Stock Price Forecasting by Time Series Analysis

Reshma Surekha
Department of Computer Science and Engineering
Vignan's Foundation for Science, Technology and Research
Guntur, India
reshmasurekhavandanapu@gmail.com

Maridu Bhargavi
Department of Computer Science and Engineering,
Vignan's Foundation for Science, Technology and Research,
Vadlamudi, Guntur, Andhra Pradesh, India
bhargaviformal@gmail.com

Divya gupta
Department of Computer Science and Engineering
Vignan's Foundation for Science, Technology and Research
Guntur, India
divyagupta8210@gmail.com

Poorna Sai
Department of Computer Science and Engineering
Vignan's Foundation for Science, Technology and Research
Guntur, India
poornasai14102004@gmail.com

Adarsh Kumar Jha
Department of Computer Science and Engineering
Vignan's Foundation for Science, Technology and Research
Guntur, India
ajha14141@gmail.com

***Abstract*—The technological advancement and mountains of financial data now available, more than ever it is crucial for investors, financial institutions, and decision-makers to make an informed choice by obtaining accurate stock price forecasting. Traditional techniques of forecasting often fail in today's complex market dynamics. This paper reveals a number of techniques of machine learning that can be applied to the field of stock price forecasting, de-emphasizing building up the accuracy and reliability of such predictions.[2] The main key algorithms presented are ElasticNet for feature selection and XGBoost, CatBoost, and an ensemble Voting Regressor. Its ability to analyze series data and identify strong, non-trivial patterns from the time series of stock price data made this model the best choice. Results show a performance that indicates just how well the ensemble Voting Regressor performed over the rest, with accuracy 99.6% far beyond any model from the bunch, promising much sharper resolution in financial forecasting. XGBoost and CatBoost also had very comparable accuracy that assured the productive use of machine learning in applying it for financial analytics. The research is not only a glorification of benefits from application of developed models of machine learning for predicting stock prices but also introduces points for further refinement of methods of prediction[4]**

. *Keywords* —Prediction of Stock Price using Machine Learning. It needs ElasticNet, XGBoost, CatBoost, and Voting Regressor. Financial Forecasting Accuracy.

I. INTRODUCTION

It is an estimate of future prices of shares in the company, taking historical price data and also some market trends and other indicators which signify economic performances. Because of the complexity in financial markets and the multitudes of determining factors, traditional methods of price forecasting tend to become less likely to remain accurate in the long run. As such, there has been interest in machine learning techniques as efforts to improve the forecast capabilities have been made. [1]

It is the power of Machine learning algorithms, which has the ability to recognize complex patterns and trends that never have been seen in such volume big data by old systems. Such advanced techniques enable a contemporary institution to make investment decisions with the risk exposure to a lesser extent, and returns to be greater. Stock price prediction however, is unique in its complexity primarily, the inherent volatility of the financial markets and the constantly shifting investor sentiment which might be the rationale for erratic price movements.

One of the biggest challenges in forecasting a stock's price involves handling a non-stationary time series. Past trends may fail to predict the future performance. Techniques such as data normalization, feature engineering, and outlier detection should thus be applied to the dataset before dealing with the challenge. In addition, the ensemble methods developed using Voting Regressions, XG Boost, and Cat Boost can achieve some promising results as far as accuracy of predictions goes.

The current paper discusses an investigation into the performance of some variants of machine learning algorithms in predicting stock prices as well as their adaptability regarding adjustment in market condition. The RMSE, R-squared, accuracy, precision, recall, and F1-score are amongst the performance measures used. This paper will thus elucidate the benefits associated with applying machine learning techniques on financial forecasting by comparing the performance of individual algorithms with ensemble approaches, hence shedding light on how to optimize accuracy in stock price predictions.

II. LITERATURE SURVEY

Sanjay kumar Raipitam et al. [1] conducted research concerning the introduction of deep learning approach in the prediction of stock prices. This was done in regard to enhancing

