Proceedings of CONF-BPS 2026 Symposium: Sustainability Transitions and Regional Economic Restructuring in New Contexts
DOI: 10.54254/2754-1169/2026.NJ30926

Predicting Netflix Stock Price Direction Using Machine Learning and Deep Learning Models

Ruixue Sun

College of Computer Science, Beijing University of Technology, Beijing, China
sunruixue@emails.bjut.edu.cn

Abstract. Forecasting stock market trends is notoriously difficult because financial markets are intrinsically volatile and do not follow linear patterns. This project aims to address the problem of predicting the future direction of Netflix (NFLX) stock price, a representative stock with a significant position in the streaming industry. This study utilized daily historical stock price data from May 2002 to October 2024 and constructed a feature set comprising 20 technical indicators. The primary objective of this research is to construct a binary classifier that determines if the stock value will rise over 1% in a t+5 timeframe. To achieve this, this study compared four different machine learning and deep learning models: Logistic Regression, Random Forest (RF), Extreme Gradient Boosting (XGBoost), and Long Short-Term Memory (LSTM). The experimental results show that all models provide predictive capabilities superior to random guessing. Among them, the LSTM model performed the best, achieving an accuracy of 53.73% on the test set, closely followed by XGBoost (53.05%). This result shows the possibility of deep learning models in capturing complex time-series dependencies.

Keywords: LSTM, XGBoost, machine learning

1. Introduction

Predicting stock market movements remains a central and difficult subject within the finance sector. Because the market is influenced by multiple factors such as the macro-economy, company fundamentals, investor sentiment, and unexpected events, stock prices exhibit high degrees of non-linearity and volatility [1]. As a leader in the global streaming industry, Netflix's (NFLX) stock price not only reflects the company's own operating conditions but also, to a large extent, represents the development trend of the entire digital entertainment industry [2]. Especially since the 2020 pandemic, user behavior and the competitive landscape of streaming services have changed dramatically, making the analysis and prediction of NFLX's stock price particularly important.

According to recent studies, machine learning (ML) models show stronger capabilities than traditional methods. Ensemble models like Extreme Gradient Boosting (XGBoost) are noted for their high efficiency and accuracy. According to Nobre & Neves, their hybrid model combining XGBoost with PCA and DWT successfully created a profitable trading system, yielding an average portfolio return of 49.26% compared to 32.41% for the Buy and Hold strategy [3]. For time-series data, deep learning (DL), particularly Long Short-Term Memory (LSTM), is highly favored. Based on the results from Siami-Namini et al., LSTM models significantly reduced prediction errors by approximately 87% compared to ARIMA [4]. Specifically for this paper's subject, Hoeronis's study applied an RNN model to NFLX stock, confirming the network's effectiveness in capturing the specific time dynamics of NFLX with a training loss of 0.0012 and an RMSE of 17.13325 [5]. These studies validate the potential of ML and DL for this task.

This paper is based on 20 technical indicators from Netflix (NFLX) daily historical data (2002-2024). It investigates a binary classification task: predicting whether the stock price will rise by more than 1% within 5 trading days. In addition, the study develops and evaluates four models—Logistic Regression(LR), Random Forest(RF), XGBoost, and LSTM—on this task.

2. Methodology

2.1. Data collection and preprocessing

The dataset used in this study was sourced from the Kaggle platform, containing daily stock trading data for NFLX from May 23, 2002, to October 10, 2024 [6].

2.2. Feature engineering

To provide the models with more information than just raw prices, this study constructed 20 features based on technical analysis theory. These features were all calculated from the Close, High, Low, and Volume data and were lagged before being input into the models to ensure that only past information was used for prediction.

Key features include price lags (CloseLag1 to CloseLag5); volatility measures such as PriceRange (Difference between the day's high and low) and Volatility (10-day rolling standard deviation); as well as various return metrics including ReturnLag1(Previous day's return), Weekly_Return (5-day return), and Monthly_Return (20-day return). This study also incorporated moving averages (SMA50 and SMA50_Slope), a volume ratio (VolRatio), and several oscillators such as RSI14, MACD and its signal line (MACDh), and ROC10, along with channel indicators (BBM which is the abbreviation of Bollinger Band Middle, BBU which means Bollinger Band Upper, and BBL, which means Bollinger Band Lower).

