NRMSE) < 10.00%. Model C2 exhibited the highest accuracy among scenarios A (A1 and A2), B (B1 and B2), and C (C1 and C2).

Evaluation in the Western North Pacific (WNP) using NRMSE, MAE, and CC parameters revealed MAEs of 1.61, 2.44, 3.10, and 3.70 m/s for 6, 12, 18, and 24-hour lead times, respectively. Corresponding CCs were 0.99, 0.97, 0.95, and 0.93, while NRMSEs were 3.09%, 4.72%, 6.00%, and 7.18%. MAE and NRMSE increased with lead time, accompanied by a gradual decrease in CC. Noteworthy is the superior performance of the XGBOOST model compared to traditional Back-Propagation Neural Network (BPNN) models for the same predictors and independent prediction samples.

G. SEASONAL AUTOREGRESSIVE INTEGRATED MOVING AVERAGE WITH EXOGENOUS VARIABLES(SARIMAX)

Singh et al. (2020) developed a new hybrid model of discrete wavelet decomposition and autoregressive integrated moving average (ARIMA) models to forecast the casualties cases of COVID-19. The study focused on predicting death cases in five countries majorly afflicted by COVID-19, namely France, Italy, Spain, the United Kingdom, and the United States. The Wavelet-ARIMA model was used to divide the input dataset into component series, which were separately subjected to an appropriate econometric model. The model produced significantly better outcomes for Italy, Spain, and the United Kingdom, and a 50% better outcome for France and the United States. The hybrid ARIMA model reduced errors by nearly 50% compared to the ARIMA model. The authors proposed the model as a better prediction model for everyday recovered cases, confirmed cases, and deceased cases in India. They used an optimized SARIMAX model and tuned the hyperparameters using grid search cross-validation to obtain better and optimal results. The study aimed to inspire and assist policymakers in making decisions based on expected outcomes and to track the spread of COVID-19. The proposed approach was designed to be useful in disease control methods and in tracking the spread of COVID-19. The study also discussed the motivation behind predicting the COVID-19 cases and the potential impact of the research on public healthcare and government decision-making.

H. FULLY CONVOLUTIONAL NETWORK (FCN)

Liu et al. [7] present a study on fault detection using Fully Convolutional Networks (FCN) in 3D seismic images. The FCN model is applied to automatically interpret faults in an oil field in eastern China. The study area contains complicated faults, and the FCN model is shown to outperform automatic and common fault detection methods. The FCN model provides higher sensitivity and continuity with less noise, making it highly efficient for fault prediction compared to seismic attributes. The paper provides a detailed description of the FCN architecture, training process, and practical application results.

III. MATERIALS

A. DATASETS

This study revolves around stock price predictions within the banking sector, specifically focusing on datasets from key financial institutions in Vietnam. The datasets are sourced from Vietnam Joint Stock Commercial Bank for Industry and Trade (CTG), Sai Gon Thuong Tin Commercial Joint Stock Bank (STB), and Joint Stock Commercial Bank for Investment and Development of Vietnam (BID). The data covers the period from January 27, 2014, to December 21, 2023, and exhibits consistent characteristics across the selected banks.

Attribute	Describe
Date	Stock trading day
Price	The closing/final price of the stock at a certain time
Open	The initial opening/price of the stock at a certain time
High	Highest opening price
Low	Lowest price of opening price
Vol.	Number of transactions during the day
Change %	Percentage change between the current day's closing price compared to the previous day.

TABLE 1: *Describe attribute of datasets.*

The objective of this article is to predict closing prices; therefore, only descriptive statistical information pertaining to the "Price" column will be provided.

1) Detail statistical

	BID	CTG	STB
Count	2471	2471	2471
Mean	25249.1279	19293.5084	16500.7811
Standard deviation	11801.6556	7937.9193	7009.3544
Max	49100	41141.3	49100
Min	8006.4	9002.9	8006.4
25%	12786.55	12758	11400
50%	26021.7	16291.2	13769
75%	34268.9	27000	21125
Mode	8952	9594.3	11200
Median	1234	16291.2	13769
Variance	693745.4286	63010563.37	49131049.5
Covariance	13927906.05	63010563.37	49131049.5
Kurtosis	-1.2561	-0.8716	-0.8716
Skewness	0.14488	0.6265	-0.2501