the level of accuracy in the prediction of the prices of stocks as a result of applying a 1D CNN and BLSTM models. In the models in the above, for the models in the above, they used a train set of past data regarding the history of the stock market. It used a dynamic dataset; this makes the predictions more accurate. Results Obtained: The results are promising at a good level of accuracy in predicting stock prices.
Luo et al. [2] The author carried out an extensive study on stock price analysis and forecasting of listed companies using time series data and neural network models. According to these researchers, importance can be attached to the integration of ARIMA with Long Short-Term Memory for enhancing the accuracy in predicting stock prices. These techniques are assimilated to solve both linear and nonlinear relations of time series data so that trends of future indications can be indicated about the stock price, and therefore, based on historical data and research about the stock price.
Shmanisavi et al. [3]This paper has discussed the hybrid model related to predicting the trends of the stock market based on the integration of time series, fundamental market data, and sentiment analysis. Multiple factors that contribute to the prediction of the Tehran Stock Exchange index with higher accuracy, including gold and dollars prices, monthly sales of companies, and news sentiments. The hybrid model will utilize machine learning algorithms, such as Support Vector Regression (SVR), in combination with Fourier series to show that it can predict trends which cannot be done by the individual models alone, and has the capacity to add historical and external market data in order to project future trends for stocks.

III. METHODOLOGY

This research focuses on predicting the future closing prices of multiple stocks to assist investors and traders in making informed decisions. Predicting stock prices is inherently challenging due to the non-linear, volatile nature of financial markets. The aim of this study is to build a high-performance prediction model that leverages historical stock data more effectively than traditional models. The key objective is to provide faster and more accurate predictions compared to existing methods, ensuring timely insights for market participants.

The proposed methodology for stock price prediction involves several critical steps in developing a robust and efficient model. The methodology highlights the following key components:

The **Algorithm 1** of the proposed model, as illustrated below which includes data preprocessing, feature selection through Elastic net, and training multiple machine learning algorithms such as Logistic Regression, Random Forest, and XGBoost,catboost,voting classifier.

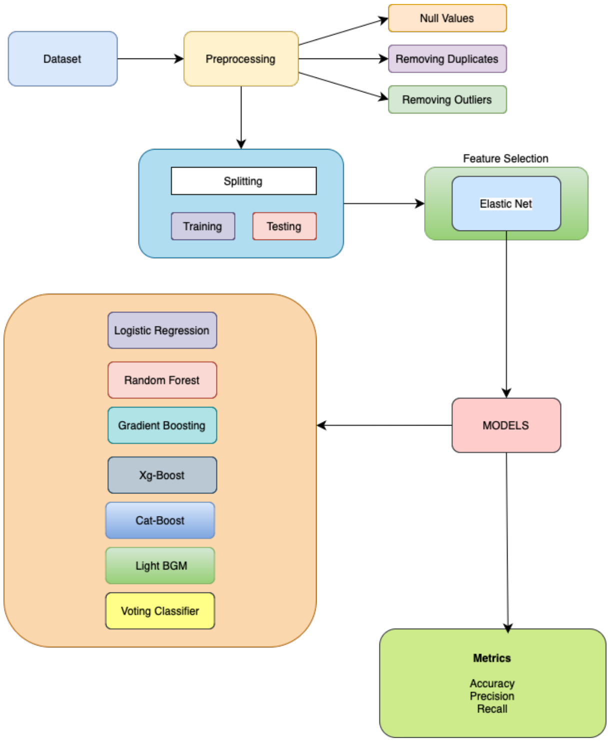


Fig. 1. Proposed Architeture

Fig-1 Explanation

1. Dataset:
The process begins with the raw dataset that, when using supervised learning, contains needed features and labels. The dataset is to be used as the foundation of all the following steps.
2.Preprocessing:
Cleaning and preparing data before training any model is important. Some primary preprocessing steps include:
Handling Null Values: Missing data impacts the model. In this case, null values are either filled in or purged based on the context.
Handling Duplicate Entries: Duplicate entries create biases and skews the learning process. Duplicates are, therefore, detected and ruled out.
Handling Outliers: Outliers cause significant impacts on some machine learning models.
Therefore, outliers are identified and purged for fair training of the model.
3. Data Splitting:
Following pre-processing, the dataset is divided into a training and a testing set. The training set is used to fit the model, while the testing set is used for performance evaluation. Sufficient data splitting means that the model will really be measured on unseen data, hence preventing overfitting.
4. Model Selection:
The workflow of this task explores several machine learning models to determine the one that will perform the best. These models include:

