Scientific Research Publishing

Journal of Computer and Communications, 2025, 13(12), 158-168
https://www.scirp.org/journal/jcc
ISSN Online: 2327-5227
ISSN Print: 2327-5219

TriFusion Stacking (TFS) for Stock Prediction: Integrating Sentiment, Technical, and Fundamental Analysis

Shawn Tang

Independent Researcher, Pennsylvania, USA
Email: shawnjtang@gmail.com

How to cite this paper: Tang, S. (2025) TriFusion Stacking (TFS) for Stock Prediction: Integrating Sentiment, Technical, and Fundamental Analysis. *Journal of Computer and Communications*, **13**, 158-168.
https://doi.org/10.4236/jcc.2025.1312009

Received: November 20, 2025
Accepted: December 23, 2025
Published: December 26, 2025

Abstract

This paper proposes a hybrid AI framework that integrates technical indicators, fundamental data, and financial news sentiment into a stacked ensemble learning model. The ensemble combines 14 algorithms, including decision tree, gradient boosting machine (GBM), gated recurrent unit (GRU), k-nearest neighbor (KNN), lasso regression, linear regression, long short-term memory (LSTM), multi-layer perceptron (MLP), random forest, ridge regression, recurrent neural network (RNN), spiking neural network (SNN), support vector machine (SVM), and temporal convolutional network (TCN). Each prediction is fused through an MLP to generate final outputs. This stacking approach yielded a maximum return of $4061 from an initial investment of $1000. Our results show that simple linear models generally outperform deep learning models in stock predictions, and single stocks are more predictable than other volatile assets or index stocks.

Keywords

Stock Price Prediction, Fusion Model, Stacking, Ensemble Methods

1. Introduction

The formal analysis and forecasting of stock prices using rigorous mathematical models has surfaced since the last century. Traditionally, stock prediction utilized statistical methods such as stochastic calculus, Brownian motion, pricing models (e.g., Black-Scholes model, binomial options pricing model), and time series analysis. These methods analyze historical price data to identify trends or hidden patterns within the market. Despite their utility, traditional methods often fail to capture the non-linear relationships in stock markets. One such reason is that all these

models require making simplistic assumptions, such as constant volatility and perfect information within historical price data, which is generally not true in the real world.

Machine learning, on the other hand, is more flexible and adapts to the given data set. Artificial intelligence (AI) techniques do not require overly simplistic assumptions and can take in any data, such as news sentiment information and any other relevant information, in addition to historical price data. Furthermore, artificial intelligence models have been experimentally proven to be more accurate than traditional statistical approaches [1]. Hence, the use of machine learning models in stock prediction has surged in popularity.

Linear models are one of the most common AI techniques applied in stock prediction. Linear regression models have demonstrated strong success for stock price prediction [2]. Other linear models, including support vector machine (SVC) models [3], Least Absolute Shrinkage and Selection Operator (LASSO) models [4] [5], Ridge regression models [6], and gradient boosting machines [7], have been shown to surpass most traditional statistical methods in forecasting stock prices. In addition, decision tree models [8] and clustering methods [9] have produced results comparable to those of linear models. Extended models based on decision trees and clustering, such as the random forest model [10] and the K-nearest neighbor model [11], have shown even more potential in predicting stocks.

With the recent advancements in neural networks, the Long Short-Term Memory (LSTM) network has become one of the most popular models for stock prediction. Other types of neural networks have also gained popularity in the game of stock prediction, such as multi-layer perceptron (MLP) [12], convolution neural networks [13], recurrent neural networks (RNN) [14], artificial neural networks (ANN), and temporal convolutional networks (TCN) [15]. These models have shown similar results or even better results than simple linear models and decision trees.

The model proposed in this paper also utilizes stacked generalization, commonly known as stacking (**Figure 1**). Stacking is an ensemble modeling technique that effectively combines multiple base models to enhance predictive performance. Introduced in Wolpert (1992), stacking [16] involves training a meta-model that leverages the predicted outputs from these base models to generate a final prediction. This method has gained popularity for its ability to integrate the strengths of various models, allowing for more robust and accurate predictions across different applications.

