Predicting S&P 500 Stock Prices Using XGBoost, LSTM and Lasso Regression: A Machine Learning Approach to Financial Time Series Forecasting

Daniel Harapiak
Computer Science
University of Western Ontario
London, ON, CA
dharapia@uwo.ca

Leyang Xing
Software Engineering
University of Western Ontario
London, ON, CA
lxing24@uwo.ca

Kai On Ng
Software Engineering
University of Western Ontario
London, ON, CA
kng368@uwo.ca

Paul Gherghel
Software Engineering
University of Western Ontario
London, ON, CA
pgherghe@uwo.ca

***Abstract*— Many individual investors lack the financial expertise needed to make informed stock market predictions. Our algorithm leverages machine learning models, including XGBoost, LSTM, and Lasso, to analyze historical stock data and generate time series forecasting predictions. To enhance interpretability, histogram and line graphs are used to visualize market volatility and stock trends. Given the complexity of the market, this tool serves as a valuable resource for investors seeking to make data-driven decisions.

I. Introduction

The stock market represents one of the most dynamic and complex financial systems, where prices fluctuate based on a multitude if factors, including economic indicators, geopolitical events, and investor sentiment. For individual investors, navigating this volatility without advance tools or expertise often leads to a suboptimal decision and significant financial losses. Traditional methods of stock price prediction, such as technical analysis or fundamental analysis, rely heavily on human interpretation and are prone to biases. In contrast, machine learning offers a data-driven approach capable of identifying intricate patterns in historical data, providing more accurate and objective predictions.

Recent advancements in machine learning have demonstrated its potential in financial time series forecasting. Models like XGBoost, Long Short-Term Memory (LSTM) networks, and Lasso Regression have shown promise in capturing both linear and non-linear relationships in stock price data. XGBoost, a gradient-boosting algorithm, excels in handling complex feature interactions and missing data. LSTM networks, a type of recurrent neural network, are particularly adept at modeling temporal dependencies in sequential data. Lasso Regression, with its L1 regularization, offers interpretability by selecting the most relevant features while maintaining predictive accuracy. Despite their strengths, each model has limitations—linear models like Lasso struggle with non-linear trends, while LSTM networks require extensive computational resources and hyperparameter tuning.

This study aims to evaluate and compare the performance of these three machine learning models—XGBoost, LSTM, and Lasso Regression—in predicting the S&P 500 stock prices. The S&P 500, a benchmark index representing the performance of 500 large-cap U.S. companies, serves as an ideal dataset due to its liquidity, historical data availability, and relevance to investors. By leveraging historical price data and technical indicators such as the Relative Strength Index (RSI) and Moving Average Convergence Divergence (MACD), we seek to develop a robust forecasting tool that can assist investors in making informed decisions. Additionally, we explore the impact of feature engineering, normalization, and hyperparameter tuning on model performance.

The broader implications of this work extend beyond individual investors. Financial institutions, portfolio managers, and algorithmic traders can benefit from accurate short-term price predictions to optimize trading strategies and mitigate risks. Furthermore, this research contributes to the growing body of literature on machine learning applications in finance, highlighting practical challenges and opportunities for future improvements.

II. Background

Investing in the stock market has always been fraught with uncertainty, particularly for individual investors who lack access to sophisticated analytical tools or expert advice. The 2008 financial crisis serves as a stark reminder of the risks associated with market volatility, where the S&P 500 lost nearly 50% of its value in just 18 months, erasing trillions of dollars in wealth [1]. Such events underscore the need for reliable forecasting tools that can help investors anticipate market movements and make data-driven decisions.

Traditional approaches to stock price prediction fall into two broad categories: technical analysis and fundamental analysis. Technical analysis relies on historical price and volume data to identify patterns and trends, often using indicators like moving averages or Bollinger Bands. While intuitive, this method is limited by its reliance on subjective interpretations and its inability to account for external factors such as economic policies or corporate earnings reports. Fundamental analysis, on the other hand, evaluates a company's intrinsic value by examining financial statements, industry conditions, and macroeconomic factors. Although comprehensive, this approach is time-consuming and may not capture short-term market dynamics [2].