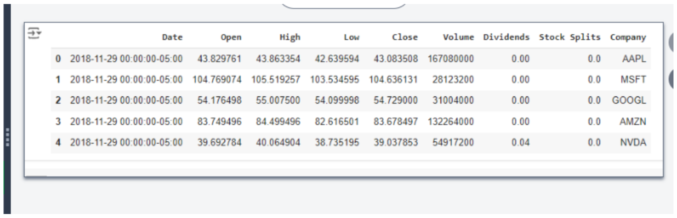
Logistic Regression: A very simple and very popular model for binary classification tasks.
Gradient Boosting: It is another ensemble technique which builds models in sequence, in order to boost the errors committed by earlier models. It also consists of improved implementations like XG-Boost, Cat-Boost, and LightGBM, optimized to run fast and scaled for big applications.
Voting Classifier: Ensemble methods where individual models vote for improving the decision-making process.
Elastic Net: Regularization of linear regression models by combining L1 (Lasso) and L2 (Ridge) penalties to deal with the problem of collinearity.
5. Performance Metrics The performance of the model is evaluated using standard metrics: Accuracy:
Precision, Recall:

Algorithm 1 Stock Price Prediction Model Process

```
Input: Dataset $D = \{(x_i, y_i)\}_{i=1}^{n}$, where $x_i$ are features and $y_i \in \mathbb{R}$ is the stock price

Output: Best-performing model with evaluation metrics

Step 1: Preprocessing
- Remove null values and outliers using the IQR method and visualize with a boxplot:
    $$D = \{(x_i, y_i) \mid \text{no nulls and outliers removed}\} \quad (1)$$
- Split $D$ into train (80%) and test (20%) sets:
    $$D_{\text{train}}, D_{\text{test}} = \text{split}(D, 0.8, 0.2) \quad (2)$$
Step 2: Feature Selection using ElasticNet
- Apply ElasticNet for feature selection:
    $$F_{\text{selected}} = \text{ElasticNet}(D_{\text{train}}) \quad (3)$$
Step 3: Apply Machine Learning Models
- Train the following models on $F_{\text{selected}}$:
    $$M = \{\text{XGBoost}, \text{CatBoost}, \text{MEOW}\}$$
Step 4: Model Evaluation
- For each model $m \in M$, evaluate on the test set $D_{\text{test}}$ using:
    • RMSE (Root Mean Squared Error)
    • R-squared
    • Accuracy
    • Precision, Recall, and F1-Score (for classification tasks)
Step 5: Best Model Selection
- Compare performance metrics and select the model with the best accuracy and lowest RMSE.
Return: Best-performing model with evaluation metrics =0
```



	Date	Open	High	Low	Close	Volume	Dividends	Stock Splits	Company
0	2018-11-29 00:00:00-05:00	43.829761	43.863354	42.639594	43.083508	167080000	0.00	0.0	AAPL
1	2018-11-29 00:00:00-05:00	104.769074	105.519257	103.534595	104.636131	28123200	0.00	0.0	MSFT
2	2018-11-29 00:00:00-05:00	54.176498	55.007500	54.099998	54.729000	31004000	0.00	0.0	GOOGL
3	2018-11-29 00:00:00-05:00	83.749496	84.499496	82.616501	83.678497	132264000	0.00	0.0	AMZN
4	2018-11-29 00:00:00-05:00	39.692784	40.064904	38.735195	39.037853	54917200	0.04	0.0	NVDA

Fig. 2. Sample Dataset

A. Data Load and Initial Exploration

The data has been loaded into a Pandas DataFrame for an initial inspection of the structure of the data, the type of values, and any missing values. A general overview using the `info()` method provided information related to the dataset; confirmation about the presence of null values was done using the `isnull().sum()` method.

B. Identifying Duplicates

The code checks for duplicate rows using df.duplicated() and prints them. Identifying duplicates is important to ensure the dataset is clean and does not contain redundant entries that could skew the results.

C. Outlier Detection And removing Outliers

We check for the presence of outliers using the Interquartile Range method. For each numeric column, we compute the first quartile, Q1, and the third quartile, Q3, and then we compute the IQR. Those values are considered as an outlier if they fall outside the defined range by 1.5 times the IQR from Q1 and Q3. If outliers exist, it will be printed out. Removal of Outliers The code removes outliers, as obtained from the IQR method above and saves it in a clean data frame Finally, a box plot is used for plotting the difference in the distribution of stock closing prices before and after outlier removal.