TABLE 2: *Detail statistical of BID, CTG, STB datasets.*

2) Visualization

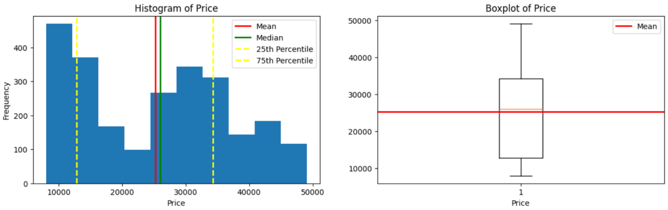

FIGURE 1: *Visualization price attribute of BID.*

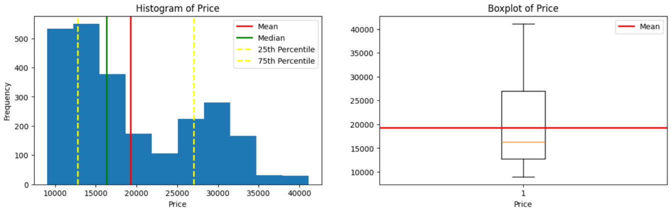

FIGURE 2: *Visualization price attribute of CTG.*

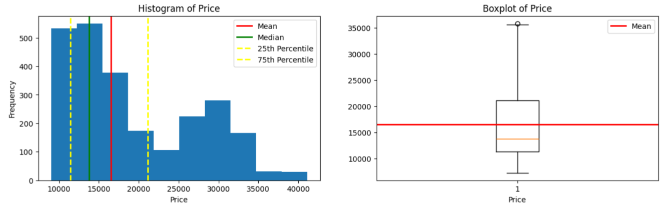

FIGURE 3: *Visualization price attribute of STB.*

B. TOOLS

1) Python

In this reasearch, Python is employed as the programming language, and the accompanying platform is the Jupyter notebook. Additionally, we make use of Python's inherent libraries such as Pandas for manipulating data in the structure of data frames. Matplotlib is utilized for creating visual representations of data through plots. Numpy is applied to facilitate mathematical and matrix operations throughout the experiment. Lastly, the Scikit-learn library provides support for machine learning and regression models.

C. DATASET SPLIT RATIO

In our research on time series data, we decided split the data into training and testing sets using the 9:1, 8:2 and 7:3 ratio. The most common ratio used for splitting time series data is the 7:3 ratio, where 70% of the data is used for training, 20% for testing. This ratio is commonly used because it provides enough data for the model to learn from, while also allowing for a separate testing set to validate its performance.

In certain instances, a 7:3 train-test split ratio is a frequent choice in machine learning for various reasons. It offers a significant volume of data for training intricate models while maintaining a sufficient dataset for assessing generalization performance. This equilibrium proves beneficial in scenarios where computational resources are constrained, aligning with established practices in the machine learning community. The ratio plays a crucial role in balancing bias and variance, facilitating effective model training and evaluation. Ultimately, the selection of the ratio hinges on the specific attributes of the dataset and the objectives of the analysis.

IV. METHODS

A. LINEAR REGRESSION

Linear regression analysis is used to predict a variable's value based on another variable's value. The variable that needs to be predicted is called the dependent variable. The variable that is used to predict the other variable's value is called the independent variable. This statistical method finds an equation that best predicts the y variables as a linear function of the x variables [8].

The formula for a univariate linear regression [9]:

$$Y = \beta_0 + \beta_1 X_1 + \varepsilon$$

The formula for a multiple linear regression:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_k X_k + \varepsilon$$

Where:

- Y is the dependent variable.
- $X_1, \ldots X_k$ are the independent (explanatory) variables.
- $\boldsymbol{\beta_0}$ is the intercept term.
- $\boldsymbol{\beta_1}, \ldots \boldsymbol{\beta_k}$ are the regression coefficients for the independent variables.
- ε is the error term.

B. ARIMA

ARIMA is a popular time series analysis and forecasting method. It combines three components: Autoregressive (AR) Component (p), Moving Average (MA) Component (q), Integrated (I) Component (d). The AR component captures the relationship between the current observation and its previous observations. It involves regressing the current value on its own past values. The MA involves modeling the relationship between the current observation and a residual error from a moving average model applied to lagged observations. The I represents the difference of the time series to make it stationary. The ARIMA model can be simply written as:

$$\begin{gathered} y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + \theta_1 \epsilon_{t-1} \\ + \theta_2 \epsilon_{t-2} + \cdots + \theta_q \epsilon_{t-q} \end{gathered}$$

Where:

- y_t is the observed data at time t
- c is a constant
- ϕ_1, ϕ_p are the Auto Regressive (AR) coefficients corresponding to order p
- θ_1, θ_q are the Moving Average (MA) coefficients corresponding to order q
- $\epsilon_{t-1}, \epsilon_{t-q}$ are the previous errors used to calculate the current value

C. SVR

Support Vector Regression, as its name implies, is a regression algorithm capable of handling both linear and non-linear regression tasks. This approach is rooted in the principles of the Support Vector Machine (SVM). Unlike SVM, which functions as a classifier for predicting discrete categorical labels, SVR serves as a regressor designed specifically for predicting continuous ordered variables.

SVR kernel function:

Kernel	Function
Polynomial	$(x_i * x_j + 1)^d$
RBF	Exp $(-\text{y}\,\lvert x_i - x_j \rvert)$
Sigmoid	tanh $(yx^T z + r)$

TABLE 3: *SVM Kernel.*

D. LSTM

The LSTM (Long Short-Term Memory) model is a specialized neural network architecture extensively employed in the realm of time series processing. First introduced by Hochreiter and Schmidhuber in 1997, it has since evolved into one of the pivotal models within the domain of deep learning for time series data.

This model addresses a significant challenge encountered in traditional Recurrent Neural Networks (RNNs), namely the problem of long-term information loss. In conventional RNNs, information propagation is constrained by a limited number of neurons, and it tends to diminish as the sequence length increases. LSTM mitigates this issue by incorporating a memory cell and gates, which regulate the flow of information during the processing of time series data.