Machine learning has emerged as a powerful alternative, capable of processing vast amounts of data and uncovering hidden patterns that elude traditional methods. For instance, LSTM networks have been successfully applied to predict stock prices by modeling sequential dependencies in time-series data [3]. Similarly, ensemble methods like XGBoost have demonstrated superior performance in financial forecasting due to their ability to handle non-linear relationships and feature interactions [4]. Lasso Regression, while simpler, provides a baseline for comparison and offers interpretability, which is crucial for understanding model decisions in financial contexts.

Despite these advancements, challenges remain. Financial data is inherently noisy and non-stationary, meaning statistical properties like mean and variance change over time. This poses difficulties for models trained on historical data, as past patterns may not always predict future behavior. Additionally, the "black-box" nature of some machine learning models, particularly deep learning architectures, can hinder trust and adoption among investors who require transparency in decision-making [5].

Our work addresses these challenges by rigorously evaluating three distinct machine learning models on the S&P 500 dataset. We focus on short-term forecasting (one-day-ahead predictions) to provide actionable insights for active traders. By incorporating technical indicators, lag features, and advanced normalization techniques, we aim to enhance model accuracy and robustness. Furthermore, we emphasize the importance of model interpretability and computational efficiency, ensuring our solutions are both practical and scalable.

III. Data Pre-Processing

Data preprocessing was an essential step to ensure the models were trained on clean and relevant features, improving prediction accuracy and reducing noise. The following techniques were applied:

A. Feature Construction

- Technical Indicators: Computed features like the Relative Strength Index (RSI) and Moving Average Convergence Divergence (MACD) to encapsulate key market trends and momentum indicators.
- Lag Features: Incorporated previous stock prices and trading volumes to introduce temporal dependencies, particularly aiding the Lasso Linear Regression and XGBoost models in understanding historical patterns.
- Time-Lagged Sequence Variables: Specifically structured time-lagged data for sequential modeling in the LSTM network, enabling it to capture complex temporal relationships inherent in stock price movements.
- Nonlinear Features: Generated advanced features such as polynomial transformations, logarithmic values, and interactions between variables to enhance the predictive capabilities of all three models by capturing intricate patterns and relationships.

B. Normalization

Min-Max Normalization was employed to scale all predictors to a consistent range of [0,1]. This technique preserved the original data points' patterns while aligning well with the project's requirements. By ensuring consistent ranges across all predictors, the normalization process facilitated faster convergence during model training and improved numerical stability.

C. Data Splitting

To preserve the temporal structure of the dataset, a time-series split strategy was employed, ensuring that the training and testing phases adhered to the chronological order of the data. This approach mitigated the risk of data leakage and maintained the integrity of the predictive analysis. The dataset was partitioned with 70% allocated for training and validation and the remaining 30% reserved exclusively for testing.

IV. Modeling and Analysis

A. Lasso Regression

1) Objective

The objective of this study was to evaluate the performance of Lasso Regression, a linear model with L1 regularization, in predicting S&P 500 prices. The focus was on feature sparsity to improve interpretability by identifying and selecting the most relevant predictors while maintaining predictive accuracy.

2) Implementation

The implementation of Lasso Regression consisted of four main stages: feature construction, feature selection, hyperparameter tuning, and evaluation.

a) Feature Construction

To enhance the model's ability to identify complex relationships within the data, nonlinear transformations such as squared, cubed, and logarithmic terms were introduced. Interaction terms were also added to capture interdependencies among key predictors.

b) Feature Selection

A forward stepwise feature selection process was conducted to iteratively evaluate and add predictors that significantly reduced the root mean squared error (RMSE). The improvement in prediction error with each additional feature is illustrated in Figure 1. This process ensured the inclusion of only the most impactful predictors, improving model efficiency and interpretability.

c) Hyperparameter Tuning (Model Selection)

The L1 regularization parameter (α) was optimized using a grid search on the validation dataset. Time-series cross-validation was employed to maintain the chronological structure of the data, ensuring reliable evaluation and preventing data leakage. This process identified the best hyperparameters for the final model, with metrics evaluated for each fold and averaged across all folds.

d) Evaluation Metrics

The final model was evaluated using several metrics, including mean absolute error (MAE), root mean squared error (RMSE), R-squared (R2), directional accuracy, and the percentage of predictions falling within a 0.5% error margin. These metrics provided a comprehensive assessment of the model's performance.

