Grenze International Journal of Engineering and Technology, January Issue

TimeHorizon Stock Market Price Prediction

Nikhil Raj[1], Parth Krishnan B Menon[2], Sidharth P[3], Anand Venugopalan[4], Sreeshma K[5] and V K Vadkeppattu[6]
[1-5]Department of Computer Science Engineering, Sreepathy Institute of Management and Technology, Vavanoor, India
Email: {nikhil.raj, parthkrishnanb.menon, sidharth.p, anand.venugopalan }@simat.ac.in, sreeshma.k@simat.ac.in
[6]AISECT Limited, Novosibirsk State University
Email: vijayananandkv.anand@gmail.com

***Abstract*— Forecasting of stock market is a way to predict future prices of stocks. It is a long-time attractive topic for researcher and investors from its existence. The Stock prices are dynamic day by day, so it is hard to decide what is the best time to buy and sell stocks. This is a work for prediction of stock market prices for both short-term and long-term investors seeking early recommendations and used in need of financial distress warnings. The work explores the application of various machine learning algorithms including Linear regression, Random Forest, and Support Vector Regression, to forecast stock market prices in the stock market. From these, machine learning algorithms are trained and evaluated for each model's performance using metrics such as accuracy, precision score etc. The finding showcased that efficiency of Support Vector Machines (SVM) and Random Forest has accurately predicted stock market values. The proposed model is SVM-ICA fusion model combining SVM and independent component analysis (ICA). The system was found to achieve an accuracy of 60% using random forest, trained on an average of 1000 data values The proposed model aims to provide forecasting on future stock prices, providing investors with insights on informed decision-making. Additionally, the practical implications of our work also depict the applications in algorithmic trading, risk management and optimization.**

***Index Terms*— National Stock Exchange (NSE), Support Vector Regression (SVM), Random Forest, Stock Market, Root Mean Square Error (RMSE), Feature Engineering, Support Vector Machine and Independent component analysis (SVM-ICA).**

I. INTRODUCTION

Forecasting stock market prices has been a key focus for both researchers and investors, driven by the challenge of predicting the movements amidst changing prices. With the creation of machine learning algorithms, new methods have emerged to overcome this issue, it offers diverse tools for forecasting future stock prices. This work mainly focuses on predicting stock market prices using data mining algorithms such as Random Forest and Support Vector Regression, aiming to compare their performance and evaluate their prediction accuracy through metrics like RMSE (Root Mean Square Error).
To improve prediction accuracy, the historical data from the NSE (National Stock Exchange) stock market is used and also has applied the pre-processing technique, including data cleaning and feature engineering so as to extract meaningful patterns from the historical data. These patterns are necessary for delivering reliable stock price forecasts. By leveraging this approach, the aim is to provide the investors with valuable insights to navigate the stock market.

Grenze ID: 01.GIJET.11.1.539

Through recognizing the uncertainty in stock price prediction, it emphasizes the need to display upper and lower price bounds to help investors understand potential trading range. Utilizing this approach, provides more comprehensive insights into the market dynamics, enabling investors to make better-informed decisions based on the potential risks and rewards.

Overall, this work highlights the importance of machine learning and in understanding stock market behavior. By making advanced forecasting models accessible, the aim is to light the investors and contribute to them by bringing more informed investment strategies, ultimately providing them with more effective wealth generation in the evolving financial landscape.

