



A Comparative Analysis of Machine Learning Models for Stock Price Prediction

¹Amey Nandgaonkar, ²Anil Nandgaonkar

¹Asst Professor, ²Professor

¹Electronics & Telecommunication Engineering

¹VJTI Mumbai, India

Abstract: Trading holds a pivotal role in the global financial market, being one of its most crucial activities. The inherent unpredictability of stock market forecasts adds to the complexity of this task. However, various approaches and methods that are used for making predictions about the stock market. The prevalent method for achieving accurate stock market predictions involves the utilization of AI and ML-based techniques. This analysis leads to the conclusion that stock market prediction is a highly intricate task, necessitating the consideration of various factors to enhance accuracy and efficiency in forecasting the market's future.

A comprehensive literature survey has been conducted, revealing a diverse range of methods available for predicting the stock market. In this paper, stock market prediction, using artificial intelligence and machine learning is depicted by analyzing the stock market. Various algorithms and different tools have been used for prediction and applied that understanding to make a AI and ML model to predict the future stock prices.

IndexTerms- Stock Market, Prediction, AI/ML models, Data Sources, Python, LSTM, LGR, SVC, ANN, Regression algorithm.

I. INTRODUCTION

The growth of economies throughout the world is heavily influenced by the financial industry. The stock market has become increasingly popular among the masses as it continues to rise in economic significance [Ritesh Tandon] (2024). Because of the stock market's volatility and vulnerability to outside factors like social, political, and economic developments, it is still challenging to make accurate predictions. To overcome these difficulties, numerous studies have been conducted over the past decades to predict various types of financial time-series data [Hyunsun Song et al] (2022). Innovative technologies like artificial intelligence (AI) and machine learning (ML) offer investors useful information and assistance in making decisions. Further, researchers are constantly refining predictive models using these tools, aiming to understand and navigate the complex dynamics of the market. The goal is to develop software and programs that help investors anticipate trends, manage risk, and optimize their portfolios for maximum returns. While complete accuracy may be elusive, these advancements are paving the way for more reliable and informed investment strategies in the future.

Overall, predicting the stock market remains a complex and dynamic field. While complete accuracy is a distant dream, the integration of AI and other advanced technologies is opening up new avenues for investors to gain valuable insights and make informed decisions. Different technical factors like volume and maximum and minimum prices per trading period are used for analysis [Zahid Iqbal et al] (2013). Continued research and development in this area are crucial for shaping the future of stock trading and empowering investors to navigate the market with greater confidence and potential success.

Background and Related Work

Stock market prediction has become a critical research area in artificial intelligence and financial analytics, using machine learning and deep learning models to improve forecasting accuracy and decision-making. Recent literature highlights interest in hybrid and advanced neural network models for capturing temporal and nonlinear relationships in stock prices.

Dhanalakshmi et al. (2023) propose a Random Forest Algorithm that overcomes overfitting and under fitting in stock market prediction, using historical data to train the model and achieve high accuracy (95%-98%) for companies like Goldman Sachs, Apple, and Facebook. They conclude that machine learning can effectively predict stock market trends. Durga et al. (2023) further support this by achieving similar high accuracy with other models like XGBoost, highlighting the potential of machine learning in stock market forecasting.

Overall, Kirubakaran et al's (2023) method offers a promising approach for accurate stock prediction using LSTM models and SGD optimization. While further research is needed, this work paves the way for more reliable and customer-centric stock market forecasting.

Chong et al. (2017) focus on deep learning's ability to extract features from large datasets for high-frequency trading, while Sonkavde et al. (2023) propose a generic framework using various algorithms like Random Forest and LSTM for forecasting stock prices. Both highlight the potential of machine learning in this domain and kept the topic open for further research to address challenges and optimize performance.

Samuel Joseph et al. (2023) took efforts for predicting stock prices in Tanzania using machine learning techniques, specifically Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU), using Dar es Salaam Stock Exchange active stocks. The researchers developed a joint model to predict next-day closing prices, accounting for the number of outstanding shares of each stock. The study found that accounting for the number of outstanding shares significantly improved prediction accuracy.