3) Observation

The results of the Lasso Regression analysis revealed both strengths and limitations. The model excelled in identifying significant predictors, producing a sparse and interpretable model without significantly compromising accuracy. Its performance was particularly robust in stable market conditions, highlighting its ability to generalize well under normal fluctuations.

However, the model struggled during periods of sharp price changes, a limitation attributed to its linear nature. This constraint reduced its ability to adapt to non-linear or volatile patterns in the data. While the predictive accuracy was high, the directional accuracy demonstrated moderate performance, indicating challenges in anticipating market turning points.

4) Results

The performance metrics of the Lasso Regression model are summarized as follows:

- MAE: 14.25
- RMSE: 22.06
- R^2: 1.00
- Accuracy (0.5% Error): 78.44%
- Directional Accuracy: 81.91%

The forward stepwise feature selection process identified a set of significant features, including High, Low, Adj Close_RSI_interaction, RSI_Lag_1, RSI_log, and Open_Lag_15. These features significantly contributed to the model's predictive performance, as evidenced by the incremental reduction in error shown in Figure 1.

5) Lesson Learned

The study demonstrated that Lasso Regression is highly effective for feature selection and provides an interpretable model for predicting stock prices in stable market conditions. However, the model's linear nature limited its adaptability to periods of rapid price changes. To address these limitations, more flexible non-linear models, such as ensemble methods or neural networks, could be considered for future studies.

B. LSTM Network

1) Objective

The purpose of this analysis was to evaluate the effectiveness of a Long Short-Term Memory (LSTM) model, a type of deep learning model, in predicting S&P 500 prices. The architecture of the LSTM model is specifically designed to capture temporal dependencies and complex sequential patterns inherent in financial time-series data.

2) Implementation

a) Model Architecture

The LSTM model consists of a single layer with 100 hidden units, designed to capture long-term dependencies and temporal patterns in the data. To reduce overfitting and enhance generalization, a dropout rate of 40% was applied during training. The model outputs a single value for next-day price prediction through a fully connected layer.

b) Feature Construction

The same feature construction strategy used for other models was applied to ensure consistency in the dataset used by the LSTM model.

c) Training and Validation

To create input data, a sequence length of 30 was used, allowing the model to observe 30 days of historical data to predict the next day's price. The dataset was split into training (70%), validation (15%), and test (15%) sets, maintaining the chronological structure to prevent data leakage. The model was trained using the Adam optimizer with a learning rate of 0.0001, minimizing the Mean Squared Error (MSE) loss.

3) Observation

The LSTM model demonstrated strong predictive performance in capturing general trends, reflected in a high R2R^2R2 score of 0.95. It performed well in predicting prices during stable and moderately volatile periods, showcasing its ability to generalize from historical trends.

However, the accuracy within a 0.5% error margin (14.18%) indicates that, while the model captures general trends, it struggles with precision in predicting exact price values. Furthermore, a directional accuracy of 50.98% suggests that the model's ability to predict the correct trend direction is close to random, highlighting limitations in trend sensitivity, especially during periods of high volatility.

4) Model Improvement

If computational resources permit, the performance of the LSTM model could be enhanced through two primary strategies. First, a comprehensive grid search could be conducted over key hyperparameters, such as the learning rate, number of hidden units, sequence length, and batch size, to identify the optimal configuration for the model. Second, incorporating an efficient feature selection process would help pinpoint the most impactful predictors, thereby reducing noise and improving the model's accuracy and generalization capabilities. These approaches collectively aim to refine the model's architecture and input data quality, ensuring better performance and robustness.

5) Results

- MAE: 101.85
- RMSE: 129.57
- R^2: 0.95
- Accuracy (0.5% Error): 14.18%
- Directional Accuracy: 50.98%

6) Lesson Learned

The performance of the LSTM model is highly sensitive to the choice of hyperparameters. Proper tuning of parameters such as the learning rate, sequence length, and hidden units could significantly improve the model's precision and trend prediction capabilities.