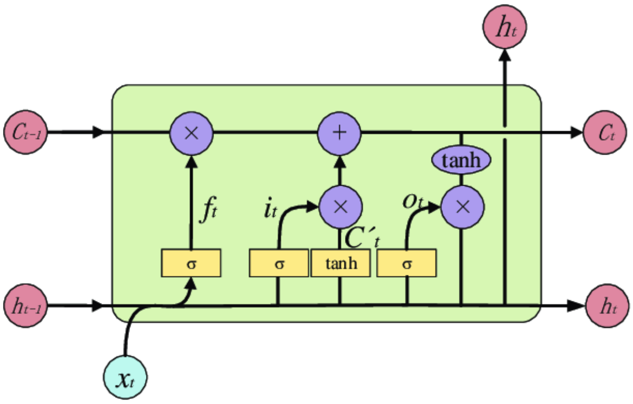

FIGURE 4: *Architectural of the LSTM model.*

The rationale behind employing the LSTM model in time series processing lies in its proficiency in managing long-term dependencies. Through the incorporation of a memory cell and gates, LSTM exhibits the capability to preserve crucial information from previous time steps during the training phase. This functionality empowers the model to generate more accurate predictions, particularly for extended and intricate sequences. The specific formula governing the LSTM's operations is as follows [10]:

- **Input gate (i):** $i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] + b_i\right)$
- **Forget gate (f):** $f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right)$
- **Output gate (o):** $o_t = \sigma\left(W_o \cdot [h_{t-1}, x_t] + b_o\right)$
- **Memory cell (C):** $\tilde{C_t} = \tanh\left(W_C \cdot [h_{t-1}, x_t] + b_C\right)$ $C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C_t}$
- **Hidden state (h):** $h_t = o_t \cdot \tanh\left(C_t\right)$

Where:

- x_t is the input at time t.
- h_{t-1} is the hidden state from the previous layer.
- i_t, f_t, o_t are the values of the gates at time t.
- C_t is the memory cell state at time t.
- h_t is the hidden state at time t.
- W_i, W_f, W_o, W_C are weight matrices.
- b_i, b_f, b_o, b_c are bias vectors.

The formulas describe how the gates and states of an LSTM are computed based on the current input and the previous state.

This process allows the LSTM to process and store crucial information from the past and influence the prediction outcomes.

E. GNN

Graph Neural Networks (GNNs) represent a specialized category of neural networks designed to operate on data organized in a graph structure. They draw inspiration from Convolutional Neural Networks (CNNs) and graph embedding techniques. GNNs excel in tasks related to predicting nodes, edges, and other graph-centric objectives.

Just as CNNs find application in image classification by processing the grid of pixels, GNNs are employed for analyzing graph structures, where each node corresponds to a specific entity. In the context of text classification, Recurrent Neural Networks (RNNs) are commonly used. Similarly, GNNs can be applied to graph structures representing sentences, where each word serves as a node.

The introduction of GNNs became necessary when Convolutional Neural Networks faced challenges in achieving optimal results, particularly when dealing with graphs of arbitrary size and intricate structures. GNNs emerged as a powerful solution to address these complexities and enhance the performance of neural networks in graph-based tasks [11]. The propagation rule for GNN can be generalized as:

$$\mathbf{h}_v^{(t)} = \sum_{u \in N(v)} f\left(\mathbf{x}_v, \mathbf{x^e}_{(v,u)}, \mathbf{x}_u, \mathbf{h}_u^{(t-1)}\right) \qquad [12]$$

Where:

- $\mathbf{h}_v^{(t)}$: The hidden feature of node v at time t
- u $\in N(v)$: Neighbor of node v
- $\mathbf{x}_v$: The feature vector of the node v
- $\mathbf{x^e}_{(v,u)}$: The edge feature of vector of the edge(v,u)
- $\mathbf{x}_u$: The feature vector for the neighboring node u
- $\mathbf{h}_u^{(t-1)}$: The hiden feature vector of node u in last time step

F. XGBOOST

XGBoost, which stands for Extreme Gradient Boosting, is a scalable, distributed gradient-boosted decision tree (GBDT) machine learning library. It provides parallel tree boosting and is the leading machine-learning library for regression, classification, and ranking problems [13].

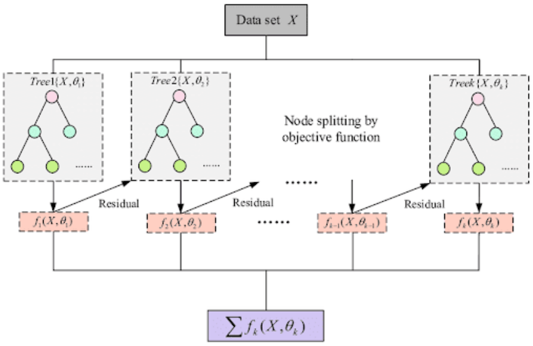

FIGURE 5: *Architectural of the XGBoost model.*

Regularized Learning Objective [14]:

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda\|w\|^2$$

With:

- γ and λ is the hyperparameters
- T is the number of tree node
- w is the vector of node

