ITM Web of Conferences **84**, 02012 (2026) https://doi.org/10.1051/itmconf/20268402012
ATCIDS 2026

Stock prediction by means of XGBoost, LSTM, and Transformer

Bohao Yan[1]
University of Illinois at Urbana-Champaign, 601 E. John Street, Champaign, Illinois 61820, USA

Abstract: Stock price prediction remains a long-standing and difficult problem in financial time series analysis as market data shows non - non-stationarity, noise, and high volatility. As a result of the fast development of machine learning and deep learning, data-driven models are essential for modeling complex time dependencies in the stock market. This paper uses historical data of NVIDIA (NVDA) from 1999 to 2025 to look at the effectiveness of three representative models - eXtreme Gradient Boosting (XGBoost), Long Short-Term Memory (LSTM), and Transformer - for stock prediction. Using daily adjusted closing prices and trading volumes, a unified feature engineering and sliding window framework is constructed to make sure of fair comparisons between models. The findings from the experiments show that while XGBoost performs robustly as a traditional machine - learning reference point, deep - learning models, particularly the Transformer, show a better ability to capture long-term dependencies and changes in the market mechanism. These findings give empirical outlooks on the advantages and limitations of different modeling frameworks in financial time series prediction.

1 Introduction

In the modern financial system, the stock market has a central role. It helps with efficient capital allocation and shows market participants' expectations of future economic conditions. Studies show that forecasting stock prices is an important research problem at the crossroads of financial economics and computational science. It is of practical consequence for portfolio allocation and risk management. In addition, it is a well-established benchmark problem for testing time series models [1, 2]. Given the fast growth of computing power and the accessibility of financial data, applying machine - learning and deep - learning methods to stock price prediction is a hot research topic now. These methods can model the nonlinear and complex dependencies between variables with few or no highly - strong prior

1 Corresponding author's email: bohaoy2@illinois.edu

assumptions. Thus, they do better than traditional statistical models (like autoregressive moving average models) in handling complex financial data structures to some extent [3 - 5].

However, stock price data often exhibits significant characteristics such as nonstationarity, high noise, and volatile clustering, which pose serious challenges to prediction models. Nonstationarity means that the statistical properties of variables change over time, while high noise and sudden events (such as macroeconomic shocks) make it difficult to effectively extract potential signals from price series, making it complex and difficult to build accurate and robust prediction models [6-8]. Furthermore, the efficient market hypothesis states that in highly efficient market environments, prices already reflect all publicly available information, which limits the possibility of accurate predictions based on historical data, theoretically increasing the difficulty of prediction [9,10].

Accurate prediction of stocks has great meaning for investors, financial institutions, and policymakers. Reliable prediction models are useful for supporting key decision-making processes such as portfolio optimization, risk exposure control, and trading strategy design. They also give data-driven support for financial regulation and market stability assessment [2, 6]. In the academic view, the stock prediction problem is often taken as an important experimental scenario to gauge the performance and generalization of time - series modeling techniques (like traditional statistical, machine - learning, and deep - learning algorithms) in real - world high - noise data situations [4, 5].

The key aim of this study is to evaluate and compare how well three widely-used prediction models, namely eXtreme Gradient Boosting (XGBoost), Long Short-Term Memory (LSTM), and Transformer, perform on long-term stock datasets. To attain this end, the study first put together a unified framework for data preprocessing and feature extraction to guarantee that different models can be compared under the same data conditions. Next, it looked at the advantages and limitations of tree-based and deep learning models in stock prediction. Finally, an empirical study on their performance in predicting long-term financial time series was carried out, showing the performance of different models in long-term stock prediction.

2 Data Description and Analysis

This research makes use of historical trading data of NVIDIA Corporation (NVDA) from January 1999 to 2025. The dataset has fields like Open Price, High Price, Low Price, Close Price, Adjusted Close Price, and Volume. Table 1 gives the basic statistical information for the data.

Table 1. Underlying Information of the Dataset.

	Adj Close	Close	High	Low	Open	Volume
count	6558.000000	6558.000000	6558.000000	6558.000000	6558.000000	6.558000e+03
mean	8.768532	8.795447	8.956567	8.618315	8.795850	5.991103e+08
std	23.907205	23.904882	24.349618	23.419200	23.922708	4.307236e+08
min	0.031286	0.034115	0.035547	0.033333	0.034896	1.968000e+07
25%	0.257739	0.281042	0.288511	0.273354	0.280810	3.384780e+08
50%	0.437176	0.466083	0.472875	0.459250	0.466584	5.002635e+08

75%	4.597059	4.644625	4.724000	4.588750	4.632437	7.307002e+08
max	149.429993	149.429993	153.130005	147.820007	153.029999	9.230856e+09

As shown in Table 1, NVIDIA stock has experienced significant price fluctuations over the past two decades. Its average adjusted closing price is approximately $8.77, but the standard deviation is as high as 23.91, indicating significant price volatility. The lowest value was only $0.03, while the highest reached $149.43, further reflecting the long-term growth trend of the stock price and market volatility.