Although forward stepwise selection was employed to identify the most impactful features, its computational cost proved prohibitive when dealing with many features. While it is more efficient than subset selection, alternatives such as mutual information or automated feature ranking could streamline the feature selection process and potentially improve the model's performance by reducing irrelevant inputs.

C. XGBoost

1) Objective

We thoroughly evaluated the XGBoost model's performance in predicting the S&P 500's future Adjusted Closing Price. XGBoost, a robust gradient-boosting algorithm, is particularly well-suited for this application because it can effectively learn complex, non-linear relationships and handle missing or incomplete data. By capitalizing on these strengths, the model can uncover intricate patterns within the stock market, ultimately improving the accuracy and reliability of its predictions.

2) Implementation

We implemented the XGBoost algorithm, primarily focusing on extensive hyperparameter tuning to optimize its performance in modelling complex financial time series data. Specifically, we carefully adjusted critical parameters such as the learning rate, maximum tree depth, number of estimators, and regularization lambda to ensure the model could accurately capture the intricate, non-linear relationships inherent in the S&P 500 dataset. Our initial approach involved training the model using absolute data, which allowed us to assess its capacity to interpret untransformed features and identify broad market patterns. This baseline configuration served as an informative starting point,

allowing us to evaluate the feasibility of using raw inputs before refining our approach to produce more nuanced and dynamic predictions.

3) Observation

XGBoost achieved superior predictive accuracy compared to the Linear Lasso model, mainly due to its capacity to capture complex feature interactions. This strength underscores XGBoost's ability to more effectively model non-linear relationships within the financial dataset. However, this improved predictive power came with notable computational demands, as the model required substantial computational resources to reach optimal performance levels. Additionally, the model exhibited underfitting when appropriate regularization techniques were not applied, causing its predictions to cluster around the mean or median of the Adjusted Closing Price. This behaviour led to a flattened output (Figure 6), reinforcing the necessity of proper regularization strategies to achieve more dynamic and representative predictions.

4) Model Improvement

We shifted our approach to training the model with relative data to address the limitations encountered when using absolute data [9]. This refinement involved calculating the percentage change of each feature relative to the previous day, effectively normalizing the dataset and emphasizing underlying trends over raw values. By directing the model to predict the next day's percentage change instead of the absolute value, we enabled it to more closely track the dynamic patterns of the S&P 500 (Figure 5). This methodological adjustment substantially enhanced the model's accuracy and responsiveness, instilling greater confidence in its performance and enabling it to capture daily market fluctuations more effectively.

5) Results

a) Performance Matrics

- MAE: 0.0002
- RMSE: 0.0018
- R^2: 0.9723
- Accuracy (0.5% Error): 57.79% (Figure 6)
- Directional Accuracy: 99.9243% (Figure 5)

6) Lesson Learned

Feature transformation emerged as a key factor in improving the performance of our XGBoost models. By employing relative data and computing percentage changes, we effectively reduced the data range to a bounded interval (e.g., between -1 and 1), thereby enhancing the model's ability to capture underlying trends and deliver more dynamic predictions. Additionally, extensive hyperparameter tuning proved critical for optimizing model accuracy and preventing both underfitting and overfitting. For example, increasing the number of estimators (n_estimators) improved predictive accuracy, while adjusting the feature subsampling parameter (colsample_bytree) introduced beneficial randomness and reduced the risk of overfitting. Together, these methodological refinements underscore the importance of data transformation and systematic parameter optimization in producing robust and reliable predictive models.

V. Visualization

We employed line graphs and scatter plots to comprehensively understand the model's predictive performance. The line graphs, depicting actual and predicted Adjusted Closing Prices, enabled us to visually assess the model's accuracy on the test set and its ability to learn from historical data (Figures 2, 3, and 5). Additionally, the scatter plots illustrated the proportion of predictions falling within a 0.5% accuracy threshold, thereby highlighting the model's precision in closely tracking actual price movements (Figures 2 and 6). These visualizations offer valuable insights into the model's strengths and limitations and inspire greater confidence in its predictive capabilities.