Objective Function: Training Loss + Regularization

$$L(\phi) = \sum_{i=n}^{n} l(yi - \hat{y}i) + \sum_{k=1}^{k} \Omega\left(f_k\right)$$

Taylor expansion:

$$L^{(t)} = \sum_{i=1}^{l}\left[l\left(y_i, \hat{y}_i^{(t-1)}\right) + g_i f_t\left(x_i\right) + \frac{1}{2} h_i f_t^2\left(x_i\right)\right] + \Omega\left(f_t\right)$$

G. SARIMAX

SARIMAX(Seasonal Auto-Regressive Integrated Moving Average with eXogenous factors) is an updated version of the ARIMA model [15]. Two different types of orders must be provided in the SARIMAX models parameter. We refer to this order as a seasonal order in which we are required to submit four integers. The first one is comparable to the ARIMAX model (p, d, and q), and the other is to indicate the influence of seasonality. The model can be described in these components: SARIMAX(p,d,q)(P,D,Q,s). Where

- (p, d, q) is the same with variable of ARIMA
- P: Seasonal AutoRegressive (SAR). The parameter P represents the seasonal autoregressive component (the number of lags in previous season)
- D: Seasonal Differencing. parameter D represents the seasonal differencing.
- Q: Seasonal Moving Average (SMA). The parameter Q represents the seasonal moving average component. s : The parameter s denotes the length of the seasonal period in the time series

H. FCN

The Fully Convolutional Network (FCN) is a deep learning architecture designed for end-to-end classification tasks on univariate time series data. It leverages the power of convolutional neural networks (CNN) to capture hierarchical features directly from the input time series.

The fundamental building block of FCN is the convolutional layer. The formula for calculating the output of a convolutional layer is as follows:

$$\operatorname{conv}\left(i, \quad \left(\sum_{u=0}^{M-1} \sum_{v=0}^{M-1} w_{u,v} x_{i+u,j+v} + b\right)\right. \qquad [7]$$

Where:

- $\operatorname{conv}(i, j)$ is the convolution result, also known as the feature map;
- M indicates the size of the convolution kernel $(\mathrm{M}\times$ M);
- $w_{u,v}$ is the weight of the convolution kernel in line u and column v;
- $x_{i+u,j+v}$ is the input;
- b is the bias;

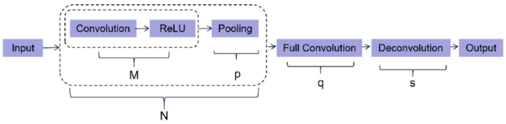

FIGURE 6: *Architectural of the FCN model.*

A convolutional block is comprised of a sequence of M consecutive convolutional layers, followed by p pooling layers. In the architecture of a convolutional network, N such convolutional blocks can be sequentially stacked. This stacking is then succeeded by q fully convolutional layers and s deconvolutional layers.

V. RESULT

A. DISCUSSION

During the training of the prediction models, we decided to split the data into training and test sets using a ratio of 9:1, 8:2 and 7:3 and employ four primary metrics after prediction: Root Mean Squared Error (RMSE), Mean Absolute Percentage Error (MAPE), and Mean Absolute Error (MAE), Mean Squared Logarithmic Error(MSLE) and we decided split the data into training and testing sets using the 9:1, 8:2 and 7:3 ratio to assess their accuracy and performance on the test dataset.

MSLE, or Mean Squared Logarithmic Error, measures the average logarithmic difference between the actual and predicted values. It calculates the mean of the squared logarithmic differences between the natural logarithm of the actual and predicted values. This metric is particularly useful when the predictions span a wide range of magnitudes.

$$L(y, \hat{y}) = \frac{1}{N} \sum_{i=0}^{N} \left(\log\left(y_i + 1\right) - \log\left(\hat{y}_i + 1\right)\right)^2 \qquad [16]$$

where:

- $\{y_i\}$ is the actual observations time series.

- $\{\hat{y}_i\}$ is the estimated or forecasted time series.
- N is the number of data points.

The purpose of the 1 in $\log_e(1+y)$ is to avoid issues when y is equal to 0 . Using $\log_e(1+y)$ helps prevent problems associated with taking the logarithm of 0 . When y is $0, \log_e(1+y)$ becomes $\log_e(1)$, which is 0 . This avoids undefined results when calculating the logarithm of 0 . Considering the outcomes of the assessments provided below, it is evident that each dataset requires its own model along with an optimal division of data. In order to achieve the highest predictive performance, we will choose the top three models, each with a distinct scaling ratio tailored to the specific characteristics of the three datasets.