Many ensemble methods (algorithms that combine multiple models) combine the power of numerous different algorithms, such as Weng *et al.* (2018) [17], Nti *et al.* (2020) [18], and Li and Pan (2021) [19]. Additionally, ensemble methods include other types of input data, such as financial news sentiment and other stock prices, such as news sentiment analysis in Halder (2022) [20], emotional factors in Zhang *et al.* (2024) [21], and gold prices.

These studies show the evolution of machine learning models in stock prediction, from linear models to ensemble models combining LSTMs and CNNs. This paper continues this development by proposing an ensemble model combining linear models and deep neural networks that take in sentiment analysis scores, fundamental analysis data, and technical indicators.

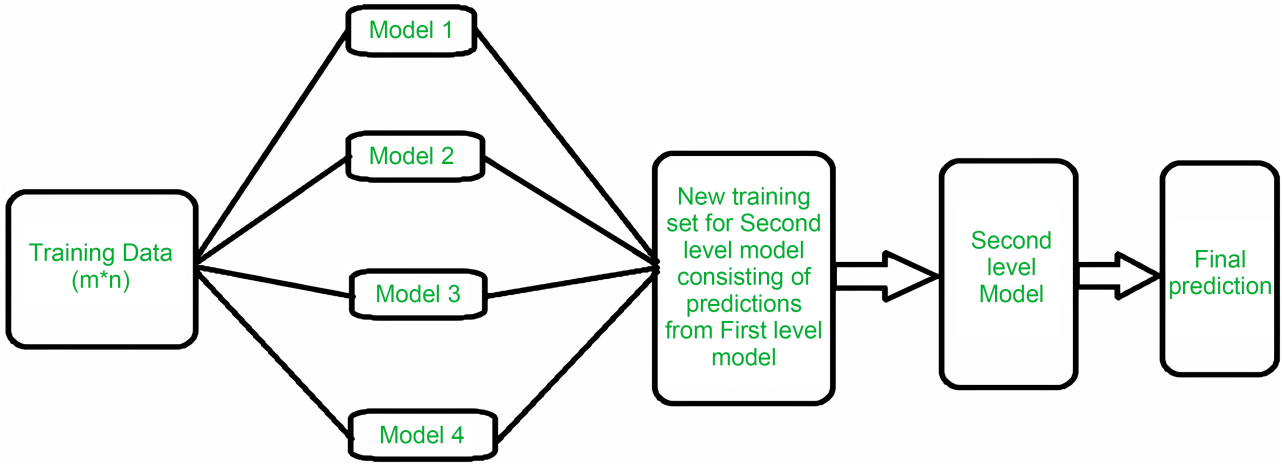

Figure 1. Flow of stacked generalization (*i.e.* stacking).

2. Materials and Methods

The proposed ensemble model contains 3 main parts: The input data integrating historical price data, financial analysis methods, and sentiment analysis scores, the hybrid model containing 14 different deep-learning algorithms, and an evaluation process.

2.1. Preparing and Preprocessing the Dataset

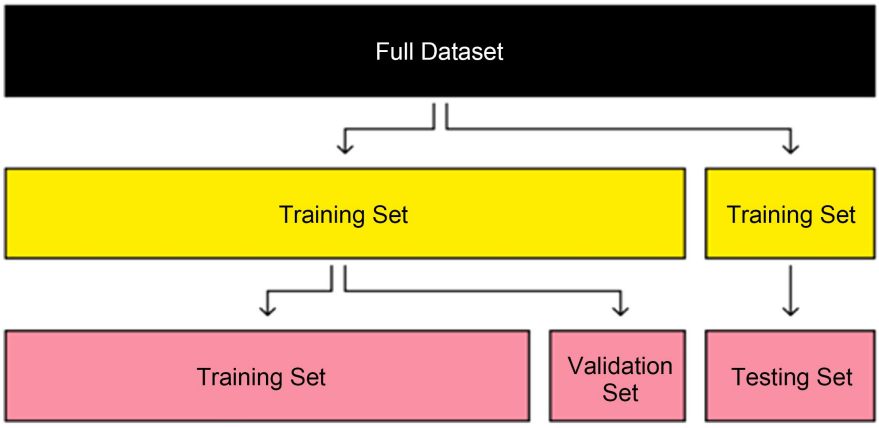

Figure 2. Process of splitting the dataset.