VI. Future work

One potential improvement to our XGBoost model is the integration of feature selection mechanisms. Currently, the model's predictions exhibit some bias, likely due to the inclusion of redundant or irrelevant features. Introducing a feature selection step could help refine the input data by isolating the most impactful predictors. Techniques such as forward stepwise selection, mutual information ranking, or Shapley value analysis could be employed to systematically identify and eliminate unnecessary features. By reducing noise and enhancing the signal, this approach would likely improve model accuracy, efficiency, and generalizability. As noted in [10] effective feature selection is critical for achieving robust predictions in financial modeling scenarios, especially in volatile market conditions.

Achieving a meaningful edge over the market often requires more than just traditional market data. Future iterations of our research could expand the dataset to include auxiliary data sources, such as sentiment analysis from social media platforms, financial news articles, and macroeconomic indicators. The CalixBoost [10] model demonstrated how incorporating sentiment analysis and macroeconomic metrics could significantly enhance prediction accuracy by providing a more holistic view of market conditions. Leveraging these additional data sources in conjunction with XGBoost or other machine learning models may enable us to capture the nuanced drivers of market behavior, especially during periods of rapid change or economic downturns.

The next steps would be taking our research results and creating a simplified visualization interface for individual investors to gain financial insight from these models. The goal being taking the result values and converting them into visual metrics a person with little to no data science background could understand.

VII. Conclusion

To conclude, we chose to mainly focus on the Boosting model, however due to complications with the predicted results we wanted to explore how other models would compare. Choosing a simpler linear lasso model and the more complex LSTM model we were able to see and compare how our predictions differed among the models. To conclude we worked hard on getting XGBoosting to provide an accurate result while also exploring different alternative models to see how they compared.

VIII. Reference

[1] Mishkin, F. S. (2011). "Over the Cliff: From the Subprime to the Global Financial Crisis." Journal of Economic Perspectives, 25(1), 49-70.

[2] Fama, E. F. (1970). "Efficient Capital Markets: A Review of Theory and Empirical Work." The Journal of Finance, 25(2), 383-417.

[3] Hochreiter, S., & Schmidhuber, J. (1997). "Long Short-Term Memory." Neural Computation, 9(8), 1735-1780.

[4] Chen, T., & Guestrin, C. (2016). "XGBoost: A Scalable Tree Boosting System." Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 785-794.

[5] Guidotti, R., et al. (2018). "A Survey of Methods for Explaining Black Box Models." ACM Computing Surveys, 51(5), 1-42.

[6] "Yahoo Finance - S&P 500 Data," yfinance Python package, [Online]. Available: https://github.com/ranaroussi/yfinance. [Accessed: December 4th,2024].

[7] "XGBoost regressor gives a flat prediction," Stack Overflow. [Online]. Available: https://stackoverflow.com/questions/77215606/xgboost-regressor-gives-a-flat-prediction. [Accessed: Dec. 5, 2024].

[8] J. Yeo and Chai Kiat Yeo, "CalixBoost: A Stock Market Index Predictor using Gradient Boosting Machines Ensemble," Jun. 2022, doi: https://doi.org/10.5121/csit.2022.121009.

FIGURES

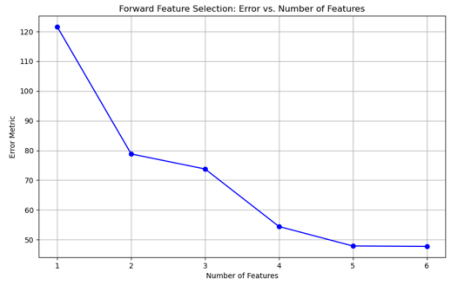

Figure 1: Line graph of the performance improvement of doing forward feature selection on Lasso

