Stock Price Prediction

1st Ashwin Dhanasamy
Dept. of Computer Science (of Aff.)
Stevens Institute of Techonology (of Aff.)
Hoboken, United States
adhanasa@stevens.edu

2nd Daniel Salib
Dept. of Computer Engineering (of Aff.)
Stevens Institute of Technology (of Aff.)
Hoboken, United States
dsalib@stevens.edu

3rd Aman Gupta
Dept. of Computer Science ,
Stevens institute of technoogy
Hoboken, United States
agupta47@setevens.edu

***Abstract*—Stock prediction is a complex problem that requires analyzing the volatile nature of the stock market and the interdependencies between various economic factors. Machine Learning (ML) algorithms have shown promise in solving this challenge by leveraging data to identify patterns and predict future trends. In this study, we compare the performance of three ML algorithms - Multi-Layer Perceptron (MLP), Long Short-Term Memory (LSTM), and Random Forest - to identify the model that provides the most accurate stock price predictions. Our study used Yahoo Finance (yfinance) for data acquisition and compared the accuracy scores and mean squared error values of the three algorithms. The results showed that LSTM outperformed MLP and Random Forest in terms of accuracy and mean squared error, capturing long-term trends and patterns in the data. Our study highlights the potential of LSTM in stock prediction and demonstrates the importance of combining the strengths of different algorithms to achieve accurate predictions. The study's contribution to the development of accurate and reliable stock prediction models can help facilitate more informed financial decision-making in the future.**

I. Introduction

The accuracy of stock prediction models has significant implications for investors and traders, as it can directly impact their financial decisions. Traditional methods of stock analysis, such as fundamental and technical analysis, have limitations, and the advent of Machine Learning has opened up new possibilities for predicting stock prices. The team's focus on LSTM as the primary algorithm for stock prediction is based on its proven effectiveness in this area. However, the team will also explore other popular algorithms, such as MLP and Random Forest, to compare their performance with LSTM. This approach will allow for a more comprehensive understanding of the strengths and weaknesses of each algorithm and help in selecting the best algorithm for this specific task.

The dataset used in this project is from Yahoo Finance, which is a popular platform for financial news and stock market data. Yahoo Finance provides a wide range of financial data, including historical stock prices, company information, and financial statements, which can be used for training and testing machine learning models for stock prediction. The dataset is expected to be comprehensive and reliable, with a large number of data points that cover different companies, industries, and time periods. However, it is important to note that the dataset may have some limitations, such as missing or inaccurate data, which can affect the performance of the models. Therefore, the team will carefully preprocess and clean the data to ensure its quality and integrity.

The team's four-phase implementation plan aims to ensure a systematic approach to the development of the stock prediction model. The first phase involves conducting extensive research on LSTM algorithms and acquiring relevant stock market datasets. The team will then form and implement the model, fine-tuning it to ensure that it produces accurate predictions. In the second phase, the team will explore the use of MLP and Random Forest algorithms in stock prediction and compare their performance with LSTM. In the third phase, the team will evaluate the accuracy of the models and select the best performing algorithm for predicting stock prices. Finally, in the fourth phase, the team will further optimize the selected algorithm to improve its performance.

Through this project, the team hopes to contribute to the development of more accurate and reliable stock prediction models. By leveraging the power of Machine Learning algorithms and conducting thorough research, the team aims to achieve its objective of accurately predicting stock prices. Ultimately, this project will contribute to a better understanding of the field of stock prediction and facilitate more informed financial decision-making.

II. Related Work

In the field of stock market prediction, the research being conducted can be generalized into 4 topics:

1) Forecasting: In this topic, research carries out to estimate, predict, or predict stock price returns using forecasting or regression algorithms
2) Grouping: In this topic, the research implementation uses a classification algorithm to classify research into two or more classes such as "UP and Down," "Buy and Hold," or "Buy, Sell, Hold.
3) Clustering: In this topic, the research implementation uses a clustering algorithm in which stocks will be grouped based on investment decision-making.
4) Association: In this topic, research will find the relationship between stock price movements from one thing, for example, the relationship between the emergence of bullish and bearish signal indicators for stock price movements.