1) RMSE, MAPE, MAE, MSLE based models comparisons BID dataset

Model	Ratio	RMSE	MAPE	MAE	MLSE
Linear Regression	9:1	3575.566	7.104	3188.989	0.007
	8:2	4381.346	9.037	3639.019	0.012
	7:3	8930.071	25.2312	7894.75	0.1123
ARIMA	9:1	5675.077	11.584	5223.077	0.018
	8:2	7542.693	15.081	6516.397	0.036
	7:3	5219.372	13.4006	4237.515	0.0307
SVR	9:1	1328.794	2.142	977.36	0.001
	8:2	5711.188	8.967	4025.042	0.019
	7:3	5131.343	6.859	3048.904	0.016
LSTM	9:1	996.081	1.809	796.857	0.022
	8:2	922.978	1.755	691.624	0.024
	7:3	770.444	1.936	578.555	0.026
GNN	9:1	820.214	1.321	579.959	0.019
	8:2	937.627	1.71	682.262	0.024
	7:3	863.872	1.637	623.492	0.023
XGBoost	9:1	3150.056	5.716	2533.101	0.005
	8:2	6222.381	12.569	5226.414	0.025
	7:3	5586.812	14.5	4577.745	0.036
SARIMAX	9:1	700.007	1.0775	476.721	0.0003
	8:2	861.796	1.408	566.194	0.0005
	7:3	794.685	1.3797	529.181	0.00045
FCN	9:1	3901.956	8.172	3631.392	0.093
	8:2	3515.799	6.948	2957.433	0.085
	7:3	2303.135	5.885	1816.375	0.075

TABLE 4: *Metric score of BID data.*

With the BID dataset, we have the best predictive models based on each ratio respectively: SARIMAX (9 : 1), SARIMAX (8 : 2), LSTM (7 : 3)

2) RMSE, MAPE, MAE, MSLE based models comparisons CTG dataset

Model	Ratio	RMSE	MAPE	MAE	MLSE
Linear Regression	9:1	1853.464	5.117	1464.145	0.003982
	8:2	3501.878	8.7571	2503.493	0.0149
	7:3	8930.071	25.2312	7894.75	0.1123
ARIMA	9:1	2208.096	5.9831	1805.127	0.0057
	8:2	5987.446	19.7287	5310.095	0.0407
	7:3	5219.372	13.4006	4237.515	0.0307
SVR	9:1	300.253	0.7812	228.209	0.0001
	8:2	387.701	0.9903	276.6902	0.0002
	7:3	9110.259	20.4597	6688.987	0.1212
LSTM	9:1	493.787	1.174	345.004	0.017
	8:2	967.919	1.851	725.68	0.025
	7:3	770.444	1.936	578.555	0.026
GNN	9:1	504.488	1.232	361.82	0.017
	8:2	653.52	1.723	478.435	0.024
	7:3	680.315	1.685	493.915	0.024
XGBoost	9:1	2237.022	6.046	1767.626	0.006
	8:2	3807.334	10.252	2917.726	0.019
	7:3	5586.812	14.5	4577.745	0.036
SARIMAX	9:1	465.467	1.128	333.603	0.00025
	8:2	559.668	1.323	373.279	0.000431
	7:3	581.798	1.317	389.267	0.000416
FCN	9:1	1062.219	2.933	843.677	0.038
	8:2	1249.009	3.631	993.409	0.047
	7:3	2303.135	5.885	1816.375	0.075

TABLE 5: *Metric score of CTG data.*

With the CTG dataset, we have the best predictive models based on each ratio respectively: SVR (9 : 1), SVR (8 : 2), SARIMAX (7 : 3)

3) RMSE, MAPE, MAE, MSLE based models comparisons STB dataset

Model	Ratio	RMSE	MAPE	MAE	MLSE
Linear Regression	9:1	6725.425	21.994	6304.856	0.06998
	8:2	9468.872	29.869	8416.653	0.162
	7:3	16467.08	58.825	15769.97	0.865
ARIMA	9:1	5094.63	15.184	4433.826	0.0364
	8:2	6007.599	21.576	4784.109	0.059
	7:3	14478.12	48.944	13430.23	0.562
SVR	9:1	377.578	1.10002	304.433	0.000189
	8:2	1782.579	2.8871	857.119	0.0035
	7:3	850.426	2.695	692.848	0.034
LSTM	9:1	1069.909	2.99	803.518	0.038
	8:2	595.21	1.614	456.363	0.021
	7:3	850.426	2.695	692.848	0.034
GNN	9:1	615.577	1.753	488.211	0.022
	8:2	685.262	2.079	520.726	0.028
	7:3	683.155	2.077	521.56	0.028
XGBoost	9:1	3479.951	10.592	2825.942	0.015
	8:2	5866.458	19.021	4678.072	0.058
	7:3	11284.01	37.169	10289.47	0.269
SARIMAX	9:1	538.432	1.322	372.267	0.000375
	8:2	596.0098	1.6055	410.628	0.00059
	7:3	607.485	1.657	423.144	0.0006
FCN	9:1	1643.017	4.746	1387.132	0.057
	8:2	2375.49	6.943	1558.836	0.103
	7:3	4153.258	14.068	3832.248	0.161

TABLE 6: *Metric score of STB data.*

With the STB dataset, we have the best predictive models based on each ratio respectively: SVR (9 : 1), LSTM (8 : 2), SARIMAX (7 : 3)