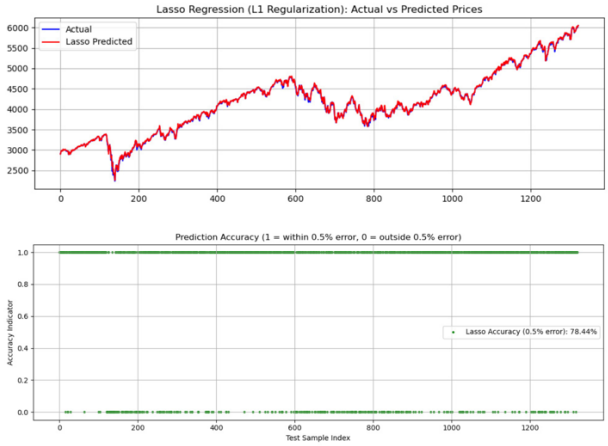

Figure2: Line graph of the actual and model prediction of the Adjusted Closed Price for S&P 500 by using Lasso

Figure3: Line graph of the actual and model prediction of the Adjusted Closed Price for S&P 500 by using LSTM

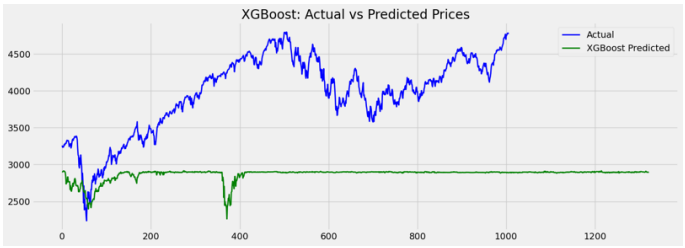

Figure 4: XGBoost Training result using absolute data

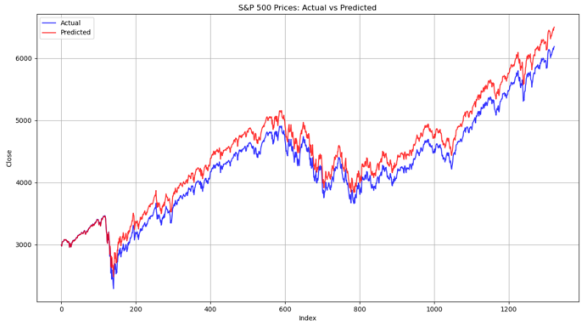

Figure 5: Line graph of the actual and XGBoost model prediction of the Adjusted Closed Price for S&P 500

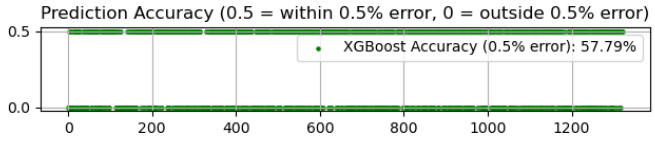

Figure 6: Scatterplot displays the 0.5% accuracy of the XGBoost prediction

Appendix

Leyang Xing	Kai On Ng	Daniel Harapiak	Paul Gherghel
Data preparation from yfinance	Setup project environment	Researching Background	Researching XGBoosting
Advanced indicators computation (coding)	Research on XGBoost implementation	Writing Background, draft and final report	Researching on time series forecasting

Set up three models' structures and did initial developing	Setup XGBoost initial structure	Writing Abstract final	Writing Future Works final
Keep developing Lasso model	Construct walk-forward validation on XGBoost	Researching XGboost model, Lasso model, LSTM model	Writing Conclusion final
Keep developing LSTM model	Research on potential problems affecting XGBoost training process	Troubleshooting XGboost	Proposal Researching Plan for Analysis
Writing data pre-processing in brainstorm, draft and final report	Improve the XGBoost model training by using relative data	Researched Stock Market Crashes and Problems	Brainstorming Plan for Analysis
Writing Lasso model analysis in final report	Write XGBoost model analysis in the final report	Assessing Market Relevance	Researching Procedure for all 3 models
Writing LSTM model analysis in final report	Write Visualization analysis in the final report	Exploring Machine Learning Applications in Finance	Video Editing for Brainstorming video
Research on accessibility and correctness of models and propose improvements possibilities	Research on multiple possible visualization methods for this project	Proposal Description of Applied Problem	Video Editing for Presentation video
		Brainstorming Applied Problem	