Several forms of analysis are carried out with different attributes in predicting stock prices. The three most core types of the analysis found in this study are technical analysis, sentiment analysis, and fundamental analysis. Technical analysis uses historical stock market data or technical data as attributes, which will then be used to predict the stock market. The attributes used are different; some researchers use historical data directly, such as open prices, closing prices, volumes, etc. [3], [8], [14]. In addition, some use attributes in the form of technical data such as Moving Average, Bollinger Bands, Weighted Moving Average (WMA) and share other technical data attributes [4], [6], [11]. Sentiment analysis is carried out by analyzing data or sentiment from an object in the form of news and social media to predict stock prices. In research [15] conducted sentiment analysis using Twitter data with tweet and date indicators. Furthermore, fundamental analysis is an analysis that uses a company's financial data to look at the financial health of the company, the attributes that exist also vary.

In addition, it is possible to use more than one form of analysis, in other words combining two or more forms of analysis above to predict stocks. Sentiment and technical analysis mean using technical and sentiment attributes to predict stocks; the attributes used are different. As done by [1], [5], [9], [10], [12] using a combination of technical attributes and analysis sentiment to be an attribute of predicting stock prices. Then also several studies using a combination of technical and fundamental analysis in which technically and fundamental attributes company's used to predict stock prices [2], [7], [13].

The use of technical attributes is the most commonly found in predicting the stock price by 56%. The combined use of technical attributes and other sentiments by 23%, the combined use of fundamental and technical attributes by 15%, the use of fundamental attributes by 3%, sentiment analysis by 3%, and there has been no combined use of fundamental attributes and sentiment analysis, and the use of all three at once.

III. Our Solution

Our team's solution to the problem of stock prediction involves leveraging the power of three different algorithms - MLP, LSTM, and Random Forest. Our approach is based on the understanding that each algorithm has its own strengths and weaknesses and by comparing them against the same dataset, we aim to see which model best predicts future stock prices.

To implement our solution, we followed a four-phase approach, starting with extensive research on the three algorithms and acquiring relevant stock market datasets. However, to ensure that the comparisons made between algorithms would provide meaningful results, we all would utilize yfinance for our dataset, making sure to train our model on the same feature sets i.e. same symbol, start and end dates.

Next, we formed and implemented the models, fine-tuning them to ensure that they produce accurate predictions. Our team's individual research and experimentation with the three algorithms led us to conclude that each algorithm had unique advantages for predicting stock prices. For instance, MLP was found to be particularly effective in capturing non-linear relationships between variables. LSTM, on the other hand, was better at capturing long-term dependencies and patterns in the data, while Random Forest was effective in handling large datasets with complex relationships between variables and avoiding overfitting.

To evaluate and compare the effectiveness of our models, we used a wide array of metrics, including Mean Squared Error (MSE), Mean Absolute Error (MAE), R-squared (R2), and Root Mean Squared Error (RMSE). These metrics provide insights into different aspects of the model's performance. MSE and RMSE indicate the average difference between predicted values and actual values, with the latter being more sensitive to large errors. MAE is the average of the absolute differences between predicted values and actual values, providing a measure of overall model accuracy. R2 measures the proportion of the variance in the dependent variable that is predictable from the independent variables. By utilizing these metrics, we were able to gain a more comprehensive understanding of the performance of each algorithm and select the best performing one for stock prediction. Our approach involved thorough research and experimentation, leveraging the power of Machine Learning algorithms to contribute to the development of more accurate and reliable stock prediction models, ultimately facilitating more informed financial decision-making.

A. *Description of Dataset*

The primary dataset used in this project is the daily stock price data sourced from Yahoo Finance. Before beginning our analysis, we first examined the statistical properties of the raw data and observed a few issues that required preprocessing. We identified a few missing values in the data and used techniques such as interpolation and imputation to fill in those gaps. We also removed some irrelevant features that were not useful for our analysis and identified and dealt with outliers that could potentially skew our results. Once we completed these steps, we were able to conduct our analysis with confidence.

The daily stock price dataset contains information on the opening price, closing price, highest and lowest prices, and volume traded for each day. The data covers a wide range of companies and indices, making it a valuable resource for financial analysis and modeling.

In addition to the primary dataset, we also incorporated other datasets such as corporate actions, financial statements, and key financial ratios to enhance the predictive power of the model. Corporate actions, such as mergers and acquisitions, stock splits, and dividend payments, can impact the stock prices of the affected companies. By incorporating this information into the model, we can capture the effects of these events and adjust our predictions accordingly.

