Deep Learning for Stock Market Forecasting: A Review of Models and Challenges

Dhaval Mehta
Computer Engineering Department
Mukesh Patel School of Technology Management and Engineering, NMIMS University
Mumbai, India
dhavalmehta5604@gmail.com

Nipun Dahiya
Computer Engineering Department
Mukesh Patel School of Technology Management and Engineering, NMIMS University
Mumbai, India
nipundahiya2004@gmail.com

Ishani Saha
Computer Engineering Department
Mukesh Patel School of Technology Management and Engineering, NMIMS University
Mumbai, India
ishani.saha@nmims.edu

Payal Mishra
Computer Engineering Department
Mukesh Patel School of Technology Management and Engineering, NMIMS University
Mumbai, India
payal.mishra@nmims.in

***Abstract*— Stock price prediction is a very thought-provoking job due to the inbuilt complexity, non-linearity, and volatility of the financial market. Various AI/ML methodologies are developed to improve the accuracy of the forecasted stock price to be used as a helpful tool for traders and investors. The literature review covers various AI algorithms such as XGBoost, RNN, LSTM, ANN, CNN, DQN, and K-Nearest Neighbor that analyze the effectiveness of those algorithms and their respective limitations. It suggests that hybrid models, though promising, still call for more optimization in order to cope with volatility in the market and generalize better to real life.**

Keywords— ANN, XG-Boost, LSTM, CNN, DQN, Algorithms, Stocks, Prediction, Price.

I. INTRODUCTION

Stock price prediction is a huge task since financial markets are inherently complex, nonlinear, and volatile. In developing lots of machine learning and AI methodologies, scientists have been investigating models targeted to enhance the precision and effectiveness in the forecasting of stock prices, which is a very good and important tool for traders and investors to support them in making better decisions. Stock price forecasting has been considered with substantial use of a number of AI/ML models, having some benefits and drawbacks. The most popular ensemble learning algorithm, XGBoost, was very efficient at stock price movement prediction owing to its handling capability for big datasets and unbalanced data. However, the algorithm is very expensive from a computational point of view and requires very intricate tuning of hyperparameters [1][3][4]. Since they are able to encode in sequential patterns, RNNs and LSTMs find widespread applications in time-series prediction. RNNs suffer from the problem of vanishing gradients, while LSTMs provide more reliable modeling but at higher computational cost. The application of ANN has also found some success in finding nonlinear trends in stock data. ANNs also enhance the traditional models like ARIMA to ensure better forecasting accuracy. However, they suffer from overfitting and are sensitive to the quality of the data[8][9][10]. Since CNNs have been dominating in extracting features, and primarily used in processing images, they have even been used to predict the trends prevailing in the market. However, they are unable to learn the meaning of long-term dependencies unless these networks are combined with other types of models, such as LSTM [12][13][14]. DQN is a reinforcement learning approach that attempts to optimize trading strategies by gaining experience interacting with the market; however, the approaches require substantial computational resources [21][23][24]. K-Nearest Neighbor is one of the simpler effective techniques that work on minimal datasets and it also finds its application in regression and stock price classification. However, for larger datasets, it performs worse, since this becomes computationally very inefficient during the prediction stage [9][10][11]. Even with the advances achieved with these methods, there are still a lot of questions that have not been answered, particularly in managing market volatility, increasing efficiency of the model, and applicability in the real world. Though hybrid models such as ANN-ARIMA tried to solve some of these problems, optimization is still needed to be performed to reduce overfitting and improve generalizability [28][30]. Future studies must be focused more on model development that can handle high-frequency data without losing computational efficiency[19][20][21].