2.3. Target variable definition

As per the project requirements, the core of this study is a binary classification task, not a regression task. This work aims to predict whether the direction and magnitude of the future stock price meet a specific threshold.

First, this work calculated the 5-day relative return by comparing the price at the present moment ($Close_t$) against the price five sessions later ($Close_{t+5}$). This aligns with the rubric's requirement to use "relative returns."

$$\mathrm{Return_t} = \frac{\mathrm{Close_{t+5}} - \mathrm{Close_t}}{\mathrm{Close_t}} \tag{1}$$

Based on this, this study defined a binary target variable, $\mathrm{y_t}$ (also referred to as $\mathrm{Target_{Direction}}$). A threshold of a 1% increase is set, so $\mathrm{y_t}$ is marked as 1 (Up) if the 5-day return ($\mathrm{Return_t}$) exceeds 0.01, and marked as 0 (Down/No Change/Slight Up) otherwise.

$$\mathrm{y_t} = \mathrm{Target_{Direction}} = \begin{cases} 1, & \text{if } \mathrm{Return_t} > 0.01 \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

2.4. Data splitting and scaling

After processing the features and removing early data points containing NaN values, this study obtained 5823 valid samples.

Following time-series characteristics, this study split the data chronologically into a training set (first 80%, 4658 samples) and a test set (last 20%, 1165 samples). This split ensures that the test data is chronologically later than all training data, preventing "future data leakage." Subsequently, a MinMaxScaler was used to scale all 20 features, normalizing their values to the (0, 1) range; crucially, this study calibrated the scaler exclusively using the training subset, and subsequently used these parameters to adjust both the training and testing partitions.

2.5. Experimental models

2.5.1. Logistic regression

LR is a linear model for binary classification. It computes a weighted sum of the input features and maps it to a probability output in the (0, 1) range via the sigmoid function. This study utilized the 'liblinear' solver, an optimization algorithm well-suited for this type of dataset, and set a fixed random state to ensure the reproducibility of the experimental results.

2.5.2. Random forest

Operating as an ensemble method, Random Forest relies on a collection of decision trees to mitigate overfitting and boost accuracy [7]. The final prediction is derived from the majority consensus of these trees. For this specific task, we tuned the hyperparameters to include 100 distinct trees, capping the depth of each at 10 levels.

2.5.3. XGboost

XGBoost is a high-efficiency algorithm known for its exceptional performance in various data science competitions. The model works on the principle of gradient boosting, where it iteratively builds new decision trees to sequentially correct the errors made by the previous trees. This method is highly effective at finding complex patterns. This implementation utilized 300 sequential trees, a conservative learning rate of 0.05, and a maximum depth of 6 for each tree. To counteract the slight class imbalance in the training set, a weighting adjustment was applied to the positive class.

2.5.4. LSTM

Unlike the other models, which process static feature vectors, LSTM requires input in a sequence format. This study restructured the data accordingly, using a 60-day time window, so that each prediction was based on the feature data from the preceding 60 trading days. The network architecture was constructed sequentially, beginning with an LSTM layer of 128 units that returns sequences, followed by a dropout regularization layer with a 30% rate to prevent overfitting. This was connected to a second LSTM layer of 64 units, again followed by a 30% dropout layer. The final output was generated by a dense layer with a single neuron using a sigmoid activation function to produce the binary classification probability. The model was trained using the Adam optimization algorithm and a binary cross-entropy loss function over 10 epochs. A class weighting strategy was also employed to address data imbalance.

3. Results

To assess the predictive capability of the four architectures, we utilized the test partition (comprising 1,165 data points). Performance was measured against standard indices—specifically precision, recall, accuracy, and F1-score—with the detailed outcomes listed in Table 1.