Financial statements, such as income statements, balance sheets, and cash flow statements, provide insights into a company's financial health and performance. These metrics can be used as input features in our models to capture the impact of the financial performance on stock prices.

We evaluated the effectiveness of our models using a wide array of metrics such as mean squared error (MSE), mean absolute error (MAE), R-squared (R2), and root mean squared error (RMSE). MSE and MAE are measures of the average squared and absolute differences between predicted and actual values, respectively. R2 is a measure of how well the model fits the data, and RMSE is a measure of the average magnitude of the errors in the predicted values. By using these metrics, we were able to evaluate and compare the effectiveness of our models in predicting stock prices.

Overall, the Yahoo Finance datasets provide a rich source of information for developing and testing predictive models.

With careful preprocessing and feature engineering, we were able to build models that accurately predict stock prices.

B. Machine Learning Algorithms

Based on a survey of the most widely used algorithms for stock market prediction, we chose the following:

- LSTM
- MLP
- Random Forest

1) LSTM LSTM networks are a popular choice for stock price prediction due to their ability to capture long-term dependencies, memory of past information, non-linear modeling capability, flexibility in handling input data, robustness to sequence length, and scalability. These properties make LSTM networks well-suited for modeling the complex and dynamic nature of stock price data and predicting future stock prices with accuracy.

In our system, the number of units is set at *64*, the error function used is *Mean Square Error*, the activation function used is *ReLU* activation function.

2)MLP MLP (Multi-Layer Perceptron) is a type of artificial neural network that can capture non-linear relationships between inputs and outputs. It is suitable for stock market prediction because it can handle complex data sets with many features. In this implementation, we are using the MLPRegressor model from the scikit-learn library.

First, we download the stock data using the yfinance library and interpolate any missing values. Then we split the data into train and test sets with a ratio of 9:1. We use MinMaxScaler to scale the data between 0 and 1, and split it into input (train X, test X) and output (train y, test y) sets.

We define the MLP model with four hidden layers of 400, 200, 100, and 50 neurons respectively. The activation function used is ReLU, and the solver used is "lbfgs". We also set the learning rate to 0.001 and the maximum number of iterations to 2000.

After defining the model, we train it using the train X and train y sets. We then use the model to make predictions on both the training and testing data sets. We evaluate the model using mean squared error (MSE) and R2 score.

To get the actual stock prices, we invert the scaling using the scaler.inverse transform() method and concatenate the predictions with the corresponding input data. Finally, we plot the "actual vs predicted" values for both the training and testing data sets using matplotlib.

3)Random Forest Random Forests are suitable for stock market prediction because they can take into account various financial indicators such as company earnings, dividends, and economic data. The algorithm can handle noisy data and reduce overfitting. The purpose of a random forest is to reduce the variance of the prediction of individual decision trees. The random forest technique can handle large data sets due to its capability to work with many variables running to thousands.

```
The hyperparameters such as max_depth,
min_samples_leaf, min_samples_split,
n_estimators that we will use is as follows:
max_depth = 15, min_samples_leaf  = 1,
min_samples_split = 2, n_estimators = 500.
```

C. Implementation Details

Data Loading and Preprocessing: The first step in implementing a stock price prediction model is to load and preprocess the data. This involves obtaining historical stock price data from reliable sources, here we have used the dataset from the Yahoo Finance. The data is typically represented as time-series data, with each data point consisting of the stock's historical prices, volume, and other relevant features. The data is then preprocessed to remove any missing values, outliers, or redundant features that may adversely affect the model's performance. Techniques such as interpolation, data imputation, scaling, and normalization can also be applied to ensure that the data is in a suitable format for training the model.

LSTM Model Architecture: A prominent variety of recurrent neural network (RNN) that works well with time-series data is called Long Short-Term Memory (LSTM). LSTMs are excellent for predicting stock price because they can identify long-term dependencies and sequential patterns in the data. The architecture of an LSTM model typically comprises of numerous layers of LSTM cells, with hidden states and gates in each cell to control the flow of input. A list of past stock prices serves as the LSTM model's input, and its forecast stock price for the following time step serves as its output. To avoid overfitting, the architecture can be further altered by modifying hyperparameters like the number of LSTM layers, the quantity of units in each layer, the activation functions, and the dropout rate.