B. VISUALIZATION

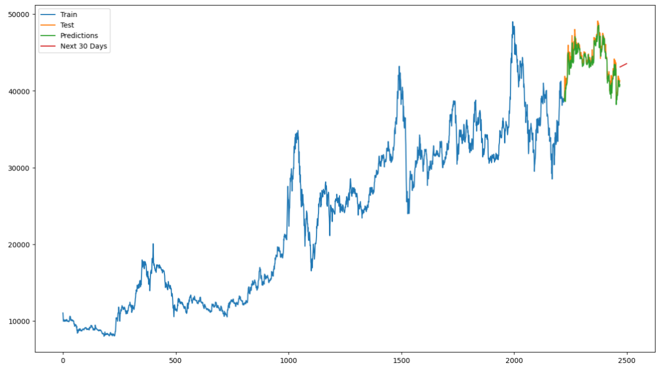

FIGURE 7: *Result of SARIMAX (9 : 1) on BID data.*

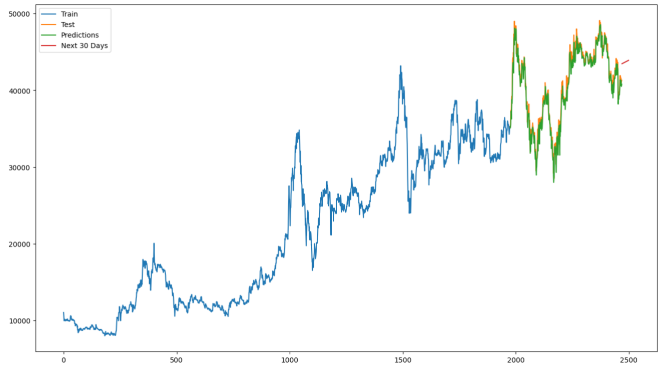

FIGURE 8: *Result of SARIMAX (8 : 2) on BID data.*

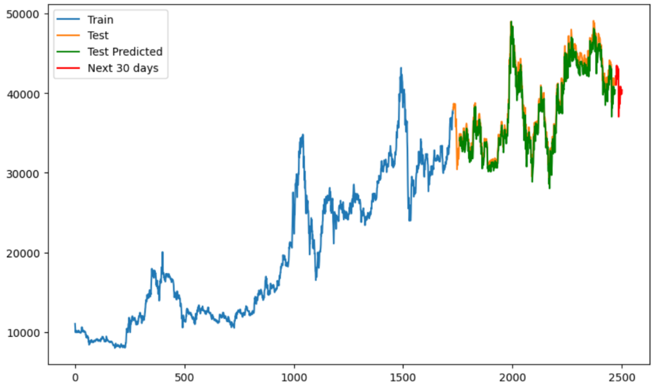

FIGURE 9: *Result of LSTM (7 : 3) on BID data.*

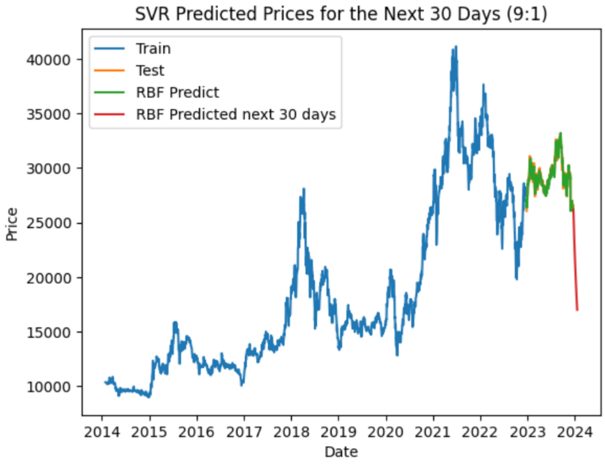
SVR Predicted Prices for the Next 30 Days (9:1)

FIGURE 10: *Result of SVR (9 : 1) on CTG data.*

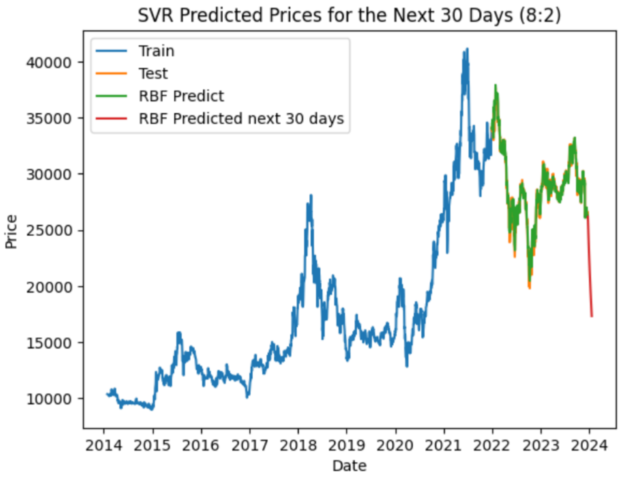
SVR Predicted Prices for the Next 30 Days (8:2)