II. RELATED WORKS

The stock market prediction patterns are seen as an important activity and it is more effective. Hence, stock prices will lead to lucrative profits from sound taking decisions. Because of the stagnant and noisy data, stock market-related forecasts are a major challenge for investors. Therefore, forecasting the stock market is a major challenge for investors to use their money to make more profit. Deepak Kumar[1] reviewed 30 research papers focused on stock market prediction using machine learning (ML) techniques. They analyzed various strategies, including ANN and NN models, to assess their effectiveness in forecasting stock prices. The paper highlights limitations in current approaches and suggests that more parameters should be considered for accurate predictions. Alice Zheng[2] focuses on improving short-term stock price prediction using time series data, an area with historically low accuracy (50-60%). While previous research has achieved better results for long-term predictions, the authors introduce financial technical indicators combined with machine learning algorithms, resulting in improved prediction accuracy for short-term stock forecasting. M. Nabipour[3] focus on predicting future stock values from four groups in the Tehran Stock Exchange using 10 years of historical data. Machine learning algorithms such as LSTM, Adaboost, Gradient Boosting, and XGBoost were employed for predictions over 1 to 30 days. LSTM produced the most accurate results, with competitive performances from tree-based models like Adaboost and Gradient Boosting. The study by Abdullah Bin Omar [4] address the challenge of stock market forecasting, especially during the Covid-19 period, using machine learning models. They assess three models: autoregressive deep neural networks (AR-DNN(1, 3, 10) and AR-DNN(3, 3, 10)) and autoregressive random forest (AR-RF(1)) for forecasting in three time frames—pre-Covid-19, during Covid-19, and overall. Their results show that these models outperform traditional methods like ARIMA under high stock price fluctuations, offering practical use for investors and policymakers to make informed financial decisions. This paper by Zaharaddeen Karami Lawal [5] reviews various supervised machine learning models used for stock market prediction, focusing on their ability to enhance prediction accuracy. Techniques such as Support Vector Machine (SVM), which stands out for its performance and accuracy, as well as Artificial Neural Networks (ANN), K-Nearest Neighbor (KNN), Naïve Bayes, Random Forest, Linear Regression, and Support Vector Regression (SVR) are explored. These methods have shown promising results in predicting stock prices, aiding investors and decision-makers in making informed financial decisions. This study by Isaac Kofi Nti [6] introduces a novel ensemble classifier called GASVM, which combines Support Vector Machine (SVM) with Genetic Algorithm (GA) for stock market prediction. The GA optimizes feature selection and SVM kernel parameters to improve prediction accuracy, addressing issues of overfitting in high-noise, high-dimensional datasets. Jilin Zhang [7] introduces a CNN-BiLSTM-Attention-based model designed to enhance the accuracy of stock price prediction. Traditional forecasting methods such as Random Forest (RF), RNN, CNN, and LSTM are effective to some extent but have limitations. The proposed model leverages CNN and BiLSTM to extract temporal features from sequence data and uses an attention mechanism to assign weights to key features. Smith John [8] reviews various machine learning techniques employed in stock price prediction, focusing on their effectiveness in capturing market trends and anomalies. The paper highlights the comparative performance of traditional models versus advanced machine learning algorithms, such as neural networks and ensemble methods, in forecasting stock prices. Brown, Alice [9] reviews time series analysis techniques for financial forecasting, including autoregressive models and moving averages. The book also covers the use of AI and ML models like recurrent neural networks and support vector machines for improved predictions. Chang Sim Vui [10] did studies on applied Artificial Neural Networks (ANN) to predict stock market trends due to their ability to capture complex patterns in volatile market data. Researchers have explored various models, including LSTM and RNN, to enhance prediction accuracy by leveraging historical stock prices and technical indicators. Troy J. Strader [11] studies have explored the use of machine learning techniques for stock market prediction, comparing them to traditional methods. Artificial Neural Networks (ANN) have been a prominent approach, offering improved accuracy for complex stock market data. Support Vector Machines (SVM) have also been widely employed due to their efficiency in handling classification tasks. Hybrid approaches, combining techniques like Genetic Algorithm. Rosdyana Mangir Irawan Kusuma [12] did a research on Stock market prediction studies have leveraged Convolutional Neural Networks (CNN) to analyze candlestick charts and forecast future price movements. Combining CNN with models like Residual Networks has shown high prediction accuracy, exceeding 90% for markets such as Taiwan and Indonesia. These deep learning models offer promising results in improving stock market forecasting. Gourav Kumar[13] stated that stock market's role in a country's economic and social framework underscores the complexity and challenges of forecasting its movements. Due to the inherently noisy, volatile, and non-linear nature of stock price time series, accurate prediction is a demanding task for investors, analysts, and researchers.

III. MATERIALS AND METHODS

A. System Architecture

Investors face a lot of challenges in predicting stock market movements, this work aims to use machine learning algorithms for forecasting NSE stock market prices. By evaluating various algorithms and enhancing the prediction accuracy, so as to provide actionable insights to empower investors in navigating the complexities of the stock market.

The system architecture of this TimeHorizon Stock Market Price Prediction is depicted in the data flow diagram below. The architecture of the system follows a client-server model, where the server and the client are loosely coupled. Initially, the relevant stock data is retrieved from a third-party data provider via the cloud. The backend system preprocesses this data and constructs predictive models. Subsequently, predictions are generated, and the results are stored on a separate cloud infrastructure. This decentralized approach ensures efficient management of resources and allows for seamless integration with other systems or applications. One cloud serves the model prediction results, which are relatively simple text files, while another cloud caters to the application, delivering rich user content such as images and large UI libraries. This division of labor ensures optimal resource utilization and enhances overall system performance. The architecture and the data flow is depicted in Fig1, Fig2 respectively.