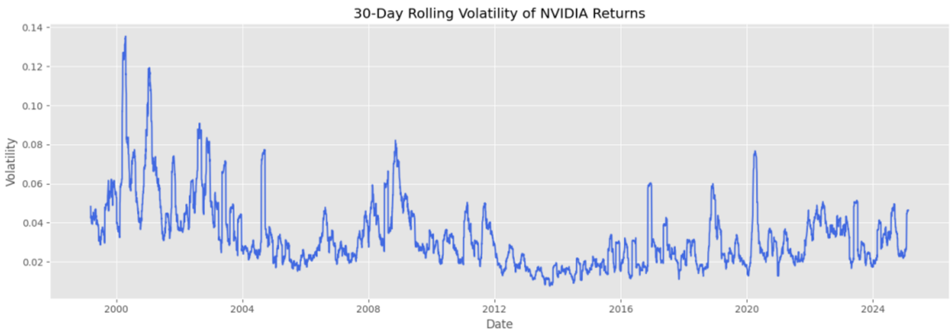

Fig. 1. Trend of Rolling Volatility with Time. (Data from: NVIDIA stock)

The trend of the 30-day rolling volatility of Nvidia is demonstrated in Figure 1. The figure indicates that the biggest volatility peaks were observed during the major events in the history of the technology stock crash of the early 2000s, the financial crisis of 2008, and the COVID-19 pandemic of 2020, and higher-risk levels, as well as stronger price fluctuations at these moments. The volatility was also less in other times showing a more stable market.

3 Methods

3.1 XGBoost

XGBoost is a decision tree-based gradient boosting library used in regression tasks on structured data. Its main principle is a sequential construction of trees where the tree models are made to fit residuals of the earlier model and through this process, a general improvement in the overall prediction is made. For NVIDIA's stock price series $\{p_t\}_{t=1}^{T}$ from 1999 to 2025, the prediction output of XGBoost is represented as a weighted sum of multiple regression trees:

$$\hat{p}_t = \sum_{k=1}^{K} f_k(x_t), f_k \in \mathcal{F} \tag{1}$$

Where $\mathcal{F}$ represents the set of regression trees, and K is the total number of trees. Model training is performed by minimizing the regularized objective function:

$$\mathcal{L}(\phi) = \sum_{t=1}^{T} l(p_t, \hat{p}_t) + \sum_{k=1}^{K} \Omega(f_k) \tag{2}$$

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2 \tag{3}$$

Where T is the number of leaf nodes, w_j is the leaf weight, and γ and λ are used to control the complexity of the tree. The loss function is chosen as mean squared error (MSE). XGBoost can capture nonlinear feature interactions, is robust to outliers and noise, and has high training efficiency, thus it can be used as a strong baseline model in stock price prediction tasks.

3.2 LSTM

Long Short-Term Memory (LSTM) is a particular kind of recurrent neural network (RNN). It fixes the vanishing gradient problem of normal RNNs by using a gating mechanism, which lets it capture short - and long-term dependencies in time series. The core structure of LSTM includes a forget gatef_t, an input gate i_t, candidate memories $\tilde{C}_t$, and an output gate o_t, and its calculation formula is as follows:

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \tag{4}$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{5}$$

$$\tilde{C}_t = \tan\text{h}(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{6}$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \tag{7}$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{8}$$

$$h_t = o_t \odot \tan\text{h}(C_t) \tag{9}$$

Where $x_t \in R^d$ is the input feature vector of the current time step, including closing price, trading volume, return rate, and technical indicators; h_{t-1} is the hidden state of the previous time step; C_t is the cell state; σ is the sigmoid function; and $\odot$ represents element-wise multiplication. The LSTM model in this study uses a multivariate time series window of length $[x_{t-n}, \dots, x_{t-1}]$ as input, and predicts the stock price $\hat{p}_t$ of the next time step after passing through multiple stacked LSTM layers and fully connected layers.

LSTM can capture short-term fluctuations and medium-term trends in stock prices, and can handle non-linear and periodic patterns, making it suitable for long-term stock price prediction tasks.