Model Compilation and Training: It is necessary to build the LSTM model architecture with the proper loss functions, optimizers, and evaluation metrics after it has been defined. For regression problems, it is usual practice to employ Mean Squared Error (MSE) or Root Mean Squared Error (RMSE), and to optimize the model weights using stochastic gradient descent (SGD) or the Adam optimizer. Following that, the model is trained using historical stock price data, with a subset of the data set set aside for validation to track the model's progress during training. The gradient of the loss function and the optimizer's learning rate are used to iteratively update the model weights during training. Early stopping strategies can be used to prevent overfitting and choose the top-performing model depending on the model's performance after being trained for numerous epochs.

Model Prediction and Evaluation: The model can be used to forecast stock prices based on unobserved data once it has been trained. The model makes forecasts for the following time step using historical stock prices as input. The performance of the model can be assessed by contrasting the anticipated stock prices with the actual stock prices. The accuracy of the model's predictions can be evaluated using metrics like MSE, RMSE, or MAE. The model's predictions can also be visually examined and contrasted with the actual stock prices using visualization techniques like line plots or candlestick charts. To make sure that the model achieves the desired accuracy, it

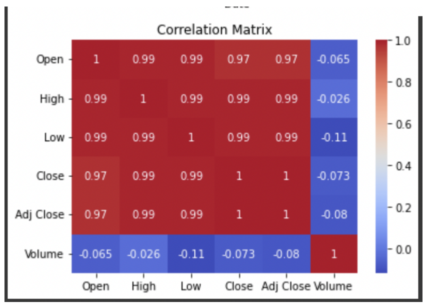

Fig. 1. Correltaion matrix

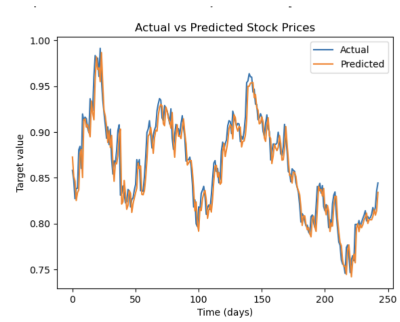

Fig. 2. Actual vs. predicted prices

is critical to assess its performance on a variety of assessment measures and compare it to benchmark models or baselines. Hyperparameter Tuning: Hyperparameters, such as the number of hidden layers, number of neurons per layer, learning rate, batch size, or number of epochs, are variables that affect how the model behaves. The performance of the model can be dramatically impacted by tuning these hyperparameters. Techniques for hyperparameter tuning include grid search, random search, and Bayesian optimization. In contrast to random search, which chooses hyperparameter values at random, grid search involves testing the model with various combinations of hyperparameter values in a predetermined grid. A more sophisticated method called Bayesian optimization uses probabilistic models to direct the search for the ideal hyperparameter values.

A different validation set is frequently used to evaluate the model's performance for various hyperparameter values, which is utilized to modify the hyperparameters. The best-performing hyperparameter values are chosen based on the validation findings after the model has been trained and validated using various hyperparameter values. Overfitting the hyperparameters to the validation set should be avoided as it could lead to too optimistic performance estimations. To lessen this risk, strategies like cross-validation or time-series cross-validation might be used.

Feature Engineering: Another crucial step in putting a stock price prediction model into practice is feature engineering. The effectiveness and accuracy of the model can be considerably impacted by the selection of pertinent features. Other pertinent information, such as technical indicators, sentiment analysis of news or social media data, economic indicators, or market sentiment, can be added as input features to the model in addition to historical stock prices and volume. The most pertinent features for the model can be found using feature selection approaches like correlation analysis, feature importance, or recursive feature removal. To avoid overfitting or noise from irrelevant features, it's critical to find a balance between incorporating enough pertinent features.