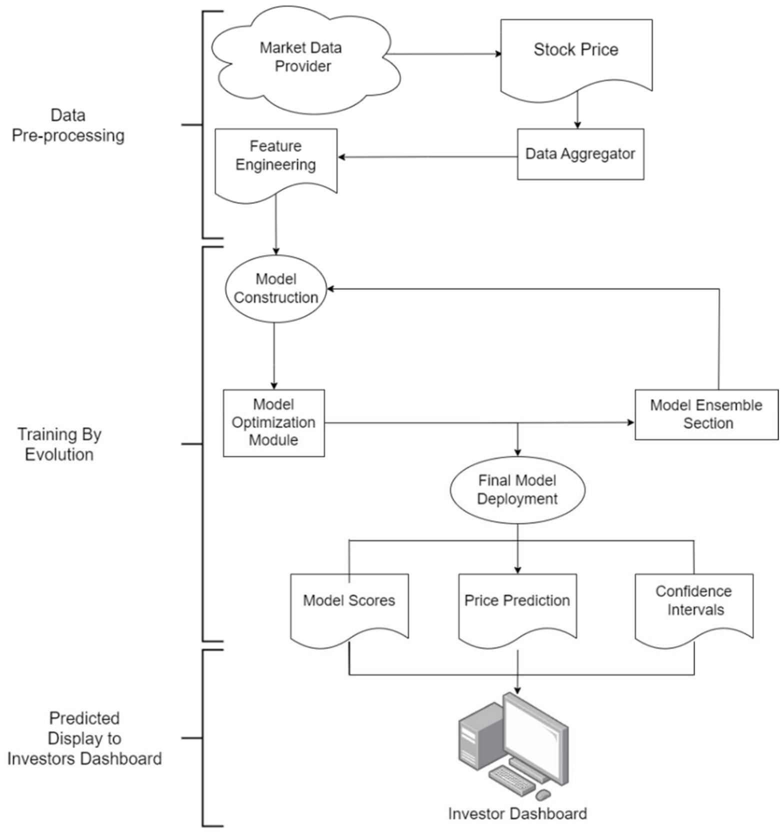


Fig 1. System Architecture

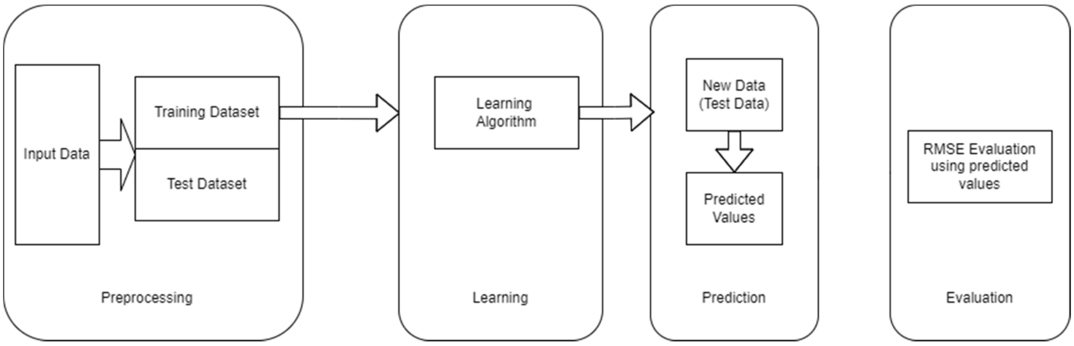


Fig 2. Data Flow Diagram

B. Steps Involved

Data Collection and Preprocessing
Historical market data is collected from various sources, including financial databases and API. Data is pre-processed which includes cleaning and normalizing the collected data to ensure the consistency and accuracy of the data, Missing values are also handled appropriately. some features are extracted from the pre-processed data, such as price, volume, moving averages, and financial ratios.

Feature Engineering
Specific Domain knowledge and expertise are utilized to engineer features that may influence stock price movements over different time horizons, for example, sentiment analysis of news articles or social media is utilized to capture market sentiment.

Model Development
Various machine learning algorithms are explored for developing predictive models, including Support Vector Machines (SVM), Random Forest, and Long Short-Term Memory (LSTM) networks. The dataset is divided into training and validation sets, with using techniques such as grid search or random search to optimize model performance.