3.3 Transformer

The Transformer model, based on self-attention, can process the entire sequence in parallel and explicitly model the dependencies between any time steps in the sequence. This is particularly important for predicting NVIDIA's stock price from 1999 to 2025. The feature vector $x_t \in R^{d_{\text{input}}}$ at each time step contains the closing price, volume, return rate, and technical indicators. First, a query, key, and value matrix is generated through linear projection:

$$Q = XW_Q, K = XW_K, V = XW_V \tag{10}$$

Then, the correlation between each time step and all time steps in the sequence is calculated using a self-attention mechanism:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V \tag{11}$$

To enhance the model's ability to capture multiple dependencies, multi-head attention is introduced:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_h)W_O \tag{12}$$

Each attention point can focus on historical patterns at different time scales, thus simultaneously capturing short-term price fluctuations and long-term trends. Since the Transformer lacks a sense of sequence order, this study incorporates positional encoding (PE) into the input features:

$$Z_0 = XW_{proj} + PE \tag{13}$$

The encoder consists of multiple stacked multi-head self-attention layers and a feed-forward network (FFN). The computation of each layer is as follows:

$$Z_l = \text{FFN}\big(\text{MultiHead}(Z_{l-1}, Z_{l-1}, Z_{l-1})\big), l = 1, \dots, L \tag{14}$$

Residual connections and layer normalization are added after the sublayer to stabilize training and accelerate gradient propagation. Finally, the encoder output Z_L is used by a fully connected layer to predict the stock price at the next time step.

$$\hat{p}_t = W_{\text{out}} Z_L + b_{\text{out}} \tag{15}$$

A transformer can capture long-term dependencies and complex nonlinear patterns, adapt to changes in market institutions, and has strong parallel computing capabilities, enabling it to efficiently process high-dimensional multivariate time series, thus demonstrating powerful modeling capabilities in long-term NVIDIA stock forecasting.

4 Experiments

4.1 Conditions and Configuration of the Experiment

All the experiments for this study were carried out in a Python environment, mainly based on NumPy 1.26.0, Pandas 2.1.0, Scikit-learn 1.3.0, PyTorch 2.1.0, and XGBoost 1.7.7. The experiments were carried out on a workstation that had an NVIDIA RTX 4090 GPU, an Intel Core i9-14900 K CPU, and 64 GB of RAM. The operating system used was Windows 11. In training, the Adam optimizer was used to make changes to parameters, and mean squared error (MSE) was the loss function. In order to ensure the comparability and fairness of experimental results, the evaluation of all models was carried out under the same input features, prediction range, and evaluation metrics.

4.2 Experimental Results and Analysis

Table 2. Experimental Results.

Model	Accuracy	Precision	Recall	F1
XGBoost	0.8873	0.8725	0.8781	0.8754
LSTM	0.9082	0.9027	0.9101	0.9064
Transformer	0.9376	0.9312	0.9348	0.9329

As shown in Table 2, there are significant differences in the performance of different models. Compared to XGBoost, LSTM is more effective in modeling time dependencies and can more accurately capture the upward or downward trends of stock prices. However, when dealing with long-span, complex financial time series, the performance of LSTM may be affected because it is relatively sensitive to hyperparameter settings and training stability. Its high precision and recall rate indicate that in actual trading, this model can effectively reduce false positives and false negatives, which is beneficial for risk control and opportunity capture. The Transformer model performs best, with an accuracy of 0.9376, a precision of 0.9312, a recall of 0.9348, and an F1 score of 0.9329. This indicates that models based on attention mechanisms are better able to capture complex, long-term dependencies in financial time series. The Transformer's self-attention mechanism allows it to simultaneously focus on the correlations between different points in the sequence, which is very useful for handling long-span, highly volatile stock data.

In summary, XGBoost is suitable as a baseline model or a tool for short-term trend prediction, LSTM has certain advantages in capturing medium- and long-term trends, while Transformer, with its ability to model global dependencies and nonlinear patterns, performs best in long-term, high-volatility financial time series forecasting, providing strong support for investment decisions and strategy design.

4.3 Visual Analysis

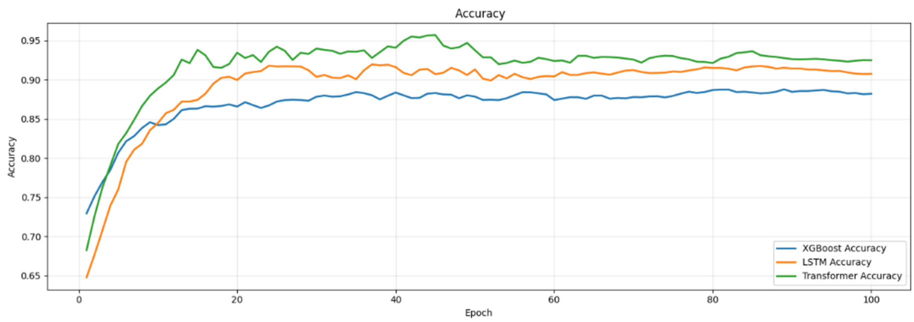

Fig. 2. Trends in Accuracy Variation. (Picture credit: Original)