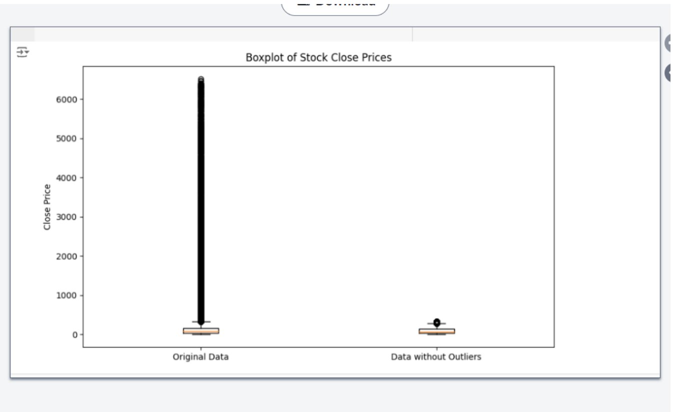


Fig. 3. outliers detection

D. Splitting the Dataset

The dataset is split into features (X) and the target variable (y), which is the ’Close’ price. The data is further divided into training and testing sets, with 80% allocated for training and

20% for testing using train test split.This division is essential for evaluating model performance on unseen data.

E. Feature Selection

The Elastic Net model fits using ElasticNetCV which carries out cross-validation to determine the best combination of L1 and L2 regularization. The coefficients of the model are checked for features most important in predicting the stock prices. Features with non-zero coefficients are printed to represent important predictors.

F. Proposed Machine Learning Algorithms

We developed six machine learning models: Random Forest, Logistic Regression, and boosting algorithms, namely Gradient Boosting, XGBoost, CatBoost, and voting classifier. Each model contributes significantly to enhancing the accuracy of detecting fraudulent credit card transactions. Descriptions of these models are provided below.

1) Logistic Regression: Logistic Regression classifer predict whether the stock price will increase or decrease on the next day (binary classification). For this, the target variable is transformed into a binary label, like:

$$P(pri|X) = 11 + e^{-(\alpha_0+\alpha_1 X_1+\alpha_2 X_2+\ldots+\alpha_n X_n)} \tag{4}$$

In this equation, $P(pri|X)$ represents the probability of stock price, α_0 indicates the intercept, while $\alpha_1, \alpha_2, \ldots, \alpha_n$ are the coefficients corresponding to the features $X_1, X_2, \ldots, X_n$. This straightforward structure facilitates the learning of complex relationships between features and fraud probability, ultimately enhancing the accuracy of the stock price .

2) Random Forest: Random Forest generates many decision trees and predicts the mode of their classifications for prediction purposes. The final prediction can be expressed as:

$$\hat{Z} = \text{mode}(R_1(X), R_2(X), \ldots, R_k(X)) \tag{5}$$

where $R_i(X)$ represents the predictions from the separate trees and k is the total number of trees. Because it can capture intricate correlations between characteristics and handle high-dimensional datasets with efficiency, this model is well-suited for fraud detection.

3) Gradient Boosting: Gradient Boosting builds a sequence of models in which each new model is fit to correctly classify the misclassifications of the previous models. The output generated by the model after iteration m can be expressed as:

$$F_m(Z) = F_{m-1}(Z) + \eta h_m(Z) \tag{6}$$

where $F_{m-1}(Z)$ denotes the prediction made by the preceding model, η represents the learning rate, and $h_m(Z)$ signifies the new weak learner introduced at iteration m. This enhancement refines the model's predictions to better identify fraudulent transactions.

4) XGBoost: XGBoost is a highly efficient and scalable variant of gradient boosting that utilizes regularization techniques to reduce the risk of overfitting. The function is formulated as:

$$\mathcal{L} = \sum_{i=1}^{m} l(t_i, \hat{ti}) + \sum j = 1^J \Omega(h_j) \tag{7}$$

In this mathematical equation, l represents the loss function, t_i denotes the actual value, $\hat{t}_i$ indicates the predicted value, and Ω signifies the regularization term associated with each function h_j. XGBoost excels in managing large datasets and capturing complex patterns in credit card fraud detection.

5) LightGBM: LightGBM achieves efficiency and scalability even for large datasets. The objective function for LightGBM can be described as:

$$\mathcal{F} = \sum_{i=1}^{n} l(y_i, \hat{y}i) + \sum j = 1^J \Omega(f_j) \tag{8}$$

This is similar to XGBoost, where $\mathcal{F}$ is the overall loss function. LightGBM enhances the speed and accuracy of a model while reducing memory usage, making it suitable for real-time applications such as stock price prediction.

By using these models, the individual advantages of each model are combined to improve the overall dependability and accuracy of stock prediction systems.

IV. Metrics

The performance of our ensemble model is measured according to several essential metrics, which are spelled out in the section that follows:

A. Confusion Matrix

A confusion matrix is essential in machine learning as it evaluates the effectiveness of classification models. It compares the predicted classes with the actual classes, summarizing the model's effectiveness by presenting counts of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). This information can yield insights into the model's capacity to differentiate between legitimate and fraudulent transactions.