II. LITERATURE REVIEW

I. Brief Discription

Various AI and machine learning procedures have been employed in the forecast of stock prices. The strengths and weaknesses of every architecture differ from the other. Ensemble learning, like XGBoost, represents very good efficiency in handling huge datasets with imbalanced data distribution but still has its disadvantages by being complex for hyperparameter tuning and also memory-intensive. RNNs get good results for temporary dependencies in time series forecasting and capture the problem of learning long-term trends. While LSTMs extend the functionality of RNNs by handling volatile data and long-term dependencies, these also require a lot of computer resources. ANNs, in particular, have been combined with ARIMA and similar models to capture the complex nonlinear patterns but are generally prone to overfitting if they are not tuned properly. CNNs work well in extracting features from financial data, though performing poorly for long-term dependencies without integrating LSTM or any other models to help it with that. Although DQN is very promising for the optimization of trading strategies by reinforcement learning, because it requires heavy training with a huge number of samples, it usually takes quite some time for training and, therefore, more computational resources. Last but not least, K-NN performs well when applied to small datasets, while for big data, performance significantly deteriorates due to the extensive computational complexity during prediction.

Table 1.1: Comparison of Machine Learning and Deep Learning Models for Stock Price Prediction.

Method	Pros	Cons	Findings
XGBoost [1][2][3][4]	– Regularization helps prevent overfitting – High predictive accuracy with large datasets – Excellent feature selection capabilities	– Complex hyperparameter tuning required – Memory-intensive for large datasets – Can be slower in execution than simpler models	– Outperforms ARIMA and LSTM in stock price prediction, particularly when combined with LightGBM – Works well with imbalanced datasets – Shows robustness when handling outliers in stock data
RNN [5][6][7][8]	– Good for sequential data and time-series prediction – Can absorb short-term dependencies efficiently – Captures temporal patterns in stock prices	– Struggles with long-term needs due to disappearing gradients – Computationally intensive – Overfitting risk in small datasets	– Achieves 89% accuracy in short-term stock price predictions – Needs to be paired with other techniques like LSTM for longer sequences – Shows promise in short-term financial predictions
LSTM [18][19][20]	– Captures long-term dependencies in stock price data – Handles volatile and non-stationary data well – Reduces vanishing gradient issues found in RNN	– Computationally intensive and requires more resources than RNN – Risk of vanishing/exploding gradients during training – Can be slow to converge, especially with large datasets	– Outperforms traditional RNNs, especially in volatile markets like during the COVID-19 pandemic – Best for financial markets with sudden changes – Effective for long-term trend prediction
ANN [27][28][29][30][31]	– Learns complex, non-linear patterns in stock data . – Adaptable to different financial time-series data – Performs well in hybrid models (e.g., ANN + ARIMA)	– Overfitting risk if not properly tuned – Highly sensitive to data quality and initial weights – Requires significant tuning to achieve optimal performance	– The hybrid ARIMA-ANN model improved forecasting accuracy – Suitable for capturing non-linear stock data trends – Can handle sudden market changes effectively in hybrid models
CNN [12][13][14][15][16][17]	– Great at feature extraction from financial time-series data – Outperforms traditional models in short-term stock trend prediction – Can model local dependencies well	– Sensitive to noise and requires clean data – Limited in capturing long-term dependencies without other models like LSTM – Computationally expensive for large datasets	– Performed better in short-term stock trend predictions, especially when combined with LSTM – Good for analysing sudden short-term changes – Best when used in hybrid models for sequential data
DQN [21][22][23][24][25][26]	– Learns optimal trading strategies using reinforcement learning – Can incorporate multi-scale data like short-term and long-term trends – Effective at balancing risk and reward in trading	– High training time due to complexity – Risk of overfitting to historical data – Requires large computational resources for optimal performance	– Improved decision-making and profitability in stock trading strategies – Incorporates multiple time scales effectively Shows potential in real-time trading simulations

KNN [9][10][11]	– Simple and intuitive model with no training phase required – Performs well on smaller datasets – Can handle both classification and regression tasks	– Sensitive to noisy data and outliers – High computational cost during prediction phase as it stores the whole dataset – Struggles with large datasets due to increased time complexity	– Showed moderate performance in stock price prediction – Effective in handling smaller datasets but struggles with larger, more complex data – KNN combined with dimensionality reduction techniques like PCA improved accuracy

II. Research Gaps

Limited Real-world Applicability and Generalization:
Real-world applicability and generalization of AI models like XGBoost and LSTM show good performance in a controlled environment, but in the real world of high-frequency trading, they falter. For example, DQN appears very promising through dynamic trading simulations, but it tends to overfit into a real live market and requires computational resources that are computationally intensive. This is particularly for developing models capable of robust real-time predictions that could effectively exploit the dynamic data, addressing sudden market changes [3][5].