FIGURE 11: *Result of SVR (8 : 2) on CTG data .*

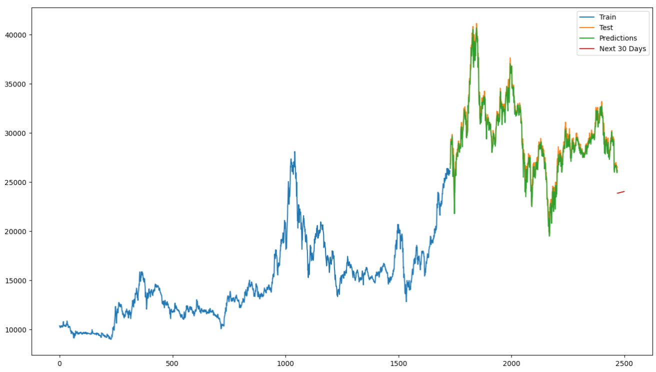

FIGURE 12: *Result of SARIMAX (7 : 3) on CTG data.*

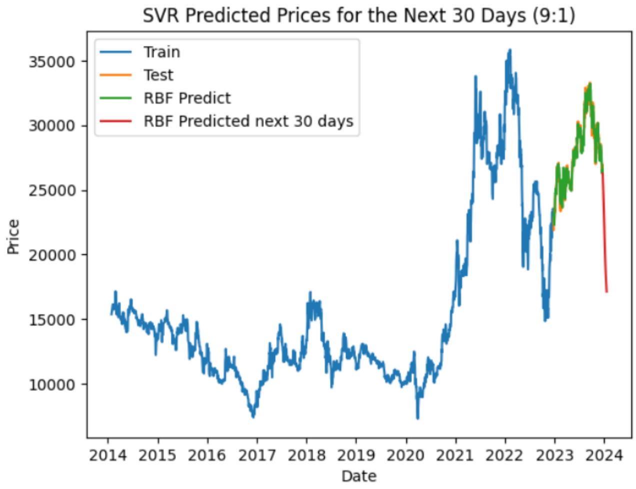

FIGURE 13: *Result of SVR (9 : 1) on STB data.*

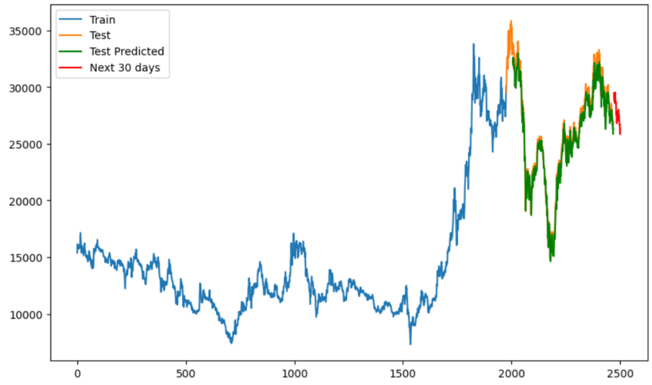

FIGURE 14: *Result of LSTM (8 : 2) on STB data.*

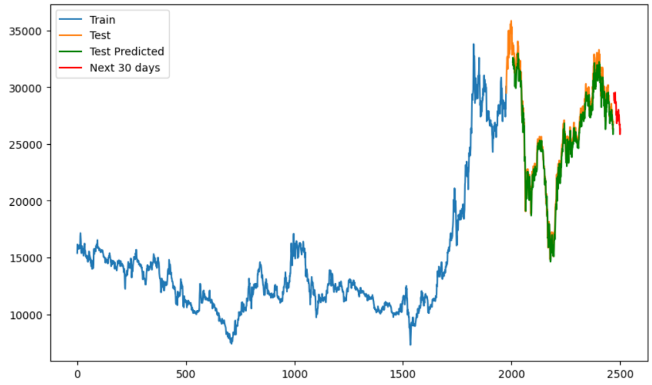

FIGURE 15: *Result of SARIMAX (7 : 3) on STB data.*

VI. CONCLUSION

A. OVERALL CONCLUSION

The provided table presents accuracy scores for eight models across three datasets (BID stock price, CTG stock price, STB stock price) under three different train-test ratios (7:3, 8:2, and 9:1). Four metrics, namely RMSE, MAPE, MAE, and MSLE, are utilized to assess the model accuracy.

The study's findings indicate that among the eight models evaluated (Linear Regression, ARIMA, SARIMAX, GNN, LSTM, SVR, XGBoost, and FCN), SVR, LSTM, and SARIMAX were the most effective in predicting the future prices of BID, CTG, and STB stocks within the resulting time series. This underscores the importance of exploring diverse modeling approaches in financial analysis. Furthermore, the study suggests the potential usefulness of LSTM, SVR, and SARIMAX models for forecasting future stock prices.

The research underscores the significance of considering a range of modeling techniques in financial analysis. It also highlights the potential efficacy of employing LSTM, SVR, and SARIMAX models for forecasting future stock prices. To validate these findings and assess the performance of other models across various stock price prediction tasks, further research could be conducted.

B. CHALLENGES ENCOUNTERED

In the pursuit of our research project titled "Predictive Modeling of Vietnamese Real Estate Trends: A Fusion of Data Analytics and Econometric Approaches for Enhanced Forecasting" we grappled with distinctive challenges that necessitated thorough consideration:

- **Complexity in data processing:** Real estate data is intricate and diverse, necessitating the use of precise data processing techniques to ensure the feasibility and accuracy of prediction models.
- **Building robust prediction models:** Constructing real estate prediction models is a complex task that demands in-depth knowledge of the field. Critical decisions, such as algorithm selection, data preprocessing methods, and determining essential variables for the models, were encountered.
- **Evaluating model effectiveness:** Various algorithmic and statistical indicators were utilized to assess the performance of the prediction models. However, the results indicated that the accuracy of the models was still unsatisfactory.