The data is divided into three distinct sets: training, validation, and test sets in 60 30 10 splits (**Figure 2**). The training set is used to fit the model, allowing it to learn from the data. The validation set, separate from the training data, is em-

ployed during the model development process to fine-tune hyperparameters and monitor for overfitting, ensuring that the model generalizes well beyond the data it has been trained on. Finally, the test set, completely isolated from both training and validation, provides an unbiased evaluation of the model's performance on unseen data, simulating real-world applications. This structured approach minimizes the risk of overfitting, enhances generalization, and allows for more reliable and accurate model performance assessment.

2.2. Fundamental and Technical Analysis Data

The historical price data comes from the Yahoo Finance stock library. The data recorded includes "High", "Low", "Open", "Close", "Volume", "Dividends", and "Stock Split". The data starts from the earliest recorded point and ends on August 31, 2024.

There were 21 technical indicators used. As shown in Table 1, they include:

Table 1. Technical indicators used.

Technical Indicator	Formula
Exponential Moving Average (EMA)	$\mathrm{EMA}_t = \frac{2}{15} \times \mathrm{Close}_t + \frac{13}{15} \times \mathrm{EMA}_{t-1} - 1$
Moving Average Convergence Divergence (MACD)	$\mathrm{MACD} = \mathrm{EMA}_{12} - \mathrm{EMA}_{26}$
Relative Strength Index (RSI)	$\mathrm{RSI} = 100 - \frac{100}{1 + \frac{\text{Avg Gain}}{\text{Avg Loss}}}$
Stochastic Oscillator (SO)	$\mathrm{SO} = \frac{\text{Current Close} - \text{Lowest Low}}{\text{Highest High} - \text{Lowest Low}} \times 100$
Awesome Oscillator (AO)	$\mathrm{AO} = \mathrm{SMA}_5 - \mathrm{SMA}_{34}$
Percentage Price Oscillator (PPO)	$\mathrm{PPO} = \frac{\text{12-period EMA} - \text{26-period EMA}}{\text{26-period EMA}} \times 100$
Price Volume Oscillator (PVO)	$\mathrm{PVO} = \frac{\mathrm{EMA}_{12} \text{ of volume} - \mathrm{EMA}_{26} \text{ of volume}}{\mathrm{EMA}_{26} \text{ of volume}} \times 100$
Accumulation/Distribution Line (ADL)	$\mathrm{ADL} = \frac{2 \times \mathrm{Close} - \mathrm{Low} - \mathrm{High}}{\mathrm{High} - \mathrm{Low}} \times \mathrm{Volume} + \mathrm{ADL}_{t-1}$
Boiling Bands (BB)	$\mathrm{BB} = \mathrm{SMA}_{20} - \text{20-day SD of price} \times 2$
Donchian Channel (DC)	$\mathrm{DC} = (20 - \text{dayhighest} + 20 - \text{daylowest}) \div 2$
Ulcer Index (UI)	$\mathrm{UI} = \sqrt{\frac{1}{14}\sum_{i=1}^{14} \mathrm{PD}^2}$, where $\mathrm{PD} = (\mathrm{Close}_i - 14 - \text{period max close}) \div 14 - \text{period max close}$
Average Directional Index (ADI)	$\mathrm{ADI} = (13 \times \mathrm{ADI}_{t-1} + \mathrm{ADI}_t) \div 14$