Handling Long-Term Dependencies and Market Volatility:
Handling Long-Term Dependence and Volatility in the Market RNNs and LSTMs work effectively in mapping the pattern in stock prices. However, they face great challenges due to long-term dependencies and market volatility. LSTMs outperform RNNs but consume a lot of time in training, besides computer resources, especially in highly volatile markets such as those that appeared during the COVID-19 pandemic period [18][19]. Future research works shall, therefore, make these models efficient enough for high-frequency trading [6][7].

Hybrid Model Integration:
Integration of Hybrid Model Hybrid models like ANN-ARIMA may tend to provide better predictability by capturing linear and nonlinear patterns. However, such integration runs the risk of overfitting and cannot handle complicated time-series data easily [4]. The future research should be directed to arriving at an optimal combination of these methods for better generalization.

Also, deep learning models like CNN and DQN complete better in feature extraction and optimization of strategies that are computationally too expensive for practical utilization in real-time trading [12][24]. Future models will have to find a balance between complexity, speed, and accuracy [20][25].

III. Inference

X-Tream Gradient Boosting (XG Boost):
XGBoost perfectly suits financial forecasting because it deals with huge datasets and nonlinear relationships. The ensemble method in this algorithm has overfitting minimization through regularization techniques; hence, pleasing prediction accuracy will be obtained accordingly [1][2]. Because of its flexibility, XGBoost can be integrated with algorithms like LightGBM, which alleviates prediction by leveraging various aspects of data. XGBoost in feature selection can also very efficiently identify impactful predictors of key importance for a truly robust trading strategy [3][4].

Recurrent Neural Networks (RNN):
RNNs are quite efficient when it comes to time-series data and can still be suitable for short-term stock forecasting. However, with long sequences, they face challenges in retaining information. Combining RNNs with LSTM units helps in the performance of long-term forecasting because now one can overcome the vanishing gradient problem [5][6]. Incorporation of exogenous factors, such as economic indicators, can enhance these prediction models [7][8].

Long Short-Term Memory (LSTM):
Long-term sequences of data are remembered by LSTMs, and they capture trends in stock prices, performing well in volatile markets for long-term forecasts [18]. LSTMs, however, require extensive hyperparameter tuning, which is time-consuming and computationally expensive [19]. LSTMs can be further improved by integrating them into ensemble methods or hybrid models like CNN-LSTM to enhance forecasting performance [20].

Artificial Neural Networks (ANN):
ANNs can learn nonlinear trends in stock data, making them applicable to classification and regression tasks. However, ANN is prone to overfitting, especially with small datasets. Techniques like dropout, early stopping, and regularization are necessary to ensure model generalization [27][28]. Hybrid approaches combining ANNs with other models can improve accuracy in complex trading strategies [29][30][31].

Convolutional Neural Networks (CNN):
CNNs are effective in capturing short-term stock movements, especially when used in hybrid models like CNN-LSTM. They work best for structured data but require recurrent layers to handle long-term dependencies [12][13]. Techniques like data augmentation or transfer learning can improve performance, allowing CNNs to capture variations in market conditions [14][15][16][17].

Deep Q-Networks (DQN):
DQN is promising for dynamic trading strategies using reinforcement learning. It adjusts actions based on changing market conditions, making it valuable in finance [21]. However, DQNs require significant computational resources and should be tested in simulations before real-world use [22][23]. Techniques like experience replay improve efficiency by allowing the model to learn from past trades [24][25][26].]

