International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Predicting Stock Markets with AI: A Survey of Surveys

[1]*Divyam Vijay*, [2]*Chirag Bansal*, [3]*Aman Bora*

[1]Department of Applied Mathematics Delhi Technological University divyamvijay_mc21a11_56@dtu.ac.in
[2]Department of Applied Mathematics Delhi Technological University chiragbansal_mc21a11_43@dtu.ac.in
[3]Department of Applied Mathematics Delhi Technological University amanbora_mc21a11_15@dtu.ac.in

ABSTRACT—

This study considers the application of artificial intelligence to stock market prediction through a model that consolidates past price information with popular technical indi- cators. We implement XGBoost regression, a high-performance machine learning methodology known for speed and precision, to predict closing prices of stocks. The dataset provides metrics like 20-day Simple and Exponential Moving Averages (SMA and EMA), Relative Strength Index(RSI), and MACD, which are normally used by traders to analyze market movements. By training and testing with real market data our model achieved an R^2 value of 0.8325, demonstrating strong predictive validity. This result indicates that mixing AI models with technical tools can greatly enhance the accuracy of forecasts and thus supply useful information for investors as well as analysts.

***Index Terms*—Stock Prediction, XGBoost, Technical Analysis, RSI, EMA, SMA, MACD, Machine Learning, AI in Finance**

I. Introduction

For quantitative finance, the most fascinating but challeng- ing task is to forecast stock prices. Stock markets are regulated by a host of factors—macroeconomic indexes, emotions of investors, global news, etc.—so that price volatility becomes intrinsically nonlinear, dynamic, and even unpredictable. Tra- ditional statistical techniques, as crucial as they may be, tend to be less adaptive to handle such volatility. To address this, Artificial Intelligence (AI) has been a powerful partner, which can extract deep insights from mountains of historical stock data [1].

This project ”Stock Analysis using AI” seeks to leverage three alternative modeling approaches—Linear Regression, Long Short-Term Memory (LSTM) networks, and XGBoost Regression—to predict stock prices with the help of an array of datasets (AAPL, GOOG, AI). Each of these models has been chosen with the aim to represent a distinctive learning cate- gory: statistical, deep learning, and ensemble-based machine learning. Comparative study allows us to understand how each model addresses the financial time-series space.

Three various predictive modeling methods were explored and compared in this study to forecast the closing prices of stocks: Linear Regression, LSTM, and XGBoost. The Linear Regression model, the baseline, is a straightforward statistical method [2] that attempts to forecast the closing price of a stock 30 days in the future using solely its current price.In order to overcome the linearity limitations of the models, an LSTM network was used, leveraging its ability to learn temporal sequences [3] and long dependencies. In this case, the model has been trained using 60 days’ worth of closing price data in order to predict the price on the following day. The data was scaled using MinMaxScaler, and the model architecture included two stacked LSTM layers with dropout regularization and a dense output layer. The LSTM model demonstrated a vast improvement in performance across all the datasets. The third approach used XGBoost (Extreme Gradient Boosting), which is a general-purpose ensemble-based ap- proach widely known for dealing with structured tabular data [4]. This model integrated both overall stock features (open, high, low, volume) and technical features unique to the field like the simple moving average (SMA), exponential moving average (EMA), relative strength indexing (RSI) and moving average convergences divergence (MACD). XGBoost handled the nonlinearity in the stock data fairly well and benefitted from feature engineering and regularization methods.

II. Literature Review

The idea of applying AI models to predict stock prices emerged due to certain constraints with conventional statistical methods. These methods presume linearity and may fail to apprehend the complex, nonlinear, and time-dependent aspects that govern financial data. Indeed, a broad dynamic array of factors influences stock prices-from historical trend movement and technical indicators to market behavior. Models such as LSTM can learn from sequential patterns [5], whereas XG- Boost makes use of engineered features quite effectively. This improved accuracy and adaptability in terms of the volatility of the market conditions towards forecasting makes these models even more applicable. Thus, AI offers a powerful way to improve prediction reliability within the area of financial analytics.