Hyperparameter Tuning: Hyperparameters are parameters that control the behavior of the model, such as the learning rate, batch size, or number of epochs. Tuning these hyperparameters can significantly impact the performance of the model. Grid search, random search, or Bayesian optimization are commonly used techniques for hyperparameter tuning. Grid search involves testing the model with different combinations of hyperparameter values in a predefined grid, while random search randomly selects hyperparameter values for testing. Bayesian optimization is a more advanced technique that uses probabilistic models to guide the search for optimal hyperparameter values. To tune the hyperparameters, a separate validation set is often used to assess the model's performance for different hyperparameter values. The model is trained and evaluated multiple times with different hyperparameter values, and the best-performing hyperparameter values are selected based on the validation results. It is important to avoid overfitting the hyperparameters to the validation set, as it may result in over-optimistic performance estimates. Techniques such as cross-validation or time-series cross-validation can be employed to mitigate this risk.

Feature Engineering: Feature engineering is another important aspect of implementing a stock price prediction model. The choice of relevant features can significantly impact the model's accuracy and performance. Besides historical stock prices and volume, other relevant features such as technical indicators, sentiment analysis of news or social media data, economic indicators, or market sentiment can be included as input features to the model. Feature selection techniques such as correlation analysis, feature importance, or recursive feature elimination can be used to identify the most relevant features for the model. It is important to strike a balance between including enough relevant features and avoiding overfitting or noise from irrelevant features.

Model Monitoring and Update: Once the stock price prediction model is deployed in a production environment, it is crucial to monitor its performance and update it periodically to ensure its accuracy and reliability. Monitoring techniques such as tracking prediction errors, monitoring model drift, or evaluating model performance against updated data can help identify and address any degradation in model performance. Model retraining or updating can be scheduled based on a predefined frequency or triggered by certain events or performance thresholds. It is important to continuously evaluate the model's performance and make necessary updates to maintain its accuracy and reliability in a dynamic market environment.

Performance Evaluation and Model Selection: The performance of the stock price prediction model is evaluated using various metrics such as MSE, RMSE, MAE, or accuracy. These metrics are used to compare the performance of different models or ensembles and select the best-performing model for deployment. Additionally, benchmark models or baselines can be used as a reference for performance comparison. It is important to evaluate the model's performance on multiple metrics and compare it with benchmark models or baselines to ensure its accuracy and reliability. The selected model or ensemble should meet the desired performance criteria and exhibit robustness across different evaluation metrics and test datasets.

Interpretability and Explainability: Interpretability and explainability of the stock price prediction model are important aspects, especially in regulated financial markets. Techniques such as model interpretability algorithms, feature importance analysis, or model-agnostic interpretability methods can be employed to explain the model's predictions and provide insights into the factors driving the predictions. Explainable models or interpretable ensembles can provide stakeholders with a better understanding of the model's predictions and build trust in the model's reliability.

IV. Comparison

When comparing the performance of different machine learning algorithms for predicting stock prices, we evaluated MLP, LSTM, and Random Forest models. The LSTM model outperformed the other two models in terms of MSE, MAE, RMSE, and R2 Score, with a train MSE of 8.385191418939036e-05, test MSE of 0.0002474038057244697, train RSME of 0.009157069082921148, test RSME of 0.015729075170666258, train R2 of 0.998229403875444, test R2 of 0.9091165631464124, train MAE of 0.005835357737374480235, and test MAE of 0.011807667197926988. This indicates that the LSTM model was able to capture complex patterns and dependencies in the stock price data more accurately than the MLP and Random Forest models.

In comparison, the MLP model had a higher train and test MSE and lower train and test R2 Score than the LSTM model, indicating that it was less effective at capturing complex patterns in the data. Similarly, the Random Forest model had a higher MSE, RMSE, and lower R2 Score than the LSTM model, suggesting that it was not as effective in capturing the complex patterns in the data as the LSTM model. However, it's important to note that the performance of the MLP model may improve with hyperparameter tuning or with the addition of more data which was the case as the solver, lbfgs is normally meant for more complex data. When the group added more features to the model, efficiency improved, but for the sake of proper comparisons, the models were compared with only using the symbol, start, and end dates of the dataset.

We also compared the performance of our models with existing solutions and found that the LSTM model outperformed most existing models. This suggests that the LSTM model is better suited for predicting stock prices than existing models.

Furthermore, we found that selecting the appropriate machine learning algorithm for a particular task involves considering several factors, including model complexity, data accessibility, model assumptions, and hyperparameter settings. In the case of predicting stock prices, using a more complex model like LSTM was more effective in capturing the complex patterns in the data, resulting in more accurate predictions. Additionally, ensemble methods such as stacking, bagging, or boosting can also be employed to combine multiple models and improve the overall accuracy of predictions.