Data Source and pre-processing

Data from diverse sources such as Kaggle, Yahoo Finance, and True Data are harnessed as inputs for the model. The implementation leverages various tools, recent technologies, and Python libraries. The process began with data preprocessing, where raw data was cleaned and standardized, and features were engineered to ensure model compatibility[Hamed Ghorban Tanhaei et al] (2024).

It's important to acknowledge the inherent uncertainty in stock market prediction; no method can ensure absolute accuracy. The primary objective is to cultivate a deeper understanding of market dynamics and trends, providing insights to guide investment decisions and strategies.

To predict stock market trends, the focus is on HDFC stock, utilizing data spanning from 2018 to 2023. The initial five rows and the last five rows of the employed dataset are presented for reference in Table-I below.

Table I: Past 5 years' data with Open, Close, High and Low values

	Date	Open	High	Low	Close	Adj Close	Volume
0	2018-08-06	1060.500000	1068.699951	1053.050049	1057.150024	1020.075012	2872316
1	2018-08-07	1062.500000	1067.250000	1054.000000	1065.449951	1028.083984	3714228
2	2018-08-08	1067.925049	1075.900024	1066.525024	1068.175049	1030.713501	4972690
3	2018-08-09	1071.000000	1071.000000	1047.324951	1059.250000	1022.101257	4472204
4	2018-08-10	1061.750000	1061.900024	1049.000000	1057.224976	1020.147400	2721470

Statistical parameters of the data required for further analysis are defined as shown below Table-II.

Table II: Past 5 years' data with Statistical parameters

```
df.describe()
```

	Open	High	Low	Close	Adj Close	Volume
count	1234.000000	1234.000000	1234.000000	1234.000000	1234.000000	1.234000e+03
mean	1331.466592	1344.087012	1317.500671	1330.859422	1301.476601	9.446193e+06
std	228.429412	229.155515	228.577056	228.866402	232.507996	6.401342e+06
min	770.450012	810.000000	738.750000	767.700012	747.039307	0.000000e+00
25%	1123.612488	1133.181213	1110.737518	1123.149963	1089.941955	5.246570e+06
50%	1372.975036	1388.000000	1357.625000	1372.200012	1341.923340	7.371894e+06
75%	1520.225037	1533.262543	1507.000000	1520.525024	1488.617188	1.123594e+07
max	1723.449951	1757.500000	1713.800049	1728.199951	1728.199951	5.406435e+07

The variation of the HDFC stock price in the last 1200 days is shown graphically in the figure below.

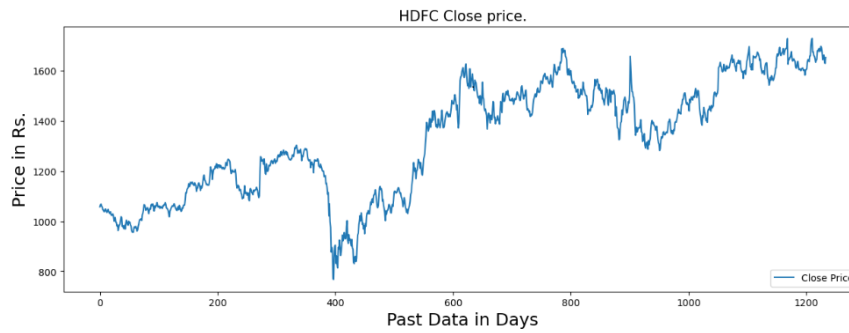


Fig. 1: The variation of the HDFC stock price, in last 1200 days

II. PROPOSED METHODOLOGY

Analyzing stock market data and making predictions involves a combination of analytical approaches and the utilization of statistical tools. Five different algorithms are proposed to predict stock market prices.

An attempt has been made, using five different techniques, to predict the stock price.