Continued

Know Sure Thing (KST)	$\mathrm{KST} = \mathrm{SMA}_{10}\left(\mathrm{ROC}_{10}\right) + 2\times\mathrm{SMA}_{10}\left(\mathrm{ROC}_{15}\right) + 3\times\mathrm{SMA}_{10}\left(\mathrm{ROC}_{20}\right) + 4\times\mathrm{SMA}_{15}\left(\mathrm{ROC}_{30}\right),$ where $\mathrm{ROC}_n = \frac{\mathrm{Close}_t - \mathrm{Close}_{t-n}}{\mathrm{Close}_{t-n}} \times 100$
Parabolic Stop and Reverse (PSAR)	$\mathrm{PSAR} = \mathrm{PSAR}_{t-1} - \mathrm{AF}\left(\mathrm{PSAR}_{t-1} - \text{Lowest Low}\right)$, where AF (Acceleration Factor) ranges from 0.02 to 0.2 in increments of 0.02
Aroon Indicator	$\text{Aroon Indicator} = \frac{\text{25-periods since 25-period high}}{25} \times 100$
Ichimoku Cloud (IC)	$\mathrm{IC} = \left(\mathrm{High}_9 + \mathrm{Low}_9 + \mathrm{High}_{26} + \mathrm{Low}_{26}\right) \div 4$
Schaff Trend Cycle (STC)	$\mathrm{STC} = \mathrm{EMA}_3\left\{\left[\mathrm{MACD} - \mathrm{Lowest}\left(\mathrm{MACD},10\right)\right] \div \left[\mathrm{Highest}\left(\mathrm{MACD},10\right) - \mathrm{Lowest}\left(\mathrm{MACD},10\right)\right] \times 100\right\},$ where $\mathrm{MACD} = \mathrm{EMA}_{23} - \mathrm{EMA}_{50}$
Triple Exponential Average (TRIX)	$\mathrm{TRIX} = \text{1-period \% change in } \mathrm{EMA}_{15}\left(\mathrm{EMA}_{15}\left(\mathrm{EMA}_{15}\left(\mathrm{Close}\right)\right)\right)$
Money Flow Index (MFI)	$\mathrm{MFI} = 100 - 100 \Big/ \left(\text{14-period Positive}\left(\text{Typical Price} \times \text{Volume}\right) \div \text{14-period Negative}\left(\text{Typical Price} \times \text{Volume}\right) + 1\right)$
Force Index (FI)	$\mathrm{FI} = \text{13-period EMA}\left(\left(\mathrm{Close}_p - \mathrm{Close}_{p-1}\right) \times \text{Volume}\right)$
Williams %R (WR)	$\mathrm{WR} = \left(\text{Highest High} - \text{Close}\right) \div \left(\text{Highest High} - \text{Lowest Low}\right) \times \left(-100\right)$

2.3. News Sentiment Score

The daily sentiment analysis is obtained by taking the average of all sentiment scores of every piece of news posted that day on Yahoo Finance. The score of each news ranged from −1 to +1, where a negative score indicates a negative connotation and a positive score indicates a positive connotation. The news is processed by the FinancialBERT Large Language Model (LLM) in Harzourli (2022) [22].

3. Results

The Mean Squared Error (MSE) results for the stock prediction using the proposed hybrid model on the following stocks over 50 days are shown in Table 2.

Mean Squared Error (MSE) is used to measure the average squared difference between the actual values (observed data) and the predicted values generated by a model. Mathematically, it is the average of the squared differences between predicted values and actual values for a dataset.

Table 3 shows the stock investment simulation results. It contains the stock invested, initial investment budget, return (or final capital), and the rate of return over 5 years.

Table 2. MSE for stock prediction.

Stock	Model Performance	
	Training MSE	Testing MSE
S&P 500	85.3	232.6
Bitcoin	847.25	59239.79
Amazon	0.45	0.66
JPMorgan Chase & Co.	0.69	1.01
Shell	2.34	3.58
Verizon	0.55	1.45
RTX Corporation	3.45	5.67
General Motors	0.22	0.46
CBRE	0.88	1.89
Google	0.15	0.39

Table 3. Return on investment for simulated stocks.

Stock	Simulated Investment Returns		
	Initial Investment	Return	Rate of Return
S&P 500	1000	1168.2	16.1%
Google	1000	4061.1	306.1%
General Motors	1000	617.5	−38.3%
Shell	1000	1162.0	16.2%
Verizon	1000	1060.2	6.0%
JPMorgan Chase & Co.	1000	1082.0	8.2%

Table 4 shows the individual model performances on Google stock prices.

Table 4. Training and testing MSE for individual components with Google's stock prices.

Model	Individual Model Performance	
	Training MSE	Testing MSE
Linear Regression	0.53	1.23
Decision Tree	0.33	0.63
Random Forest	0.12	0.34
K-nearest neighbor	0.05	0.15
Support Vector Machine	0.00	0.46
Gradient Boosting Machine	0.06	0.30
Lasso Regression	0.96	1.06
Ridge Regression	0.84	1.03
Gated Recurrent Unit	7.06	9.02
Multi-layer perceptron	0.50	5.13
Long Short-Merm Memory	51.64	56.77
Temporal Convolutional Network	16.53	18.98
Spiking Neural Network	3902.33	4210.94
Recurrent Neural Network	9.53	9.98
Multi-layer Perceptron (Combined output)	0.10	0.90