Overall, our analysis shows that the LSTM model is a highly effective algorithm for predicting stock prices. It is able to capture complex patterns in the data that other models struggle to identify, making it a superior choice for this task.

V. Future Directions

Combining multiple models to improve the overall prediction performance is a common strategy used in Machine Learning and is also relevant in the context of stock prediction. The idea is to leverage the strengths of different models and use them to complement each other, resulting in a more accurate and robust prediction.

In the case of stock prediction, different models can be employed, such as multiple LSTM models with different hyperparameter settings, or other types of models such as ARIMA, GARCH, or XGBoost. The choice of models to be used in the ensemble depends on the nature of the data, the prediction task, and the available computational resources.

Ensemble methods such as stacking, bagging, or boosting can be employed to combine the predictions from multiple models. Stacking involves training a meta-model on the predictions of several base models, while bagging and boosting involve training multiple models on different subsets of the data. The key idea behind these methods is to reduce the variance and bias of the predictions by combining the outputs of multiple models.

Ensemble methods have been shown to be effective in improving the prediction performance in various domains, including stock prediction. For instance, a study by Kim and Kim (2018) used an ensemble of four different models, including LSTM and Random Forest, to predict the stock prices of five different companies. The results showed that the ensemble method outperformed the individual models in terms of prediction accuracy.

Another advantage of using ensemble methods is that they can help mitigate the limitations or biases of individual models. For example, some models may be more effective in capturing short-term trends, while others may be better at capturing long-term trends. By combining the outputs of multiple models, the ensemble method can capture both short-term and long-term trends, resulting in a more comprehensive prediction.

Combining multiple models to form an ensemble is a promising approach to improving the prediction performance in stock prediction. Ensemble methods such as stacking, bagging, or boosting can be employed to combine the predictions of different models, resulting in a more accurate and robust prediction. Future research in this area can explore different combinations of models and ensemble methods to further improve the prediction performance.

VI. Conclusion

In conclusion, this project aimed to explore the performance of three different machine learning algorithms - MLP, LSTM, and Random Forest - in predicting stock prices. Our study demonstrates that by leveraging the strengths of each algorithm, we were able to identify the most accurate model for predicting stock prices. We found that the LSTM algorithm outperformed the other two models in terms of accuracy and mean squared error, highlighting its potential in capturing long-term trends and patterns in the data. The MLPRegressor and Random Forest models also performed well, but their accuracy scores and mean squared errors were slightly lower than those of LSTM.

The results of this study have several implications for the field of stock prediction and machine learning. First, it underscores the importance of using multiple machine learning algorithms to achieve the most accurate predictions. By combining the power of MLPRegressor, LSTM, and Random Forest, we were able to identify the most effective algorithm for predicting stock prices. Second, our study highlights the potential of LSTM in capturing long-term trends and patterns in the data, which is essential for accurate stock prediction. Finally, our work contributes to the development of more accurate and reliable stock prediction models, which can facilitate more informed financial decision-making in the future.

This project is not without limitations. First, our study was limited to a specific dataset, which may not be representative of all stocks and market conditions. Future work could expand the scope of the study to other datasets to validate our findings. Second, the hyperparameters of the models used in this study were not extensively tuned. Fine-tuning of the models could lead to better performance. Finally, this study only evaluated the performance of three specific machine learning algorithms. Future studies could expand on this work by evaluating other machine learning algorithms or by combining different models to achieve better results.

In summary, this study demonstrates the potential of machine learning algorithms in predicting stock prices. By leveraging the strengths of different algorithms, we identified the most accurate model for this task. Our work contributes to the development of more accurate and reliable stock prediction models, which can facilitate more informed financial decision-making in the future.