In this study, three different stock market datasets were used to perform an analysis and comparison of the predictive capabilities of several models. The data was obtained from reliable financial websites—Yahoo Finance (using the yfinance Python package) and Kaggle—to guarantee reliability and current financial precision. We chose three firms that exhibit different types of market behaviors: AAPL (Apple Inc.), a large and stable technology firm with great trading volume and market stability; GOOG (Alphabet Inc.), another tech giant with considerable historical data and liquidity; and AI (C3.aiInc.), a comparatively new and highly volatile AI software firm. These choices allowed us to evaluate model robustness with varying levels of volatility, stability, and market maturity. All datasets included fundamental financial characteristics like Open, High, Low, Close prices, and Volume, which are imperative for time-series forecasting and technical analy- sis. The duration of coverage ranged from January 2015 to December 2023, providing close to nine years of daily trading data and more than 2,000 records per stock, providing sufficient data for training and testing of models, especially

deep learning models.

III. Methodology

A. *Dataset Preprocessing*

To pre-process the datasets for model training, several pre- processing operations were used. The missing values, usually entries encountered in the first or last slots, were tackled by forward-filling methods or eliminated depending on the ade- quacy of the data. Feature scaling was performed employing the MinMaxScaler to adjust the data on a [0, 1] scale range, which proves particularly useful to models such as LSTM by strengthening numerical stability as well as convergence. The target variable was defined specifically for each model: for Linear Regression, it was defined as the closing price 30 days later (with a shift(-30) transformation); for LSTM, a sliding window approach was taken with the closing prices of the last 60 days to forecast the following day; and for XGBoost, the next-day actual Close price was utilized along with engineered technical features like SMA, EMA, RSI, and MACD for improving prediction ability [6]. These preprocessing choices ensured consistency across models and improved the learning process by standardizing inputs and incorporating temporal and domain-specific insights.

B. *Model Training*

To compare different predictive models in stock price pre- diction, we used three different models: Linear Regression, LSTM (Long Short-Term Memory), and XGBoost Regressor, each of which is a different predictive algorithm.

- **Linear Regression:** Linear Regression was our control model because it is easy to interpret and understand [2]. It makes a linear relationship between input and target features, so it is simple but restrictive in addressing the non-linear and unstable nature of financial markets. In this project, the model was trained to forecast the closing price of the stock 30 days ahead with a shift(- 30) operation. Although simple to deploy with minimal computational expense, its built-in assumptions overlook the sequential interdependence of stock prices and are therefore less appropriate for long-term predictions in changing conditions.
- **LSTM Model:** LSTM, a specific type of Recurrent Neural Networks (RNNs) [7], was used to model tem- poral relationships and intricate patterns that form a part of stock market data. The model was constructed with two stacked LSTM and dropout regularization to avoid overfitting, followed by a dense output layer for making the final prediction. All of the training inputs were a sequence of 60 consecutive days' closing prices, so the model could pick up on prior context and learn from them. The data was normalized by MinMaxScaler prior to being fed into the network. Training was performed over several epochs with a constant batch size, aided by time-series data splits to maintain chronological order. The architecture of the memory cells in LSTM allowed it to access long-term dependencies and thus existed as a robust sequential forecasting