4. Discussion

This TFS model demonstrated an improvement over individual models, achieving a lower Mean Squared Error (MSE) than all neural networks. However, compared to the lowest individual MSE of 0.15 for the KNN model, the hybrid model performs worse, likely influenced by additional complexity and non-linearity. This suggests that linear models, in this case, demonstrated stronger performance than neural networks, which may have introduced more complexity without significantly improving accuracy, thus slightly reducing accuracy in the hybrid model. One major issue with neural networks is that they have many more hyperparameters than simple linear models, and these hyperparameters are not optimized to produce the best prediction. If there were more computational power and time, neural networks would theoretically perform better than simple linear models due to their ability to capture non-linear relationships within the market.

Although the MSE of the TFS model is higher than the individual simple linear models, KNN, and decision tree models, the investment simulation using the TFS model and the KNN model both have identical returns at $4061 for Google. This is because the trading strategy employed is purely directional, relying on the binary decision to buy or sell.

$$\text{Trading Signal} = \begin{cases} \text{BUY}, & \text{if } P_{t+1} > P_t \\ \text{SELL}, & \text{if } P_{t+1} \le P_t \end{cases}$$

In this context, the small difference in MSE (the magnitude of the error) does not affect the prediction enough to change the sign of the forecasted price. Therefore, both models made the exact same series of correct directional decisions throughout the simulation, resulting in identical practical performance and returns, despite having different statistical error metrics.

The performance of some models, particularly the SNN, also suffered due to untuned hyperparameters. The SNN model has an unusually high MSE at around 4000 for both training and testing. Hyperparameter optimization techniques, such as grid search or Bayesian optimization, a technique using a probabilistic model to guide the search, could improve model performance by finding the ideal parameter values. This would help balance the complexity and generalization of the models, ultimately reducing errors. The SNN is also one of the newer models developed without a prebuilt library to use; thus, coding the entire model from scratch may introduce a lot of errors, such as convergence issues and unoptimized architecture and hyperparameters. Therefore, it also introduces a lot of random noise into the final TFS model, decreasing its accuracy.

In addition, Bitcoin and the S&P 500 had an especially high MSE. This is likely due to the volatile nature of cryptocurrency, showing that predicting equity or cryptocurrency with high fluctuations in price is a much harder task, compared to more stable assets like Google or JPMorgan Chase stocks. However, index stocks are far less volatile than single stocks, so theoretically the TFS model should perform better at the S&P 500 as opposed to single stocks. However, the model

performs significantly worse on index stocks. One potential explanation is that there is a lot more noise in the S&P 500 because the model is essentially trying to predict the combined movement of 500 major companies instead of 1 singular company, making the index value highly sensitive to broader macroeconomic trends that may not be reflected in individual firm's stock prices. Therefore, although index funds are much more stable, their trends contain additional variables like high-level macroeconomic policies, such as change in interest rates or GDP, ultimately leading to a worse performance compared to single stocks predictions.

Moreover, the S&P 500 and Bitcoin do not have fundamental data. The model used the individual firm's quarterly revenue, shareholder equity, operating cash flow, EPS, P/E ratio, and dividend yield which index funds and cryptocurrency data did not have. In addition, the S&P 500 news sentiment analysis is the collection of all news regarding the 503 companies, making each piece of news less important to the overall model, and it requires much more computational power to obtain all the news sources and their sentiment. Bitcoin's speculative and unpredictable nature makes it even more difficult for traditional models to forecast accurately.

Regarding the input data, financial news sentiment analysis scores were not standard and hard to find. Much financial news data is not directly related to the stock being predicted, and some news is weighed more heavily than others. Moreover, using Large Language Models (LLMs) to score news sentiment introduces inherent errors and randomness. If one had access to all the available news sources about a specific company and could weigh each one according to how much it affects the stock market, sentiment analysis would contribute a lot more to the model, as opposed to news sentiment analysis scores not having much effect on the model. There was also a lack of sufficient and diverse data, particularly with natural language processing (NLP) inputs for sentiment analysis. NLP models tend to require large datasets to identify meaningful patterns, and in this case, the limited data led to higher error rates. At the same time, there was also insufficient data for news sentiment analysis to yield a concrete conclusion. The model should perform better if it had 20 - 30 years of data to train on, as opposed to 5-10 years of data.