K-Nearest Neighbours (KNN):
KNN is simple and interpretable, working best with smaller datasets and non-time-sensitive predictions. Dimensionality reduction techniques like PCA can enhance its performance [9]. However, KNN is not ideal for high-frequency trading

due to its slower prediction times compared to other algorithms [10][11]. Careful tuning of parameters and distance metrics is required for optimal performance.

IV. Analysis of Cited Papers by Year

The image (Fig 1.1) showcases a bar chart titled "Number of Papers Cited vs. Year," capturing the citation trends of academic papers across various years. It features data from 2013, 2018, 2019, 2020, 2021, 2022, 2023, and 2024, with each year highlighted in a distinct color for easy reference. The chart reveals that the peak number of cited papers reaches an impressive 10, signaling a notable interest in certain years, while other years present a mix of citation counts. This visual representation not only sheds light on the academic impact of these research publications but also reveals how citation frequency has evolved over time, giving us a glimpse into the changing landscape of academic interest and relevance.

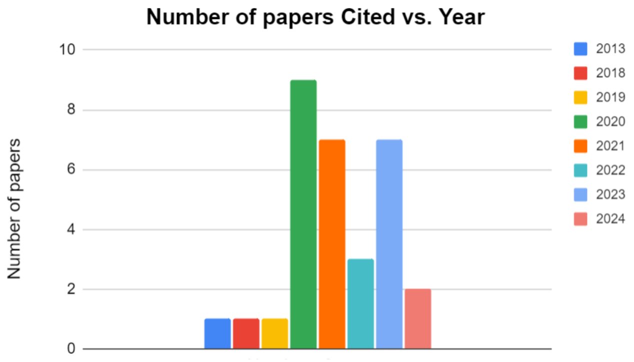

Fig 1.1: Distribution of Cited Papers by Year

III. Proposed Methodology

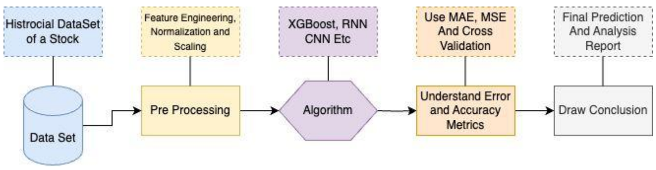

Fig 1.2: Proposed Methodology

Historical Dataset: Prepare a prepared, diversified historical dataset of any particular stock to ensure quality and quantity of data in order to make the model more robust.
Pre-Processing: Cleaned data set like normalizing value, removing null values, etc.
Apply Algorithm: Implementation of the different like XGBoost, RNN, CNN, ANN
Understand Error and Accuracy Metrics: Use metrics like MAE, MSE, and cross-validation to assess model performance and identify overfitting or underfitting.
Draw Conclusion: Aggregate the performance metrics to find the best model with a good balance of complexity and efficiency over here.
Final Prediction and Analysis Report: Put together a comprehensive report complete with visualizations and actionable insights in order to document one's findings and effectively engage each stakeholder. Identify further scope where improvement and research are possible.

The following reasons underline the basis for our methodology:
Time Variations: As our dataset offers at least four different durations of the same stocks, we are not limited to the superficial analysis of performance but can identify those algorithm performances that excel across different time frames. This provides a strong comparison of algorithmic behavior both from a long-term and a short-term perspective.
Quantitative Approach: We will ensure, in most of our analyses, that the outcomes are quantitative. More importantly, we shall be in a position to calculate with high precision the performance for each of the algorithms above introduced by using accuracy/error metrics. This gives us an objective insight into their effectiveness.
Explore Temporal Dynamics: The method will be developed so it can portion the performance of algorithms' evolution with regard to time. This will showcase how each model will adapt to market fluctuations and temporal changes with greater detail.
Real-World Efficiency: The research is based on bridging the gap between theoretically obtained results and those actually obtained in the real world. Because this research focuses on time-sensitive stock data for applications that can be deployed in real time, the methodology ensures findings relevant to actual financial markets, therefore enhancing practical utility of models.