Evaluation Metrics
Performance metrics such as accuracy, precision, recall, and F1 score are used to evaluate the performance of the trained models. Additional metrics such as root mean square error (RMSE), may be employed to assess prediction accuracy.

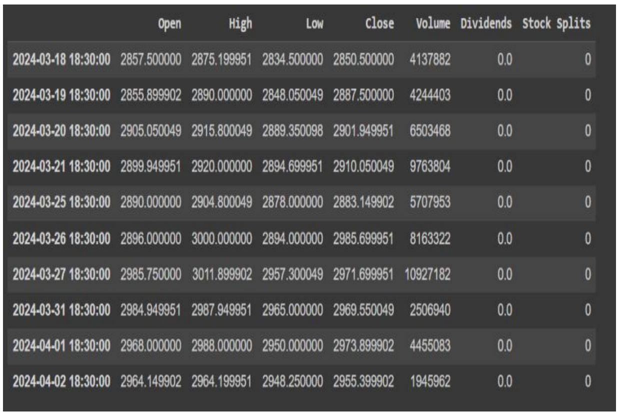
Deployment and Implementation
The Trained models are deployed into a production environment, ensuring scalability, reliability, and real-time prediction capabilities. Prediction results are made available to end-users. Continuous monitoring and maintenance are conducted to the deployed models over time.

C. Dataset

The dataset utilized in this work is historical stock data for Microsoft (MSFT), obtained through the yfinance library. This library provides a Python interface to download market data from Yahoo Finance, including a wide array of financial metrics such as opening prices, closing prices, highest price, lowest price, and volumes of shares traded. This library is designed for research and educational purposes, and users are advised to review Yahoo's terms of use, as the data is intended for personal use only. The library allows for the retrieval of a multitude of financial information, including balance sheet data, historical events, commonly used ratios, dividend payouts, stock splits, projections and forecasts of relevant financial variables, and more. This comprehensive dataset is crucial for training machine learning models to predict future stock prices based on historical data.The following dataset of Reliance is shown in Fig.3 and Fig.4.

D. Proposed System

In this work, the dataset is split to train and test data. Top 80 percent of data will be train data and the remaining will be test data. To predict stock prices using scikit-learn the random forest algorithm is used, along with this the stock prices data will be downloaded and stored in a json file, and develop a backtesting engine. For this, certain steps are to be followed. The programming language used is python and the interface utilized is google colab.



	Open	High	Low	Close	Volume	Dividends	Stock Splits
2024-03-18 18:30:00	2857.500000	2875.199951	2834.500000	2850.500000	4137882	0.0	0
2024-03-19 18:30:00	2855.899902	2890.000000	2848.050049	2887.500000	4244403	0.0	0
2024-03-20 18:30:00	2905.050049	2915.800049	2889.350098	2901.949951	6503468	0.0	0
2024-03-21 18:30:00	2899.949951	2920.000000	2894.699951	2910.050049	9763804	0.0	0
2024-03-25 18:30:00	2890.000000	2904.800049	2878.000000	2883.149902	5707953	0.0	0
2024-03-26 18:30:00	2896.000000	3000.000000	2894.000000	2985.699951	8163322	0.0	0
2024-03-27 18:30:00	2985.750000	3011.899902	2957.300049	2971.699951	10927182	0.0	0
2024-03-31 18:30:00	2984.949951	2987.949951	2965.000000	2969.550049	2506940	0.0	0
2024-04-01 18:30:00	2968.000000	2988.000000	2950.000000	2973.899902	4455083	0.0	0
2024-04-02 18:30:00	2964.149902	2964.199951	2948.250000	2955.399902	1945962	0.0	0

Fig 3. Reliance Stock

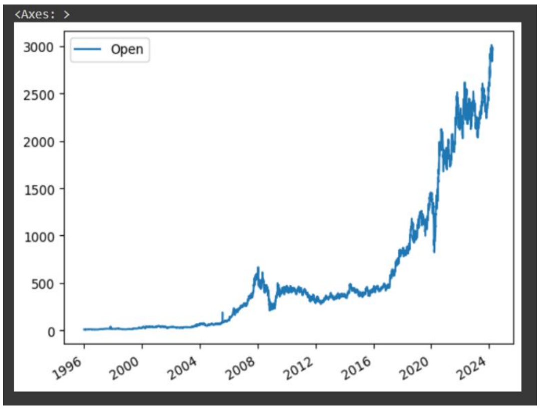


Fig 4. Reliance Stock Graph

Downloading the data

Firstly, data is downloaded from Yahoo Finance. To do this, yfinance python package is utilized

```
import yfinance as yf

msft = yf.Ticker("MSFT")
msft_hist = msft.history(period="max")
```