One important takeaway from this analysis is the potential benefit of including more linear models and reducing reliance on neural networks (NNs). Linear models provide greater interpretability and stability, especially when working with datasets where relationships are more straightforward. Neural Networks, while powerful, can be prone to overfitting and may not always be the most effective choice for structured financial data. The large difference between the training MSE and testing MSE in Table 1 indicates overfitting in the model, despite using L2 regularization. It can also be shown from the individual model results that the linear models generally performed better than the neural networks. Therefore, a shift toward incorporating more linear models could yield more consistent results.

Using an autoencoder to preprocess the data turned out to make the model perform worse. A likely reason is that there is already too much noise in the stock market, and that extra noise hinders the performance of the model.

Looking ahead, incorporating additional data sources could enhance model performance. For instance, sentiment analysis from other news sources and social media could provide insights into market behavior, especially for volatile assets like Bitcoin. Similarly, macroeconomic indicators such as interest rates and inflation data could offer more context and improve predictions for traditional financial institutions. Additionally, more frequent data, such as the stock price at every hour instead of every day, could further refine predictions. In conclusion, the model would certainly benefit from an increased volume of data, more accurate data, and more computational power.

Finally, to improve the hybrid model further, weighted averaging could be used to give more importance to models that consistently perform well on specific subsets of the data. Overfitting should also be addressed by adding extra dropout layers and early stopping. Through careful tuning and combination of outputs, the hybrid model can evolve into an even more accurate and versatile prediction system.

5. Conclusion

In this study, we proposed the TriFusion Stacking (TFS) model, a hybrid approach that integrates fundamental data, technical indicators, and sentiment analysis to enhance stock price predictions. Our results show that the proposed TFS model is successful, as it has an MSE of less than 1 for predicting stock prices. However, using a standalone model of simple linear regression or K-nearest neighbor also produces a small MSE. Our findings highlight that linear models can sometimes outperform more complex neural networks when dealing with structured financial data. This suggests a need to re-evaluate the role of simpler models in ensemble strategies for stock prediction. Our findings also highlight that predicting the prices of volatile stocks like Bitcoin and index stocks like the S&P 500 is much more challenging. Future works in improving the model could include better quality news sentiment data and trying different meta-models (the model generating the final prediction, in this case, the MLP). Ultimately, our work contributes to the active field of stock price prediction by demonstrating the effectiveness of combining machine-learning algorithms and an enhanced understanding of which models are more important in stock price prediction and which stock types are more challenging to predict.

Acknowledgements

I would like to thank my mentor Odysseas Drosis for helping me with the model and guiding me through the writing process.

Conflicts of Interest

The author declares no conflicts of interest regarding the publication of this paper.

References

[1] Bhattacharjee, I. and Bhattacharja, P. (2019) Stock Price Prediction: A Comparative

Study between Traditional Statistical Approach and Machine Learning Approach. 2019 4*th International Conference on Electrical Information and Communication Technology* (*EICT*), Khulna, 20-22 December 2019, 1-6. https://doi.org/10.1109/eict48899.2019.9068850

[2] Seethalakshmi, R. (2018) Analysis of Stock Market Predictor Variables Using Linear Regression. *International Journal of Pure and Applied Mathematics*, **119**. 369-377.

[3] Yang, J. (2023) Support Vector Machine-Based Stock Prediction Analysis. *Highlights in Business, Economics and Management*, **3**, 12-18. https://doi.org/10.54097/hbem.v3i.4625

[4] Lee, J.H., Shi, Z. and Gao, Z. (2022) On LASSO for Predictive Regression. *Journal of Econometrics*, **229**, 322-349. https://doi.org/10.1016/j.jeconom.2021.02.002

[5] Roy, S.S., Mittal, D., Basu, A. and Abraham, A. (2015) Stock Market Forecasting Using LASSO Linear Regression Model. In: Abraham, A., Krömer, P. and Snasel, V., Eds., *Advances in Intelligent Systems and Computing*, Springer International Publishing, 371-381. https://doi.org/10.1007/978-3-319-13572-4_31

[6] Li, Y. (2023) Stock Price Prediction Based on Multiple Regression Models. *Highlights in Science, Engineering and Technology*, **39**, 657-662. https://doi.org/10.54097/hset.v39i.6622