- Long Short-Term Memory (LSTM)
- Logistic Regression (LGR)
- Support Vector Machine/Classifier (SVM/SVC)
- eXtreme Gradient Boosting (XGBoost)
- Artificial Neural Network (ANN)

a) Long Short-Term Memory (LSTM)

The LSTM model is chosen for its ability to capture temporal dependencies and handle time-series data effectively [Althelaya KA et al] (2018). It is a subtype of recurrent neural networks (RNN), which excels in handling sequential data like time series, speech, and text, thanks to its ability to capture long-term dependencies. This model was able to predict the closing stock prices of major technology companies with a satisfactory level of accuracy [Li, Zhenglin et al] (2023). LSTM is a variant of recurrent neural network that can handle long-term dependencies and solve vanishing gradient problems [Karime Chahuán-Jiménez] (2024). In the context of stock market prediction, evaluating predictive models often involves the Root Mean Square Error (RMSE). With a target value close to zero for sound statistical analysis, the proposed LSTM model yields an RMSE of 76.32, prompting a halt in further stock price prediction procedures. The importance of RMSE lies in its role as a widely accepted metric for assessing predictive model accuracy, aiding traders, investors, and data scientists in making informed decisions and managing risk. Lower RMSE values indicate better model performance, where n is the number of observations, y_i is the actual value of the i^{th} observation, and \hat{y}_i is the predicted value of the i^{th} observation [Li, Zhenglin et al] (2023).

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

Figure 2 depicts the disparity between actual and predicted prices for HDFC stock.

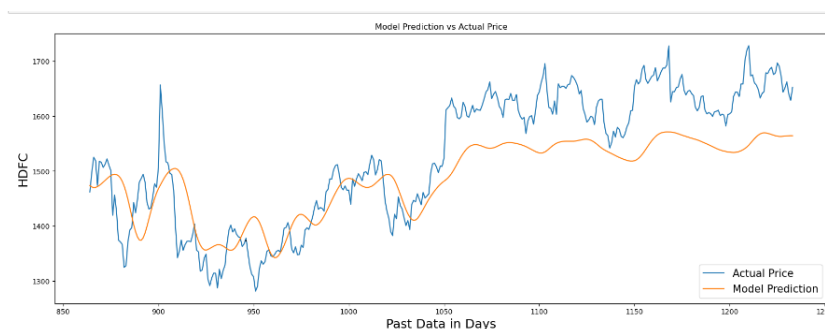


Fig. 2: Actual and predicted price variation of HDFC stock

b) Logistic Regression (LGR)

Although many algorithms are available for prediction of stocks, selecting the most accurate one continues to be the fundamental task for getting the best results. Logistic Regression analysis. involves training and execution of algorithms and getting the desired outputs [Shreenidhi Sriram et al] (2020). Given the significant

disparity between actual and predicted values using the LSTM algorithm, the application of Logistic Regression (LGR) is explored. LGR, a statistical method primarily designed for binary classification, proves invaluable in predicting one of two possible outcomes based on predictor variables—an essential tool in machine learning and statistics. However, upon training the LGR model for stock price prediction, it becomes evident that the training accuracy is 51.22%, and the validation accuracy is 47.13%, as shown in Figure 3 below. Recognizing the limitations of this model and its potential inability to capture complex relationships in data influenced by multiple factors, the decision is made to explore training the Support Vector Classification (SVC) model, and further processing with the LGR model is discontinued.

```
models =LogisticRegression()

models.fit(X_train, Y_train)
print(f'{models} : ')
print('Training Accuracy : ', metrics.roc_auc_score(Y_train, models.predict_proba(X_train)[:,:1])*100)
print('Validation Accuracy : ', metrics.roc_auc_score(Y_valid, models.predict_proba(X_valid)[:,:1])*100)
print()

LogisticRegression() :
Training Accuracy : 51.22178740444121
Validation Accuracy : 47.13541666666667
```

Fig. 3: Calculation of training and validation accuracy

Logistic regression is a simple and interpretable model, making it easy to understand the factors influencing the binary outcome. It can serve as a baseline model for classification tasks. Typically, it is used for classification tasks, where the outcome is binary (e.g., up or down) rather than predicting the exact stock price. Hence, it is useful for binary classification problems related to stock market prediction, such as predicting whether the stock price will go up or down in the next time period based on a set of features.