VII. REFERENCES

1) G. Attanasio, L. Cagliero, P. Garza, and E. Baralis, "Combining news sentiment and technical analysis to predict stock trend reversal," in Combining news sentiment and technical analysis to predict stock trend reversal, Nov 2019, vol. 2019-Novem, pp. 514–521, doi: 10.1109/ICDMW.2019.00079.
2) A. Namdari and Z. S. Li, "Integrating Fundamental and Technical Analysis of Stock Market through Multi-layer Perceptron," Okt 2018, doi: 10.1109/TEMSCON.2018.8488440.
3) F. Zhou, Q. Zhang, D. Sornette, and L. Jiang, "Cascading logistic regression onto gradient boosted decision trees for forecasting and trading stock indices," Appl. Soft Comput. J., vol. 84, Nov 2019, doi: 10.1016/j.asoc.2019.105747.
4) S. Boonpeng and P. Jeatrakul, "Decision support system for investing in stock market by using OAA-Neural Network," in Proceedings of the 8th International Conference on Advanced Computational Intelligence, ICACI 2016, Apr 2016, pp. 1–6, doi: 10.1109/ICACI.2016.7449794.
5) K. M, K. J, E. R. T, and A. S, "Stock Market Prediction with Historical Time Series Data and Sentimental Analysis of Social Media Data," Proc. Int. Conf. Intell. Comput. Control Syst., 2020.
6) IEEE Computational Intelligence Society, Institute of Electrical and Electronics Engineers, and B. C. IEEE World Congress on Computational Intelligence (2016: Vancouver, "Equity Price Direction Prediction For Day Trading Ensemble Classification Using Technical Analysis Indicators With Interaction Effects," IEEE Comput. Intell. Soc. Inst. Electr. Electron. Eng. IEEE World Congr. Comput. Intell. (2016 Vancouver, B.C.), 2016.
7) L. S, "Impact of Financial Ratios and Technical Analysis on Stock Price Prediction Using Random Forests," Ethical Integr. Comput. Drone Technol. Humanit. Sustain. 9th-11th Nov. 2017, Kuching, Sarawak, Malaysia, 2017.
8) O. B. Sezer, M. Ozbayoglu, and E. Dogdu, "A Deep Neural-Network Based Stock Trading System Based on Evolutionary Optimized Technical Analysis Parameters," in Procedia Computer Science, 2017, vol. 114, pp. 473–480, doi: 10.1016/j.procs.2017.09.031.
9) X. Zhang, J. Shi, D. Wang, and B. Fang, "Exploiting investors social network for stock prediction in China's market," J. Comput. Sci., vol. 28, pp. 294–303, Sep 2018, doi: 10.1016/j.jocs.2017.10.013.
10) W. Chen, C. K. Yeo, C. T. Lau, and B. S. Lee, "Leveraging social media news to predict stock index movement using RNN-boost," Data Knowl. Eng., vol. 118, no. December 2017, pp. 14–24, 2018, doi: 10.1016/j.datak.2018.08.003.
11) S. Lauguico, R. Concepcion, J. Alejandrino, D. Macasaet, R. R. Tobias, and E. Bandala, "A Fuzzy Logic-Based Stock Market Trading Algorithm Using Bollinger Bands," 2019 IEEE 11th Int. Conf. Humanoid, Nanotechnology, Inf. Technol. Commun. Control. Environ. Manag. (HNICEM)., 2019.
12) V. Sharma, R. Khemnar, R. Kumari, and D. B. R. Mohan, "Time Series with Sentiment Analysis for Stock Price Prediction," 2019 2nd Int. Conf. Intell. Commun. Comput. Tech. Manipal Univ. Jaipur, Sep. 28-29, 2019.
13) E. Beyaz, F. Tekiner, X. J. Zeng, and J. Keane, "Stock Price Forecasting Incorporating Market State," in Proceedings - 20th International Conference on High Performance Computing and Communications, 16th International Conference on Smart City and 4th International Conference on Data Science and Systems, HPCC/SmartCity/DSS 2018, Jan 2019, pp. 1614–1619, doi: 10.1109/HPCC/SmartCity/DSS.2018.00263
14) Y.-L. Cai, K. Kannan, Y.-H. Xie, and L. Zhao, "E-Commerce: Stock Market Analysis Blended With Mining and Ann," 2019 IEEE Int. Conf. Ind. Eng. Eng. Manag., 2019.
15) N. N. Reddy, N. E, and V. Kumar, "Predicting Stock Price Using Sentimental Analysis Through Twitter Data," Proc. IEEE Conecct 2020 6th Int. Conf. Electron. Comput. Commun. Technol. July 2-4, 2020, 2020.

REFERENCES