C. FUTURE INTENTION

- **Enhancing data selection and processing skills:** We will continue researching and implementing state-of-the-art methods for data selection and processing to ensure that our prediction models operate with feasible and accurate input data.
- **Employing advanced prediction models:** Exploration of advanced techniques such as Deep Learning and Reinforcement Learning to develop more sophisticated and accurate prediction models, with the potential to enhance overall effectiveness in predicting real estate trends.
- **Strengthening model evaluation methods:** Investigation and adoption of the latest and widely accepted indicators within the field of real estate trend prediction, such as Mean Absolute Scaled Error (MASE), Mean

Absolute Error Percentage (MAPE), and Symmetric Mean Absolute Percentage Error (SMAPE).

By implementing these solutions, we are confident that we can significantly improve the accuracy and effectiveness of our real trend prediction models in the future.

.

ACKNOWLEDGMENT

I'm truly grateful for the unwavering support of **Assoc. Prof. Dr. Nguyen Dinh Thuan, Mr. Nguyen Minh Nhut, and Ms. Nguyen Thi Viet Huong. Assoc. Prof. Dr. Nguyen Dinh Thuan**'s guidance has been pivotal, deepening our understanding of the subject. Mr. Nguyen Minh Nhut's innovative teaching and encouragement made learning engaging and rewarding, while Ms. Nguyen Thi Viet Huong's commitment to a positive educational environment enhanced our experience.

Special thanks to **Mr. Nhut** and **Ms. Nguyen Thi Viet Huong** for their outstanding contributions to our project, showcasing expertise and collaborative spirit. Reflecting on this journey, I'm grateful for the transformative role played by these esteemed educators and mentors in my academic and personal development. Thank you for the privilege of learning under your guidance.

REFERENCES

[1] Sonali Antad, Saloni Khandelwal, Anushka Khandelwal, Rohan Khandare, Prathamesh Khandave, Dhawal Khangar, and Raj Khanke. Stock price prediction website using linear regression-a machine learning algorithm. In ITM Web of Conferences, volume 56, page 05016. EDP Sciences, 2023.

[2] AS Babu and SK Reddy. Exchange rate forecasting using arima. Neural Network and Fuzzy Neuron, Journal of Stock & Forex Trading, 4(3):01–05, 2015.

[3] Ni Guo, Weifeng Gui, Wei Chen, Xin Tian, Weiguo Qiu, Zijian Tian, and Xiangyang Zhang. Using improved support vector regression to predict the transmitted energy consumption data by distributed wireless sensor network. EURASIP Journal on Wireless Communications and Networking, 2020, 06 2020.

[4] Rahmi Yunida, Mohammad Reza Faisal, Muliadi Muliadi, Fatma Indriani, Friska Abadi, Irwan Budiman, and Septyan Prastya. Lstm and bi-lstm models for identifying natural disasters reports from social media. Journal of Electronics Electromedical Engineering and Medical Informatics, 5:241–248, 10 2023.

[5] Zexi Huang, Mert Kosan, Arlei Silva, and Ambuj Singh. Link prediction without graph neural networks. 05 2023.

[6] Qingwen Jin, Xiangtao Fan, Jian Liu, Zhuxin Xue, and Hongdeng Jian. Using extreme gradient boosting to predict changes in tropical cyclone intensity over the western north pacific. Atmosphere, 10:341, 06 2019.

[7] Jizhong Wu, Bo Liu, Hao Zhang, Shumei He, and Qianqian Yang. Fault detection based on fully convolutional networks (fcn). Journal of Marine Science and Engineering, 9, 03 2021.

[8] About Linear Regression | IBM — ibm.com. [Accessed 27-12-2023].

[9] Multiple Linear Regression — corporatefinanceinstitute.com. [Accessed 27-12-2023].

[10] Pyae Phyo and Yungcheol Byun. Hybrid ensemble deep learning-based approach for time series energy prediction. Symmetry, 13:1942, 10 2021.

[11] A Comprehensive Introduction to Graph Neural Networks (GNNs) — datacamp.com. [Accessed 28-12-2023].

[12] Jonathan Hui. Graph Neural Networks (GNN, GAE, STGNN) — jonathan-hui.medium.com. [Accessed 28-12-2023].

[13] What is XGBoost? — nvidia.com. [Accessed 27-12-2023].

[14] XGBoost - GeeksforGeeks — geeksforgeeks.org. [Accessed 27-12-2023].

[15] Complete Guide To SARIMAX in Python for Time Series Modeling. [Accessed 28-12-2023].

[16] Md Ashraful Haque, MA Zakariya, Samir Salem Al-Bawri, Zubaida Yusoff, Mirajul Islam, Dipon Saha, Wazie M Abdulkawi, Md Afzalur Rahman, and Liton Chandra Paul. Quasi-yagi antenna design for lte applications and prediction of gain and directivity using machine learning approaches. Alexandria Engineering Journal, 80:383–396, 2023.