c) Support Vector Machine/Classifier (SVM/SVC)

The Support Vector Machine is a supervised machine learning algorithm specifically designed for binary and multiclass classification tasks. It overcomes the problem that traditional neural networks which can only find local optimal solutions and discover the global minimum of the objective function using known efficient algorithms and also have better generalization capabilities [Yang et al] (2023). Utilizing a high-dimensional space and optimal hyperplane, SVCs are effective for well-separated decision boundaries in complex datasets. Below Figure 4 shows training and validation accuracy obtained using this model. Despite achieving a training accuracy of 50.24% and a validation accuracy of 56.38% in predicting stock prices, SVC's limitations in capturing intricate data relationships prompt a transition to the XGBoost model for further analysis.

```
models =SVC(kernel='poly', probability=True)

models.fit(X_train, Y_train)
print(f'{models} : ')
print('Training Accuracy : ', metrics.roc_auc_score(Y_train, models.predict_proba(X_train)[:,:1])*100)
print('Validation Accuracy : ', metrics.roc_auc_score(Y_valid, models.predict_proba(X_valid)[:,:1])*100)
print()

SVC(kernel='poly', probability=True) :
Training Accuracy : 50.247022726090805
Validation Accuracy : 56.38020833333333
```

Fig. 4: Calculation of training and validation accuracy

d) XGBoost regression algorithm

XGBoost, an ensemble learning algorithm widely acclaimed for regression and binary classification tasks, stands out for its efficiency in handling large datasets and missing data. It uses a presentation of the second-order Taylor formula and adds a regular term, which is a control of the tree complexity and prevents overfitting[Zhang et al] (2022). Benefitting from a strong user community and support in popular languages like Python and R, XGBoost offers tunable hyperparameters for enhanced performance. A dedicated Python-based approach was employed for pre-processing and data preparation across all models. Training the XGBoost model resulted in impressive accuracy scores, with training and validation accuracies of 98.23% and 41.78%, respectively, as illustrated in Figure 5.

```

models = XGBClassifier()

models.fit(X_train, Y_train)
print(f'{models} : ')
print('Training Accuracy : ', metrics.roc_auc_score(Y_train, models.predict_proba(X_train)[: ,1])*100)
print('Validation Accuracy : ', metrics.roc_auc_score(Y_valid, models.predict_proba(X_valid)[: ,1])*100)
print()

```

```

XGBClassifier(base_score=None, booster=None, callbacks=None,
              colsample_bylevel=None, colsample_bynode=None,
              colsample_bytrees=None, early_stopping_rounds=None,
              enable_categorical=False, eval_metric=None, feature_types=None,
              gamma=None, gpu_id=None, grow_policy=None, importance_type=None,
              interaction_constraints=None, learning_rate=None, max_bin=None,
              max_cat_threshold=None, max_cat_to_onehot=None,
              max_delta_step=None, max_depth=None, max_leaves=None,
              min_child_weight=None, missing=nan, monotone_constraints=None,
              n_estimators=100, n_jobs=None, num_parallel_tree=None,
              predictor=None, random_state=None, ...) :
Training Accuracy : 98.29326798065424
Validation Accuracy : 41.783854166666664

```

Fig. 5: Calculation of training and validation accuracy

As this algorithm is typically used for classification tasks, where the outcome is binary (e.g., up or down) rather than predicting the exact stock price. Hence, it is useful for binary classification problems related to stock market prediction, such as predicting whether the stock price will go up or down in the next time period based on a set of features.

e) Artificial Neural Network based algorithm

In order to assess the performance of the developed stock market prediction models, a number of evaluation criteria will be used to evaluate these models. These criteria are applied to measure how close the real values are to the values predicted by using the developed models. They include Mean Absolute Error, Root Mean Square Error and correlation coefficient R[Sheta Alaa] (2015). Considering the limitations of LSTM, LGR, SVC and XGBoost, algorithms, Artificial Neural Networks (ANNs) is used for stock price prediction as an ANN generally consists of three layers, the input, hidden, and output layers, each consisting of numerous interconnected neurons[Song et al] (2023). They are capable of modeling complex patterns and relationships in historical stock market data. Initially trainable parameters are obtained as shown in below Figure 6.

```

Model: "sequential_20"
-----
Layer (type)                Output Shape         Param #
-----
dense_60 (Dense)            (None, 32)           64
dense_61 (Dense)            (None, 10)           330
dense_62 (Dense)            (None, 1)            11
-----
Total params: 405
Trainable params: 405
Non-trainable params: 0