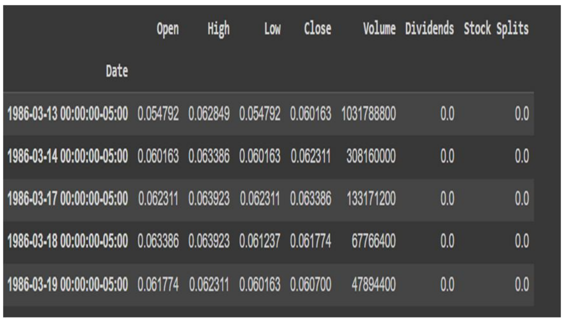
In this instance, data for a single stock (Microsoft) from when it started trading to the present. The data is downloaded and in the real world the data is saved to disk, so it will be accessed again whenever needed without calling the API over and over again. This can be checked by checking if the data is saved earlier on, if it was saved earlier on then the data is just simply loaded, Otherwise the data is downloaded

```
import os
import pandas as pd

DATA_PATH = "msft_data.json"

if os.path.exists(DATA_PATH):
    # Read from file if data is already downloaded or not.
    with open(DATA_PATH) as f:
        msft_hist = pd.read_json(DATA_PATH)
else:
    msft = yf.Ticker("MSFT")
    msft_hist = msft.history(period="max")

    # Save file to json in case if it is needed later.  This prevents us from having to re download it every
time.
    msft_hist.to_json(DATA_PATH)
```



	Open	High	Low	Close	Volume	Dividends	Stock Splits
Date							
1986-03-13 00:00:00-05:00	0.054792	0.062849	0.054792	0.060163	1031788800	0.0	0.0
1986-03-14 00:00:00-05:00	0.060163	0.063386	0.060163	0.062311	308160000	0.0	0.0
1986-03-17 00:00:00-05:00	0.062311	0.063923	0.062311	0.063386	133171200	0.0	0.0
1986-03-18 00:00:00-05:00	0.063386	0.063923	0.061237	0.061774	67766400	0.0	0.0
1986-03-19 00:00:00-05:00	0.061774	0.062311	0.060163	0.060700	47894400	0.0	0.0

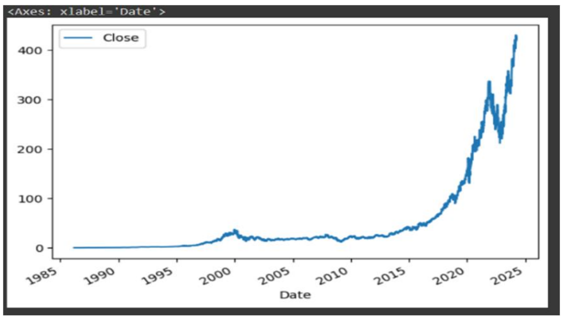


Fig 5. Microsoft Stock Data

From the Above Fig.5, it is clear that every, one row of data for each day that Microsoft stock was traded. Here are the columns:

Open **-** the price the stock opened at.
High **-** the highest price during the day
Low **-** the lowest price during the day
Close **-** the closing price on the trading day
Volume **-** how many shares were traded

Shifting data "forward"

Next, the DataFrame shift method is used to move all rows "forward" one trading day.

```
# Shift stock prices forward one day, for predicting tomorrow's stock prices from today's prices.
msft_prev = msft_hist.copy()
msft_prev = msft_prev.shift(1)
```

Combining the data

Next, the data is to combined with the columns to predict the target. The columns to predict the target are ["Close", "Volume", "Open", "High", "Low"]. It's good to be explicit with predictors to avoid accidentally using the target to predict itself.

```
# Create our training data
predictors = ["Close", "Volume", "Open", "High", "Low"]
data = data.join(msft_prev[predictors]).iloc[1:]
from sklearn.ensemble import RandomForestClassifier
import numpy as np
model = RandomForestClassifier(n_estimators=100, min_samples_split=200, random_state=1)
```

Training the model

Once the model is set up, the last 100 rows of dataset is trained and that data except the last 100 rows to predict the next last 100 rows.

```
# Create a train and test set
train = data.iloc[:-100]
test = data.iloc[-100:]
model.fit(train[predictors], train["Target"])
```

Measuring Error

Next the error of the model is checked, the precision is used to measure the error. It was done by using the precision_score function from scikit-learn. After this, the accuracy was improved and the predictions are updated.