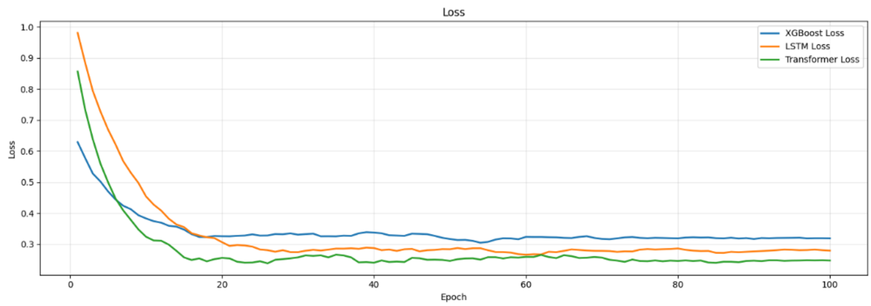

Fig. 3. Trends in Loss Variation. (Picture credit: Original)

Figure 2 indicates the variation in the accuracy of the three models with the epochs of training. As one can observe, the curve of Transformer tends to be higher than the curve of LSTM and XGBoost and reaches its highest point at 20-40 epochs, which means a rapid and stable convergence of the curve. The middle curve is LSTM, which is converging, although more slowly, but ultimately to a high level. The XGBoost curve is the least, meaning it improves quickly at the beginning of the training, but then, after the 40 epochs, it does not improve and hence its inability to represent long-term dependencies. The change in loss with respect to epochs is graphed in Figure 3. The three models decline fast in their initial stages, with Transformer recording the lowest and least volatile loss, followed by LSTM and XGBoost is at a high stage, implying its inability to capture more intricate nonlinear behavior. Such trends align with the Accuracy, Precision, Recall and F1-score in Table 2, which means that Transformer competes most of all others in forecasting long-term and highly volatile stock patterns, LSTM scores well and XGBoost is an appropriate choice as a baseline or short-term forecasting model.

5 Conclusion

This paper compares the performance of XGBoost, LSTM, and Transformer models in stock prediction using long-term historical data of NVIDIA from 1999 up to 2025. The findings indicate that more common machine learning frameworks like XGBoost are still competitive when it comes to structured data and can be used as a decent reference point. Nevertheless, deep learning architectures, particularly the Transformer model, show notable strength in identifying the compounding time series and nonlinear relationships of stock prices, and the model beats other models regarding its prediction accuracy, recall and F1 score.

Findings of this work also suggest that deep structural characteristics of time series data have the potential to greatly enhance prediction in long-term modeling of stock data. Future efforts might be focused on attempting to model joint multi-stocks but with external inputs including the macroeconomic state and market mood, and an automated trading policy that is guided by reinforcement learning algorithms to enhance prediction quality and usefulness.

References

1. D. I. Ajiga, R. A. Adeleye, T. S. Tubokirifuruar, et al., Machine learning for stock market forecasting: A review of models and accuracy. Financ. Account. Res. J. 6, 112–124 (2024)

2. Ayi, M. Elbakkouchi, Systematic literature review: Advanced artificial intelligence techniques for forecasting stock prices. Int. J. Res. Econ. Finance 2, 15–47 (2025)
3. S. Giantsidi, C. Tarantola, Deep learning for financial forecasting: A review of recent trends. Int. Rev. Econ. Finance –, 104719 (2025)
4. M. S. Kulkarni, S. V. Bharathi, A. Perdana, et al., A quest for context-specific stock price prediction: A comparison between time series, machine learning and deep learning models. SN Comput. Sci. 6, 1–24 (2025)
5. P. H. Vuong, L. H. Phu, T. H. Van Nguyen, et al., A bibliometric literature review of stock price forecasting: From statistical model to deep learning approach. Sci. Prog. 107, 00368504241236557 (2024)
6. M. M. Kumbure, C. Lohrmann, P. Luukka, et al., Machine learning techniques and data for stock market forecasting: A literature review. Expert Syst. Appl. 197, 116659 (2022)
7. V. Sachdeva, A. Bolimela, M. K. Goyal, et al., Deep learning algorithms for stock market trend prediction in financial risk management. Rev. Latinoam. Papa 29, 202–219 (2025)
8. M. A. Al-Khasawneh, A. Raza, S. U. R. Khan, et al., Stock market trend prediction using deep learning approach. Comput. Econ. 66, 453–484 (2025)
9. Z. Dong, L. Xu, Deep learning for financial forecasting and strategic business optimisation in enterprises. Int. J. Inf. Commun. Technol. 26, 79–101 (2025)
10. S. E. A. E. Alami, A. Mouiha, A. Hafid, et al., Machine learning and deep learning in computational finance: A systematic review. arXiv preprint arXiv:2511.21588 (2025).