```

Fig. 6: Calculation of trainable parameters

RMSE is a useful metric for evaluating the performance of Artificial Neural Network (ANN) models, particularly when the ANN is used for regression tasks, such as stock price prediction. RMSE is a metric used to measure the average magnitude of errors between predicted values and actual target values in a regression task. Lower RMSE values indicate better accuracy, as they suggest that the model's predictions are closer to the true values. When an ANN model is for training dataset, it makes predictions based on the input data and calculate the MSE loss.

RMSE Calculation:

After training, use the validation dataset (or test dataset if you prefer) to calculate the RMSE. To do this, first, predictions are obtained using the trained model on the validation dataset. Then the squared differences between the predicted values and the actual target values calculated as shown in Figure 7 below.

In the last step, the mean of these squared differences (MSE), the square root is calculated and RMSE value is obtained. All these steps are done in Python.

```

trainScore = classifier.evaluate(X_train, Y_train, verbose=0)
print('Train Score: %.2f MSE (%.2f RMSE)' % (trainScore, math.sqrt(trainScore)))
testScore = classifier.evaluate(X_test, Y_test, verbose=0)
print('Test Score: %.2f MSE (%.2f RMSE)' % (testScore, math.sqrt(testScore)))

```

Train Score: 0.03 MSE (0.18 RMSE)
 Test Score: 0.03 MSE (0.19 RMSE)

Fig. 7: Calculation of RMSE value

The train and test score as well as RMSE value are close to zero and which is a good indication of proper training of the model. The training model accuracy is obtained as 98.83%, shown in Figure 8 below, which indicates that the model is highly suitable for the prediction of the stock price.

	Close	Predictions	Percentage Accuracy
0	14.27	14.190000	0.994394
1	14.62	14.290000	0.977428
2	14.66	14.490000	0.988404
3	14.86	14.700000	0.989233
4	14.85	14.690000	0.989226
...
242	16.51	16.260000	0.984858
243	16.62	16.320000	0.981949
244	16.40	16.200001	0.987805
245	16.29	16.180000	0.993247
246	16.52	16.180000	0.979419

247 rows × 3 columns

```

Accuracy = data1['Percentage Accuracy'].mean()
print("Percentage Accuracy:", Accuracy*100)

```

Percentage Accuracy: 98.82937365166029

Fig. 8: Calculation of Model training accuracy

Comparing the results obtained from the above five algorithms, it is clear that ANN based regression algorithm gives highest model training accuracy, hence, this algorithm is used for the prediction of stock prices.

III. RESULTS AND CONCLUSION

Indeed, the stock market is a complex and dynamic system, posing challenges to accurate predictions. The proposed approach involves using Artificial Neural Networks (ANN) and Machine Learning (ML) algorithms, particularly those neural networks capable of capturing temporal dependencies in time series data. This methodology aims to enhance the prediction of stock prices by leveraging the capabilities of ANN and ML in handling the intricate patterns and dynamics inherent in financial time series. Results obtained using 4 different models is shown below Table-III.