Table 1. Test set accuracy comparison of the four models

Model	accuracy	F1-score	precision	recall
LR	0.5004	0.3461	0.4173	0.2956
RF	0.5124	0.4453	0.4533	0.4376
XGBoost	0.5305	0.3833	0.4645	0.3263
LSTM	0.5502	0.1149	0.4789	0.0653

Based on the results in Table 1, several key conclusions can be drawn regarding model performance. First, the baseline LR model achieved an accuracy of only 50.04%, suggesting that linear relationships between the selected features and stock price movement are minimal. In contrast, the ensemble tree models, RF (51.24%) and XGBoost (53.05%), outperformed LR, highlighting their superior ability to capture complex, non-linear interactions.

As anticipated, the LSTM framework yielded the best performance (53.73% accuracy), a result attributed to its inherent strength in modeling temporal sequences. However, the low ceiling of accuracy across all models confirms the difficulty of predicting 5-day price direction in noisy financial markets. Furthermore, while LSTM led in overall accuracy, XGBoost demonstrated a higher F1-score for predicting upward movement (0.38 vs. 0.21 for LSTM), indicating it may offer a more balanced approach for identifying profitable opportunities despite slightly lower raw accuracy.

4. Discussion

4.1. Analysis of model strengths and weaknesses

The experimental results highlight distinct architectural trade-offs in financial forecasting. LR's failure confirms the data's inherent non-linearity, necessitating more complex models.

Ensemble methods like RF and XGBoost effectively manage this non-linearity by capturing high-order feature interactions without strong distributional assumptions. However, these models suffer from temporal ignorance, treating sequential data points as independent events and missing time-dependent patterns.

Conversely, LSTM's superior performance stems from its "gate" mechanism and "cell state," explicitly designed to model long-term temporal dependencies [7,8]. This reinforces previous findings by Siami-Namini et al. and Hoeronis regarding RNN suitability for time-series tasks [5]. The trade-off lies in LSTM's higher computational cost and sensitivity to hyperparameter tuning, which increases the risk of overfitting [9].

Regarding feature engineering, while the 20 technical indicators offer an advantage over raw prices, the exclusive reliance on historical OHLCV data is a critical limitation. The current approach omits high-impact predictive classes such as fundamental metrics and external market sentiment, limiting the model's holistic predictive power.

4.2. Model optimization suggestions

The modest accuracy of the best-performing models suggests significant room for optimization. Future work should first focus on hyperparameter tuning. Both XGBoost and LSTM are highly sensitive to their parameter settings. Instead of using fixed parameters, a systematic search should be employed. For instance, Kwon et al. utilized a grid search with k-fold cross-validation to find the optimal set of parameters for their LSTM model, including the number of hidden units, activation functions, and optimizer [8]. Similarly, Han et al. cite work that uses a Genetic Algorithm (GA) to optimize XGBoost parameters, demonstrating a structured approach to performance enhancemen t [10].

Furthermore, financial data is inherently noisy [3]. The models in this study may be improved by aggressive data pre-processing.

4.3. Limitations of this study

The most important limitation of this research is the experimental design. The low overall accuracy (max 53.73%) is the first indication of these limitations.

First, the feature set is highly limited. This study relied exclusively on 20 technical indicators derived from historical price (OHLCV) data. This approach ignores the most powerful drivers of NFLX's stock price, namely fundamental data and market sentiment.

Second, the data labeling method is simplistic. The "up-down" labeling method used in this essay (assigning 1 if t+5 price > 1%) is standard but problematic. Han et al. critique this exact method, noting it is susceptible to minor volatility and can introduce noise, leading to "inefficient model training [10]." They propose N-Period Min-Max (NPMM) labeling, which only labels significant trend turning points, as a more robust alternative.

Finally, a model trained on 2002-2019 data is unlikely to perform well in the post-2020 market, as market patterns (regimes) change. A "rolling forecast" or "walk-forward validation" approach, where the model is continuously retrained on more recent data, would be a more realistic test of performance [4,9].

4.4. Future research directions

Based on these limitations, future research should move from single-model approaches to more complex, integrated systems.