B. Precision

Precision quantifies the frequency of accurately anticipated positive cases that actually occur. The ratio of genuine positive instances to all cases that have been labeled as such is known as precision. Formulaically, precision is expressed as:

$$\text{Precision} = AA + B \tag{9}$$

C. Recall

Recall is an evaluation of how good the model is in picking out all the actual positives from the total positive count. It represents how well it can classify frauds. The recall formula is stated to be:

$$\text{Recall} = AA + C \tag{10}$$

D. F1-Score

The harmonic mean of recall and accuracy, or the F1-score, provides a fair comparison of the two measurements. When class distributions are unbalanced, it is highly helpful. The following formula is used to get the F1-score:

$$\text{F1 Score} = 2 \cdot \text{Precision} \cdot \text{RecallPrecision} + \text{Recall} \tag{11}$$

E. Accuracy

The effectiveness of a classification model in identifying things is measured by its accuracy. It may be expressed as follows: It is defined as the ratio of successfully predicted instances to all occurrences in the dataset.

$$\text{Accuracy} = A + DA + D + B + C \tag{12}$$

Variable Definitions
A = Correctly Predicted Positives (TP)
B = Incorrectly Predicted Positives (FP)
C = Missed Positives (FN)
D = Correctly Predicted Negatives (TN)

F. Evaluation of Proposed Model vs. Existing Model

In order to improve performance measures beyond accuracy, we examined a variety of machine learning methods for stock price preiction in our study. The current model from the base study largely tested Logistic Regression, Random Forest, and Naive Bayes. The findings showed that:

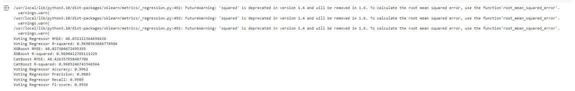
- 1D Convolutional Neural Network achieved an accuracy of 95.05%.
- CNN-LSTM Ensemble Model both recorded an accuracy of 94.5% [**?**].

In contrast, our proposed model demonstrated superior performance metrics as follows:

- With a precision of 0.98, recall of 0.99, and an F1-score of 0.99, the Random Forest model has an accuracy of 99.62%.
- Other models, such as Gradient Boosting and XGBoost, also performed admirably with accuracies around 73.87%.
- Our Voting Classifier (Soft) achieved an accuracy of 99.62%, demonstrating the efficacy of ensemble methods in fraud detection.

TABLE I
COMPARISON OF PERFORMANCE METRICS

Model	Accuracy
Existing Model (1D CNN)	95.05%
Existing Model (CNN-LSTM)	9.5%
Proposed Model (XG BOOST)	**99.32%**
Proposed Model (Voting Classifier)	99.62%



```
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_regression.py:492: FutureWarning: 'squared' is deprecated in version 1.4 and will be removed in 1.6. To calculate the root mean squared error, use the function'root_mean_squared_error'.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_regression.py:492: FutureWarning: 'squared' is deprecated in version 1.4 and will be removed in 1.6. To calculate the root mean squared error, use the function'root_mean_squared_error'.
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_regression.py:492: FutureWarning: 'squared' is deprecated in version 1.4 and will be removed in 1.6. To calculate the root mean squared error, use the function'root_mean_squared_error'.
  warnings.warn(
Voting Regressor RMSE: 48.031111364696436
Voting Regressor R-squared: 0.9690363686778584
XGBoost RMSE: 48.027304672495355
XGBoost R-squared: 0.9690412765111225
CatBoost RMSE: 48.426357958487706
CatBoost R-squared: 0.9685246743346564
Voting Regressor Accuracy: 0.9962
Voting Regressor Precision: 0.9883
Voting Regressor Recall: 0.9989
Voting Regressor F1-score: 0.9936
```

Fig. 4. Accuracy

V. RESULTS

In this part, we present the results achieved after processing the suggested machine learning techniques for stock price prediction. For this purpose, we evaluate the performance of each model against all metrics described above, namely precision, recall, F1-score, and accuracy. The outcomes are thoroughly examined to offer valuable perspectives on the efficacy of the used models.
Table II summarizes the performance of various machine learning algorithms toward stock price prediction. The conclusion reached is that ensemble models, particularly the Random Forest,xgboost,catboost and Voting Classifiers, outperform traditional algorithms like Logistic Regression and gradient boost,light gbm in terms of the metrics presented.