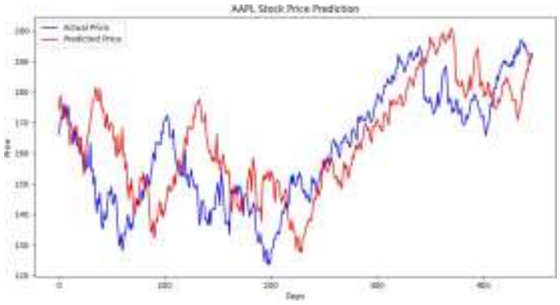

Fig. 1. AAPL STOCK PRICE PREDICTION

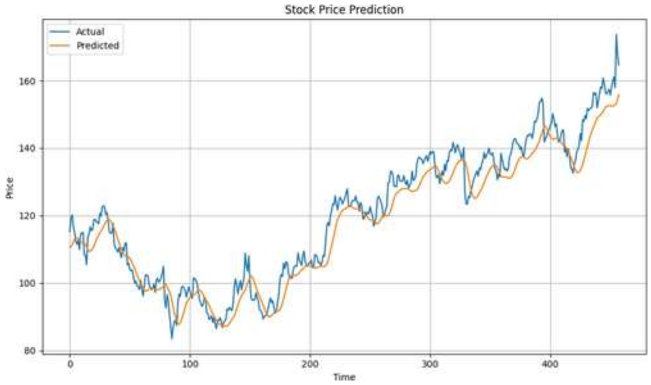

Fig. 2. STOCK PRICE PREDICTION

- **XGBoost Regressor:** We applied the XGBoost gradient boosting decision tree algorithm because of strong perfor- mance on structured data and regularization techniques- based robustness against overfitting. In comparison with LSTM, learning sequential data, XGBoost had strong feature engineering [8]. We supplemented the dataset by incorporating domain-specific technical factors such as Simple Moving Average (SMA), Exponential Mov- ing Average (EMA), Relative Strength Index (RSI), and Moving Average Convergence Divergence (MACD) [6]. These features enabled the model to comprehend short- term and long-term price movements, momentum, and market volatility. The hyperparameters like learning rate, depth, and number of estimators were set in order to gain maximum performance on every set of data. XGBoost can work very well with non-linear relationships and was able to perform well across different stocks; hence it made itself up to be a strong candidate along with LSTM.

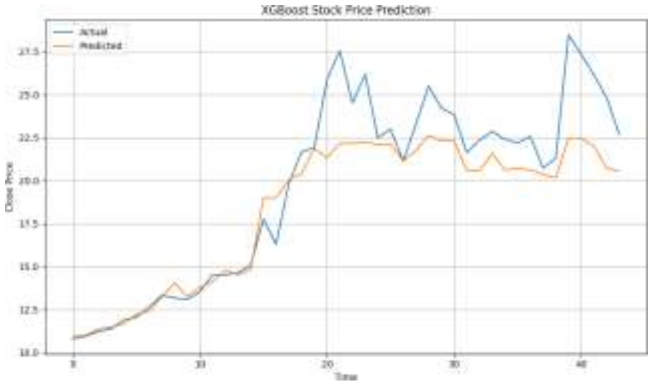

Fig. 3. XGBOOST STOCK PRICE PREDICTION

IV. Experimentation and Results

All experiments were conducted in a Python environment using powerful libraries for machine learning, deep learning, and technical analysis. TensorFlow was used for building and training the LSTM model; scikit-learn for building Linear Regression, data preprocessing tasks, and evaluation metrics; XGBoost for running gradient boosting regression. Besides that TA-Lib utilized domain-specific technical indicators like SMA, EMA, RSI, MACD [6] which helped greatly in fea- ture richness for the XGBoost model. This code was ex- ecuted on a machine having an Intel Core i7 CPU 11th Gen 16 GB RAM NVIDIA RTX 3060 GPU 6GB VRAM

adequate computing capacity to train deep learning models at an optimal pace. Model performance was assessed by analyzing the conventional regression measures of Root Mean Squared Error (RMSE), Mean Squared Error (MSE), and R² Score(Coefficient of Determination). The prediction and fit of the model were very adequately captured by these measures. Each stock's dataset was partitioned into training and testing sets using a conventional 80:20 split. That is to say, 80% of the historical data was used for training while 20% was held for testing. A rolling window approach was adopted in models such as time series models LSTMs to maintain temporal consistency. There was an experiment setup that used fair comparison amongst the models based on best financial time-series forecasting practises. The accuracy of the three predictive models—Linear Regression, LSTM, and XGBoost—was evaluated with three sets of stock data: AAPL, GOOG, and AI. Detailed comparison was done with quantitative metrics as well as graphical analysis. Line graphs were plotted for each set to show the relation between the real and predicted stock prices to determine the accuracy of the prediction through visual inspection. Additionally, an abridged table of significant error measures such as RMSE, MSE, and R² Score was prepared to display a numerical basis for model performance analysis.

The results showed that Linear Regression, though simple and intuitive, performed poorly with non-linearity and tem- poral dependencies, returning lower R² scores (e.g., 0.3343 on AAPL). The LSTM model demonstrated to have good capability to learn time-series trends, especially on stable and large data like AAPL and GOOG, with the R² measures as high as 0.8910 and 0.8734, respectively. Nonetheless, it did relatively lower on the more volatile AI dataset. The XGBoost model was a strong contender and performed well across all datasets, particularly GOOG ($R^2 = 0.9813$) and AI ($R^2 = 0.8325$), due to the incorporation of technical indicators like SMA, EMA, RSI, and MACD.