[7] Nabi, R.M., Ab. M. Saeed, S. and Harron, H. (2020) A Novel Approach for Stock Price Prediction Using Gradient Boosting Machine with Feature Engineering (GBM-WFE). *Kurdistan Journal of Applied Research*, **5**, 28-48. https://doi.org/10.24017/science.2020.1.3

[8] Kamble, R.A. (2017) Short and Long Term Stock Trend Prediction Using Decision Tree. 2017 *International Conference on Intelligent Computing and Control Systems* (*ICICCS*), Madurai, 15-16 June 2017, 1371-1375. https://doi.org/10.1109/iccons.2017.8250694

[9] Datta, T. and Ghosh, I. (2015) Using Clustering Method to Understand Indian Stock Market Volatility. *Communications on Applied Electronics*, **2**, 35-44. https://doi.org/10.5120/cae2015651793

[10] Zheng, J., Xin, D., Cheng, Q., Tian, M. and Yang, L. (2024) The Random Forest Model for Analyzing and Forecasting the US Stock Market under the Background of Smart Finance. In: Li, K., Li, Q., Hong, W.-C., et al., Eds., *Atlantis Highlights in Computer Sciences*, Atlantis Press International BV, 82-90. https://doi.org/10.2991/978-94-6463-419-8_11

[11] Tajmouati, S., Wahbi, B.E.L., Bedoui, A., Abarda, A. and Dakkon, M. (2024) Applying *k*-nearest Neighbors to Time Series Forecasting: Two New Approaches. *Journal of Forecasting*, **43**, 1559-1574. https://doi.org/10.1002/for.3093

[12] Namdari, A. and Durrani, T.S. (2021) A Multilayer Feedforward Perceptron Model in Neural Networks for Predicting Stock Market Short-Term Trends. *Operations Research Forum*, **2**, Article No. 38. https://doi.org/10.1007/s43069-021-00071-2

[13] Chen, S. and He, H. (2018) Stock Prediction Using Convolutional Neural Network. *IOP Conference Series*: *Materials Science and Engineering*, **435**, Article 012026. https://doi.org/10.1088/1757-899x/435/1/012026

[14] Zhu, Y. (2020) Stock Price Prediction Using the RNN Model. *Journal of Physics*: *Conference Series*, **1650**, Article 032103. https://doi.org/10.1088/1742-6596/1650/3/032103

[15] Guo, S., Ai, H. and Li, S. (2023) Stock Movement Prediction via Temporal Convolutional Network and Interactive Attention Network. 2023 *International Conference*

on Intelligent and Innovative Technologies in Computing, Electrical and Electronics (*IITCEE*), Bengaluru, 27-28 January 2023, 413-417. https://doi.org/10.1109/iitcee57236.2023.10091033

[16] Wolpert, D.H. (1992) Stacked Generalization. *Neural Networks*, **5**, 241-259. https://doi.org/10.1016/s0893-6080(05)80023-1

[17] Weng, B., Lu, L., Wang, X., Megahed, F.M. and Martinez, W. (2018) Predicting Short-Term Stock Prices Using Ensemble Methods and Online Data Sources. *Expert Systems with Applications*, **112**, 258-273. https://doi.org/10.1016/j.eswa.2018.06.016

[18] Nti, I.K., Adekoya, A.F. and Weyori, B.A. (2020) A Comprehensive Evaluation of Ensemble Learning for Stock-Market Prediction. *Journal of Big Data*, **7**, Article No. 20. https://doi.org/10.1186/s40537-020-00299-5

[19] Li, Y. and Pan, Y. (2021) A Novel Ensemble Deep Learning Model for Stock Prediction Based on Stock Prices and News. *International Journal of Data Science and Analytics*, **13**, 139-149. https://doi.org/10.1007/s41060-021-00279-9

[20] Halder, S. (2022) Finbert-LSTM: Deep Learning Based Stock Price Prediction Using News Sentiment Analysis. arXiv:2211.07392.

[21] Zhang, R. and Mariano, V.Y. (2024) Integration of Emotional Factors with Gan Algorithm in Stock Price Prediction Method Research. *IEEE Access*, **12**, 77368-77378. https://doi.org/10.1109/access.2024.3406223

[22] Hazourli, A. (2022) Financialbert—A Pretrained Language Model for Financial Text Mining. Research Gate, 2.