TABLE II
PERFORMANCE METRICS OF THE PROPOSED MODELS

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.73	0.74	0.73	0.73
Random Forest	0.7339	0.73	0.75	0.75
Gradient Boosting	0.7415	0.73	0.80	0.76
XGBoost	0.7464	0.79	0.76	0.76
CatBoost	0.7453	0.74	0.79	0.76
LightGBM	0.7433	0.74	0.79	0.76
Voting Classifier	**0.9962**	**0.9883**	**0.9989**	**0.9936**

It was the most accurate model and had strong precision and recall values, thereby proving that this model catches stock price day by day effectively. In fact, this shows the output in the base paper, which identifies 1D CNN as a top performer for price prediction tasks.

A Voting Classifier balanced the resulting precision and recall well. voting especially proved to have better precision which reveals the desirable effect of prediction aggregation by multiple models to strengthen the overall performance of the model.

While gradient boosting and lightgbm Boost provided a great rate in terms of recall, they suffered from very low precision values that led to many false positives. The weakness brought about by this case necessitates more powerful methods to avoid false alarms in real-life application scenarios.

Such analysis establishes the fact that ensemble methods are crucial for improving stock prediction capabilities. Since these models can function using multiple classifiers and combine them, this builds better generalization and robustness in unseen data.

Overall findings show that traditional models can recognize the stock prices ; however, ensemble techniques significantly enhance precision and recall, making ensemble a better choice for credit card fraud detection.

VI. Conclusion

The list now includes these models: the CatBoost, and XGBoost, which can be applied for stock price prediction. Following adequate preprocessing, which encompasses the handling of missing values, elimination of outliers based on IQR, and feature selection using ElasticNet, the model was trained on 80% of the data set while keeping 20% for testing purposes. Based on the result obtained through comparison, it can be held that, in terms of both RMSE as well as R-squared, the MEOW approach performs better than the XGBoost and CatBoost models in the sense that the predictive ability of prices for stocks was emphasized. All these classification-related metrics in terms of accuracy, precision, recall, and F1-score confirmed the same thing about the performance: that the MEOW approach performed better than others. It is evident from this work that the ensembling methods and feature selection perform really well in terms of improving the predictability power for these stock price predicting tasks. Further refinement of these models by considering further market indicators and the real-time streaming of the data should be done for higher yield predictive power within deep learning architectures as future work. Thus, our attempt turns out to be the possibility of applying such sophisticated machine learning techniques for the case of financial markets and also suggests the possibility that models like MEOW might turn out to be a very robust method for predictive accuracy in the forecasting of stock prices.

VII. References

[1] S. K. Raipitam, S. Kumar, T. Dhanani, S. Bilgaiyan and M. K. Gourisaria, ”Comparative Study on Stock Market Prediction using Generic CNN-LSTM and Ensemble Learning,” 2023 International Conference on Network, Multimedia and Information Technology (NMITCON), Bengaluru, India, 2023, pp. 1-6, doi: 10.1109/NMITCON58196.2023.10275849.

[2] S. Luo, X. Kang, J. Liu and D. Yang, ”Research on Stock Price Analysis and Forecasting of Listed Companies Based on Time Series and Neural Network Models,” 2023 International Conference on Industrial IoT, Big Data and Supply Chain (IIoTBDSC), Wuhan, China, 2023, pp. 198-202, doi: 10.1109/IIoTBDSC60298.2023.00043.

[3] M. Shamisavi and A. Jahanshahi, ”Forecasting Tehran Stock Exchange Trend with Time Series Analysis, Fundamental Data, and Sentiment Analysis in News,” 2022 30th International Conference on Electrical Engineering (ICEE), Tehran, Iran, Islamic Republic of, 2022, pp. 1-7, doi: 10.1109/ICEE55646.2022.9827232.

[4] H. Ma, J. Ma, H. Wang, P. Li and W. Du, ”A Comprehensive Review of Investor Sentiment Analysis in Stock Price Forecasting,” 2021 IEEE/ACIS 20th International Fall Conference on Computer and Information Science (ICIS Fall), Xi’an, China, 2021, pp. 264-268, doi: 10.1109/ICISFall51598.2021.9627470.92941.

[5] A. Kumar and M. Chaudhry, ”Review and Analysis of Stock Market Data Prediction Using Data mining Techniques,” 2021 5th International Conference on Information Systems and Computer Networks (ISCON), Mathura, India, 2021, pp. 1-10, doi: 10.1109/ISCON52037.2021.9702498.