IV. Conclusion

The application of AI as well as machine learning algorithms in the area of stock price prediction is very immense. There are many models, such as XGBoost, RNN, LSTM, ANN, CNN, and DQN, that have already shown remarkable success. Each of the algorithms has an advantage: XGBoost functions much better on big data, while RNNs and LSTMs model sequence and long-term dependencies, and even hybrid models like ANN-ARIMA, a combination of strengths to increase accuracy. But overfitting, inefficiency with computational resources, and real-world applications often raise their own set of issues. As a matter of fact, high-frequency trading environments and volatile markets reveal the Achilles' heel of these models, requiring high computational resource demands and complex hyperparameter tuning. Hybrid models show promise but still need optimization to avoid overfitting and enhance generalization. Future directions of research must be sure that the models being developed are computationally efficient in handling real-time data and volatility. This would be important in developing a more applicable and reliable AI-driven stock price prediction for live market conditions.

V. REFFERENCES

[1] Yang, Y. Wu, P. Wang, and X. Jiali, "Stock price prediction based on XGBoost and LightGBM," E3S Web of Conferences, vol. 275, p. 01040, Jan. 2021. doi: 10.1051/e3sconf/202127501040.

[2] A. B. Gumelar et al., "Boosting the accuracy of stock market prediction using XGBoost and Long Short-Term Memory," in Proc. 2020 International Seminar on Application for Technology of Information and Communication (iSemantic), Sep. 2020. doi: 10.1109/isemantic50169.2020.9234256.

[3] J. Yao, "Stock prediction of Google based on ARIMA, XGBoost and LSTM," BCP Business & Management, vol. 44, pp. 414–421, Apr. 2023. doi: 10.54691/bcpbm.v44i.4850.

[4] A. Almaafi, S. Bajaba, and F. Alnori, "Stock price prediction using ARIMA versus XGBoost models: The case of the largest telecommunication company in the Middle East," International Journal of Information Technology, vol. 15, no. 4, pp. 1813–1818, Apr. 2023. doi: 10.1007/s41870-023-01260-4.

[5] A. Moghar and M. Hamiche, "Stock market prediction using LSTM recurrent neural network," Procedia Computer Science, vol. 170, pp. 1168–1173, 2020. doi: 10.1016/j.procs.2020.03.049.

[6] P. Dey, E. Hossain, M. I. Hossain, M. A. Chowdhury, M. S. Alam, M. S. Hossain, and K. Andersson, "Comparative analysis of recurrent neural networks in stock price prediction for different frequency domains," Algorithms, vol. 14, no. 8, p. 251, 2021. doi: 10.3390/a14080251.

[7] J. Zhao, D. Zeng, S. Liang, H. Kang, and Q. Liu, "Prediction model for stock price trend based on recurrent neural network," Journal of Ambient Intelligence and Humanized Computing, May 2020. doi: 10.1007/s12652-020-02057-0.

[8] J. Patel, M. Patel, and M. Darji, "Stock price prediction using RNN and LSTM," Journal of Open Source Developments, vol. 5, no. 3, pp. 26–34, 2018.

[9] S. Gao, "Trend-based K-nearest neighbor algorithm in stock price prediction," Atlantis Highlights in Computer Sciences, pp. 746–756, 2023. doi: 10.2991/978-94-6463-304-7_78

[10] K. Alkhatib, H. Najadat, I. Hmeidi, and M. K. A. Shatnawi, "Stock price prediction using K-nearest neighbor (kNN) algorithm," International Journal of Business, Humanities and Technology, vol. 3, no. 3, pp. 32-44, Jan. 2013. [Online].

[11] S. Reddy, M. Praneeth, K. P. Reddy, and A. S. Reddy, "Stock market trend prediction using K-nearest neighbor (KNN) algorithm," International Journal for Innovative Engineering & Management Research, vol. 13, no. 5, 2021. [Online].

[12] Wu, J. M., Li, Z., Srivastava, G., Frnda, J., Diaz, V. G., & Lin, J. C. (2020). A CNN-based Stock Price Trend Prediction with Futures and Historical Price. A CNN-based Stock Price Trend Prediction With Futures and Historical Price.