Utilization of technical indicators clearly enhanced the predictability of the XGBoost model, most notably in terms of momentum detection and trend reversal. While LSTM learned better long-term trends, it was more prone to overfitting with small datasets without proper regularization. In contrast, XGBoost demonstrated better generalizability, which might be due to the ensemble architecture and strong regularization used. Overall, the results show that there is no one-size-fits-all model; rather, performance varies based on the data profile, with XGBoost [4] being superior on feature-rich structured data and LSTM more appropriate for smoother, temporally consistent datasets.

V. Conclusion

In our research, we investigated and contrasted the perfor- mance of three predictive models—Linear Regression, LSTM, and XGBoost—in stock price forecasting based on past data. Through our research, we identify that although Linear Re- gression offers a straightforward and understandable baseline model, it is incapable of being able to detect non-linear and sequential patterns in stock market data. Contrariwise, LSTM and XGBoost greatly exceeded Linear Regression's performance on every dataset, and while XGBoost made use of engineered technical features, LSTM effectively captured temporal dynamics.

What this indicates most of all is that models capable of including time-series dynamics (LSTM) or feature-fectched within domains (XGBoost) produce dramatically greater levels of prediction accuracy. The lesson learned from this is both the need for temporal modeling as well as richer features in terms of stock price prediction. But our method has its limitations. The models make use of only technical indicators and do not factor in external data like real-time financial news, economic announcements, or sentiment, which can affect stock prices too. Future efforts can incorporate these elements to make predictions even stronger.

Although this study has achieved encouraging results by us- ing LSTM and XGBoost models, there are significant avenues of further improvement. One major extension would be to integrate real-time financial news and social media sentiment analysis by employing Natural Language Processing (NLP) methods because such extraneous factors tend to influence short-term market movements. Moreover, the use of more sophisticated deep learning models like Transformer-based time series models (e.g., Temporal Fusion Transformers or Informers) might be able to learn intricate dependencies over longer periods.

VI Future Prospect

Subsequent research may also widen the scope by extending to heterogeneous asset classes like cryptocurrencies, ETFs, and commodities, thus putting the model through its paces in a variety of financial landscapes. Lastly, creating hybrid ensemble models that synergize the respective strengths of LSTM (sequential learning) and XGBoost (feature-based deci- sion making) might yield better generalization and precision. Such optimizations can have a substantial impact on the prac- tical applicability of AI-based models to real-world financial forecasting.

References

[1] O. B. Sezer, M. U. Gudelek, and A. M. Ozbayoglu, "Financial time series forecasting with deep learning: A systematic literature review: 2005–2019," *Applied Soft Computing*, vol. 90, p. 106181, 2020.

[2] J. Patel, S. Shah, P. Thakkar, and K. Kotecha, "Predicting stock market index using fusion of machine learning techniques," *Expert Systems with Applications*, vol. 42, no. 4, pp. 2162–2172, 2015.

[3] T. Fischer and C. Krauss, "Deep learning with long short-term memory networks for financial market predictions," *European Journal of Operational Research*, vol. 270, no. 2, pp. 654–669, 2018.

[4] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794, 2016.

[5] P. Chandarana and D. Badelia, "A survey on stock market prediction: using machine learning and deep learning techniques," *International Journal of Computer Applications*, vol. 182, no. 39, pp. 19–27, 2021.

[6] E. Chong, C. Liang, and H. H. Goh, "Deep learning networks for stock market analysis and prediction: Methodology, data representations, and case studies," *Expert Systems with Applications*, vol. 83, pp. 187–205, 2017.

[7] S. Selvin, R. Vinayakumar, E. A. Gopalakrishnan, A. M. Soman, and K. P. Soman, "Stock price prediction using lstm, rnn and cnn-sliding window model," *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pp. 1643–1647, 2017.

[8] R. G. Sawhney, M. S. Akhtar, and T. Chakraborty, "Stock selection via spatiotemporal transformer networks," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 5, pp. 4464–4471, 2021.