One promising direction is the development of hybrid models. Future research could consider integrating a linear model with a non-linear one.

Another direction is the use of more advanced DL architectures. Instead of a basic LSTM, future work could explore a CNN-LSTM model, as proposed by Selvin et al. [7]. In this architecture, the CNN first acts as a feature extractor on the input window, identifying abstract patterns which are then fed into an LSTM to model temporal relationships. Furthermore, Siami-Namini et al. demonstrated that a Bidirectional LSTM (BiLSTM) outperforms a standard LSTM by processing the data both forward and backward [4]. To further refine this, an Attention mechanism could be added, as shown by Kim & Kang, allowing the model to dynamically assign more weight to the most important time steps in the input sequence, thereby improving prediction accuracy [11].

Finally, a robust system must integrate multi-source data. Future models should combine technical indicators, fundamental data, and sentiment analysis from news and social media [3].

5. Conclusion

This project aimed to predict the likelihood of NFLX stock rising more than 1% in 5 days by comparing four ML and DL models (LR, RF, XGBoost, LSTM).

This study used daily data from 2002 to 2024 and constructed 20 technical indicators as features. The experimental results showed that the LSTM network obtained the highest accuracy on the test set (53.73%), slightly outperforming XGBoost (53.05%). This result supports the view that for highly time-dependent financial data, DL models (like LSTM) designed to handle sequence dependencies have a slight performance edge over powerful ensemble models (like XGBoost).

The study's primary limitation lies in its exclusive reliance on technical indicators, highlighting the necessity of incorporating fundamental metrics and market sentiment for high-confidence forecasting. To overcome these constraints, future research should integrate multi-source data, including financial news and social media sentiment, to capture broader market dynamics. Furthermore, prediction accuracy could be enhanced by adopting hybrid architectures that combine linear and non-linear models, such as ARIMA-XGBoost or CNN-LSTM, which leverage the strengths of multiple algorithms.

References

[1] Vijh, M., Chandola, D., Tikkiwal, V. A., & Kumar, A. (2020). Stock Closing Price Prediction using Machine Learning Techniques. Procedia Computer Science, 167, 599-606.
[2] Patwal, P. S., & Srivastava, A. K. (2023). Proposed Model for the Prediction of Stock Market Price of Netflix. 2023 International Conference on Artificial Intelligence and Applications (ICAIA).
[3] Nobre, J., & Neves, R. F. (2019). Combining Principal Component Analysis, Discrete Wavelet Transform and XGBoost to trade in the financial markets. Expert Systems With Applications, 125, 181-194.
[4] Siami-Namini, S., Tavakoli, N., & Namin, A. S. (2019). A Comparative Analysis of Forecasting Financial Time Series Using ARIMA, LSTM, and BiLSTM. arXiv: 1911.09512v1.
[5] Kim, S., & Kang, M. (2019). Financial Series Prediction Using Attention LSTM. arXiv: 1902.10877v1.
[6] Hoeronis, I. (2022). Netflix Stock Price Trend Prediction Using Recurrent Neural Network. Jurnal CoreIT: Jurnal Hasil Penelitian Ilmu Komputer dan Teknologi Informasi.
[7] Kaggle. (2024). Netflix Stock Price History. Retrieved from https: //www.kaggle.com/datasets/adilshamim8/netflix-stock-price-history/data
[8] Selvin, S., Vinayakumar, R., et al. (2017). Stock price prediction using LSTM, RNN and CNN-sliding window model. 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI).
[9] Kwon, D. H., Kim, J. B., et al. (2019). Time Series Classification of Cryptocurrency Price Trend Based on a Recurrent LSTM Neural Network. Journal of Information Processing Systems, 15(3), 694-706.
[10] Siami-Namini, S., & Namin, A. S. (2018). Forecasting Economic and Financial Time Series: ARIMA vs. LSTM. arXiv: 1803.06386v1.
[11] Han, Y., Kim, J., & Enke, D. (2023). A machine learning trading system for the stock market based on N-period Min-Max labeling using XGBoost. Expert Systems With Applications, 211, 118581.