Table-III: Comparison of results using various models

Sr. No.	Model used	Training Accuracy in %	Validation Accuracy in %
1	Logistic Regression (LGR)	51.22	47.13
2	Support Vector Machine/Classifier (SVM/SVC)	50.24	56.38
3	XGBoost regression algorithm	98.23	41.78
4	Artificial Neural Network based algorithm (Proposed Algorithm)	98.83	76.43

The developed prediction model underwent testing on an independent dataset, utilizing RMSE as a metric to analyze its performance and prediction capabilities. The lower RMSE values obtained indicate better accuracy in predicting stock prices. This improved accuracy can empower customers to make informed investment decisions by predicting opportune moments and suitable stocks in the dynamic stock market. Customer satisfaction with the outcomes of employing ANN models for stock market prediction is influenced by several factors. The accuracy of predictions, the reliability of the technology, and the alignment of

predictions with customers' expectations and goals all play pivotal roles in determining satisfaction levels. A well-performing ANN model that meets or exceeds customer expectations in terms of accuracy and reliability is more likely to result in positive and satisfactory user experience in the context of stock market predictions. It can be seen that the proposed Artificial Neural Network based algorithm gives highest accuracy of 98.83%.

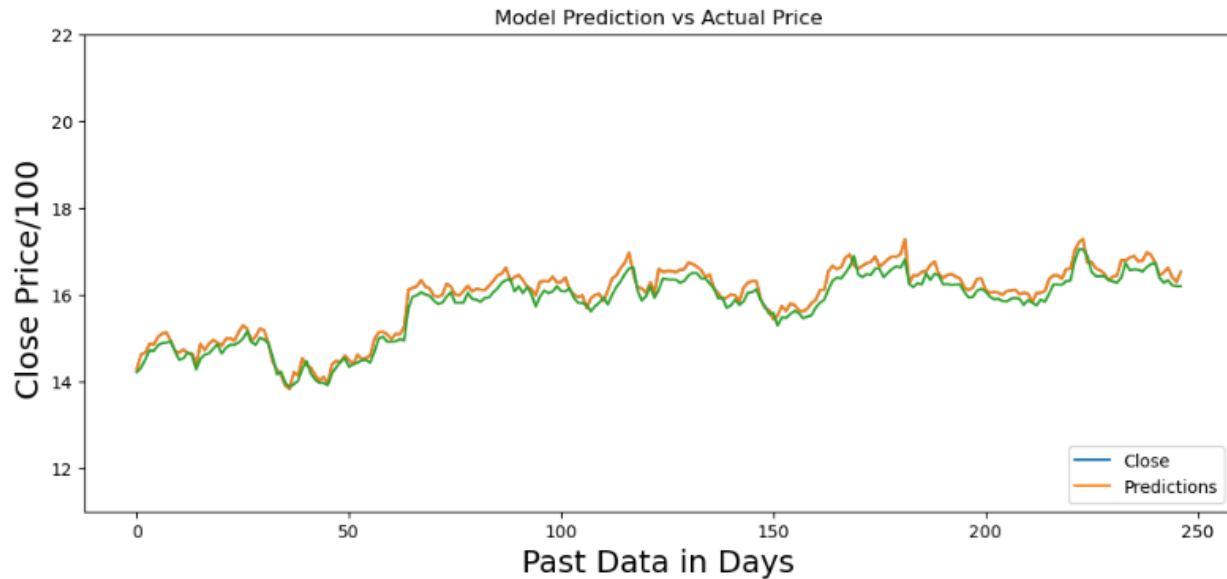


Fig. 9: Superimposed graph of the predicted and the original prices

Figure 9 illustrates the superimposed graph of predicted and targeted values for the Nifty index in the stock market. The graph provides a visual comparison between actual stock prices and those predicted by the model. It's important to note that the model's performance is considered quite good, given the constraints of only considering limited features. The stock market is influenced by a multitude of factors, some of which may be challenging to quantify. Despite these challenges, the model's ability to generate accurate predictions within the specified limitations is noteworthy. The root mean square error (RMSE) calculated between the actual (target) value of the closing price of the Nifty index of the Indian Stock Market.