[13] K. Bhardwaj, "Convolutional Neural Network(CNN/CONVNet) in stock price movement prediction," arXiv (Cornell University), Jan. 2021, doi: 10.48550/arxiv.2106.01920

[14] Cavalli, S., & Amoretti, M. (2020). CNN-based multivariate data analysis for bitcoin trend prediction. Applied Soft Computing, 101, 107065.

[15] Chandar, S. K. (2022). Convolutional neural network for stock trading using technical indicators. Automated Software Engineering, 29(1).

[16] K. Bhardwaj, "Convolutional Neural Network(CNN/CONVNet) in stock price movement prediction," arXiv (Cornell University), Jan. 2021, doi: 10.48550/arxiv.2106.01920

[17] S. Mehtab, J. Sen and S. Dasgupta, "Robust Analysis of Stock Price Time Series Using CNN and LSTM-Based Deep Learning Models," 2020 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, 2020, pp. 1481-1486, doi: 10.1109/ICECA49313.2020.9297652.

[18] Ko, C., & Chang, H. (2021). LSTM-based sentiment analysis for stock price forecast. PeerJ Computer Science, 7, e408.

[19] Bathla, G., Rani, R., & Aggarwal, H. (2022). Stocks of year 2020: prediction of high variations in stock prices using LSTM. Multimedia Tools and Applications, 82(7), 9727–9743.

[20] G. Bathla, R. Rani, and H. Aggarwal, "Stocks of year 2020: prediction of high variations in stock prices using LSTM," Multimedia Tools and Applications, vol. 82, no. 7, pp. 9727–9743, Feb. 2022, doi: 10.1007/s11042-022-12390-5.

[21] Chen, X., Wang, Q., Hu, C., & Wang, C. (2024). A Stock Market Decision-Making Framework based on CMR-DQN. Applied Sciences, 14(16), 6881.

[22] Gao, L. (2024). Comparison of DQN and double DQN reinforcement learning Algorithms for stock market prediction. In Advances in intelligent systems research/Advances in Intelligent Systems Research (pp. 169–177).

[23] Huang, Y., Lu, X., Zhou, C., & Song, Y. (2023). DADE-DQN: Dual Action and Dual Environment Deep Q-Network for enhancing stock trading strategy. Mathematics, 11(17), 3626.

[24] Carta, S., Ferreira, A., Podda, A. S., Recupero, D. R., & Sanna, A. (2020). Multi-DQN: An ensemble of Deep Q-learning agents for stock market forecasting. Expert Systems With Applications, 164, 113820.

[25] Awad, A. L., Elkaffas, S. M., & Fakhr, M. W. (2023). Stock market prediction using deep reinforcement learning. Applied System Innovation, 6(6), 106.

[26] Chakole, J., & Kurhekar, M. (2019). Trend following deep Q-Learning strategy for stock trading. Expert Systems, 37(4).

[27] Ajoku, K. K., Nwokonkwo, O. C., John-Otumu, A. M., & Oleji, C. P. (2021). A Model for Stock Market Value Forecasting using Ensemble Artificial Neural Network. Journal of Advances in Computing Communications and Information Technology, 2, 1–13.

[28] Musa, Y., & Joshua, S. (2020). Analysis of ARIMA-Artificial Neural Network Hybrid Model in forecasting of stock market returns. Asian Journal of Probability and Statistics, 42–53.

[29] Ma, Q. (2020). Comparison of ARIMA, ANN and LSTM for stock price prediction. E3S Web of Conferences, 218, 01026.

[30] Kumari, B., & Swarnkar, T. (2023). Forecasting daily stock movement using a hybrid normalization based intersection feature selection and ANN. Procedia Computer Science, 218, 1424–1433.

[31] Awad, A. L., Elkaffas, S. M., & Fakhr, M. W. (2023b). Stock market prediction using deep reinforcement learning. Applied System Innovation, 6(6), 106.