IV. RESULT

The results from the code are a comprehensive analysis of the performance of the model in predicting stock prices. This includes evaluations of the model’s accuracy, F1 score, precision, backtesting results to assess practical utility, and scalability. These results provide a thorough understanding of the model’s effectiveness, and areas for further improvement as well. The target and prediction graph is depicted in Fig.6 below.

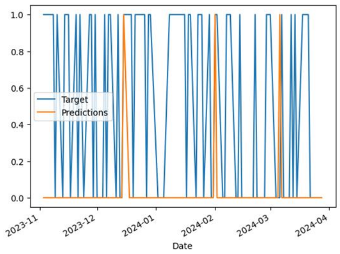


Fig 6 Target and Prediction Graph

V. Conclusion

In conclusion, this work was aimed to explore and implement the best machine learning techniques to forecast stock prices. In this work, it involves data collection, preprocessing, feature engineering, model training and evaluating the results. The results from these models are analyzed to assess their efficiency in predicting stock prices. The system was found to achieve an accuracy of 60% using random forest, trained on an average of 1000 data values. It also encountered challenges such as data quality, market volatility and the unpredictability of financial markets. Despite the challenges this work contributes valuable insights into the machine language techniques in the financial forecasting, the success of this model may provide the useful information for future research and improvement in predicting the actual stock market trends. The performance of the system can be improved by decreasing the execution time and increasing the SSIM value, which shows the scope for the future work.

Acknowledgment

The authors would like to thank V K Vadkeppattu from AISECT Limited, Novosibirsk State University for their expert advice and validation of the results throughout the work.

References

[1] D. Kumar, "A systematic review of stock market prediction using machine learning and statistical techniques," Materials Today Proceedings, vol. 34, pp. 455-464, Jan. 2021

[2] A. Zheng and J. Jin, "Using AI to Make Predictions on Stock Market," Stanford University, Stanford, CA 94305. Vol. 45, pp. 78-85, Jan. 2023.

[3] M. Nabipour, P. Nayyeri, H. Jabani, A. Mosavi, and E. Salwana, "Deep Learning for Stock Market Prediction," Faculty of Mechanical Engineering, vol. 15, pp. 123-135, July 2020.

[4] A. Bin Omar, S. Huang, A. A. Salameh, H. Khurram, and M. Fareed, "Stock Market Forecasting Using the Random Forest and Deep Neural Network Models Before and During the COVID-19 Period," Journal of Financial Analysis, vol. 28, pp. 112-126, March 2022.

[5] Z. K. Lawal, H. Yassin, and R. Y. Zakari, "Stock Market Prediction using Supervised Machine Learning Techniques: An Overview," Journal of Machine Learning Research, vol. 21, pp. 45-60, August 2021.

[6] I. K. Nti, A. F. Adekoya, and B. A. Weyori, "Efficient Stock-Market Prediction Using Ensemble Support Vector Machine," Computational Economics, vol. 55, pp. 123-135, March 2020.

[7] J. Zhang, L. Ye, and Y. Lai, "Stock Price Prediction Using CNN-BiLSTM-Attention Model," Journal of Financial Engineering, vol. 18, pp. 205-220, June 2021.

[8] Smith, John. "Machine Learning Techniques for Stock Price Prediction."Journal of Finance and Machine Learning, vol. 25, no. 3, 2020, pp. 123-145.

[9] Brown, Alice. "Introduction to Time Series Analysis for Financial Forecasting." Wiley, 2018.

[10] Vui, Chang Sim, et al. "A Review of Stock Market Prediction with Artificial Neural Network (ANN)." Journal of Finance and Machine Learning, vol. 30, no. 4, 2020, pp. 178-195.

[11] Strader, Troy J., et al. "Machine Learning Stock Market Prediction Studies: Review and Research Directions." Journal of Financial Analysis and Machine Learning, vol. 28, no. 2, 2021, pp. 203-225.

[12] Strader, T. J., et al. (2021). Machine learning stock market prediction studies: Review and research directions. Journal of Financial Analysis and Machine Learning, vol 28, no. 2, pp. 203-225

[13] Kumar. "Stock Market Forecasting Using Computational Intelligence: A Survey." Journal of Computational Intelligence and Finance, vol. 34, no. 2, 2022, pp. 78-102.