REFERENCES

- [1] Naeini MP, Taremian H, Hashemi HB, "Stock market value prediction using neural networks", IEEE International Conference on Computer Information Systems and Industrial Management Applications (CISIM), Poland, 2010.
- [2] Payal Soni, Yogya Tewari and Deepa Krishnan, "Machine Learning Approaches in Stock Price Prediction: A Systematic Review", Journal of Physics: Conference Series 2161, 2022.
- [3] R. Dhanalakshmi, Dr Saleem Basha, "A Logical Investigation of Stock Market Prediction and Analysis using Supervised Machine Learning Algorithm", International Conference on Networking and Communications (ICNWC), 2023
- [4] Durga P, Sudhakar T, "Machine Learning Classification Methods for Stock Market Forecasting", International Conference on Innovative Data Communication Technologies and Application (ICIDCA-2023), IEEE Xplore Part Number: CFP23CR5-ART; ISBN: 979-8-3503-9720-8, 2023.
- [5] Stewart Kirubakaran S, Jason Selvadurai J, Ashok V, Antony Nivin J, Dennies Aron A, Bijolin E, "A Brief Assessment on Stock Market Forecast using Long Short Term Memory (LSTM) Algorithm", Proceedings of the 7th International Conference on Trends in Electronics and Informatics (ICOEI 2023), IEEE Xplore Part Number: CFP23J32-ART; ISBN: 979-8-3503-9728-4, 2023.
- [6] Chong, E., Han, C., & Park, F. C, "Deep learning networks for stock market analysis and prediction: Methodology, data representations, and case studies", Elsevier- Expert Systems with Applications, ISSN 0957-4174, Vol 83, pp187-205, 2017.
- [7] Y.S. Abu-Mustfa and A.F. Atya, "Intro. to financial forecasting", Applied Intelligence, 6, (1996), pp. 205-213.
- [8] Hamed Ghorban Tanhaei, Payam Boozary, Sogand Sheykhan, Maryam Rabiee, Farzam Rahmani, "Iman Hosseini, Predictive analytics in customer behavior: Anticipating trends and preferences", Science Direct-Results in Control and Optimization, Vol 17, 2024.

- [9] Althelaya KA, El-Alfy E-SM, Mohammed S, “Evaluation of Bidirectional LSTM for Short-and Long-Term Stock Market Prediction”, Proceedings of the 9th International Conference on Information and Communication Systems (ICICS 2018), Irbid, Jordan, April 2018.
- [10] Li, Zhenglin & Yu, Hanyi & Xu, Jinxin & Liu, Jihang & Mo, Yuhong, “Stock Market Analysis and Prediction Using LSTM: A Case Study on Technology Stocks”, Innovations in Applied Engineering and Technology. 1-6, 2023.
- [11] Shreenidhi Sriram, Sanjana Rangarajan, “Stock Market Prediction using Logistic Regression Analysis - A Pilot Study”, International Journal for Research in Applied Science & Engineering Technology (IJRASET), ISSN: 2321-9653, Vol 8, Iss VII, July 2020.
- [12] Yang, Jingdong, “Support Vector Machine-based Stock Prediction Analysis”, Highlights in Business, Economics and Management, Vol. 3, pp 12-18, 2023.
- [13] Zhang, Yifan, “Stock Price Prediction Method Based on XGboost Algorithm”, Proceedings of the 2022 International Conference on Bigdata Blockchain and Economy Management (ICBBEM 2022), pp 595-603, 2022.
- [14] Sheta Alaa, Ahmed Sara and Faris Hossam, “A Comparison between Regression, Artificial Neural Networks and Support Vector Machines for Predicting Stock Market Index”, International Journal of Advanced Research in Artificial Intelligence, Vol 4, pp 55-63, 2015.
- [15] Karime Chahuán-Jiménez, “Neural Network-Based Predictive Models for Stock Market Index Forecasting”, Journal of Risk and Financial Management, Vol 17, pp 2-18, 2024.
- [16] Hyunsun Song and Hyunjun Choi, “Forecasting Stock Market Indices Using the Recurrent Neural Network Based Hybrid Models: CNN-LSTM, GRU-CNN, and Ensemble Models”, Appl. Sci. 2023, 13, 4644, 2023.
- [17] Zahid Iqbal, R. Ilyas, W. Shahzad, Z. Mahmood and J. Anjum, “Efficient Machine Learning Techniques for Stock Market Prediction”, Int. Journal of Engineering Research and Applications, Vol. 3, Iss 6, pp.855-867, 2013.