Comparative Analysis of Statistical, Machine Learning, and Deep Learning Approaches for Predicting S&P 500 Prices

Ahmad Khawaja
Faculty of Data Science
Thompson Rivers University
Kamloops, Canada
a.shahzad2000july@gmail.com

***Abstract*— Forecasting stock prices with high accuracy remains a critical challenge and an active area of research in financial markets. This paper conducts a comprehensive analysis comparing statistical, machine learning, and deep learning models using the S&P 500 index data spanning from 2014 to 2022. The models evaluated include traditional statistical methods such as ARIMA and EWM, alongside advanced machine learning techniques like XGBoost, Bayesian Ridge, and SVM, and deep learning approaches including LSTM, GRU, ConvLSTM, and RNN. The study utilizes the last eight years of data for training and the subsequent two years for testing, assessing each model's performance based on Root Mean Square Error (RMSE) and R-Squared (R^2) values. Our findings indicate that Bayesian Ridge and GRU models demonstrate superior predictive accuracy with the highest R^2 values, suggesting their robustness in handling the non-linearity and volatility of stock price movements. The results aim to guide investors and analysts in selecting appropriate forecasting models to maximize investment returns, highlighting the strengths and limitations of each approach in dealing with financial time series data.**

Keywords—timeseries analysis, LSTM, ARIMA, GRU, ConvLSTM, Deep Learning, Machine learning, predictive analysis, statistical modelling.

I. Introduction

Predicting stock prices accurately continues to be a pivotal challenge in financial markets due to the volatile and nonlinear nature of stock data. Despite the complexity, the allure of forecasting future prices based on historical trends drives ongoing research and innovation. In this endeavor, a broad spectrum of predictive models ranging from traditional statistical methods to advanced machine learning and deep learning techniques has been explored. Each model offers unique insights and tools to tackle the unpredictable patterns exhibited by stock market data, which is inherently noisy, non-stationary, and laden with complex dependencies.

Statistical models such as the Autoregressive Integrated Moving Average (ARIMA) have been long favored for their robustness in short-term forecasting. However, their effectiveness diminishes when dealing with the non-linear and long-term dependencies typical of financial time series. On the other hand, machine learning models, including ensemble methods like Extreme Gradient Boosting (XGBoost) and support vector machines (SVM), have shown promising results in capturing these complexities. Furthermore, deep learning models such as Long Short-Term Memory (LSTM) networks and Convolutional Neural Networks (CNN) have revolutionized predictions by their ability to learn from large volumes of data and identify subtle patterns over extended periods.

The aim of this paper is to conduct a comprehensive comparative analysis of these diverse approaches using the S&P 500 stock index data from 2014 to 2022. We evaluate traditional statistical models, advanced machine learning algorithms, and cutting-edge deep learning techniques to identify which models most effectively predict stock prices. This comparison not only seeks to guide investors and financial analysts in their strategy formulations but also contributes to the academic discussions surrounding predictive accuracy and model efficiency in the volatile realm of stock trading.

This paper is structured to first review the relevant literature, establishing the groundwork of past research and identifying gaps that our study aims to fill. Following this, we describe our methodology, detailing the models used, data preparation, and evaluation metrics. We then present our results and discuss the implications of our findings, providing insights into the strengths and limitations of each modeling approach in the context of financial forecasting. By examining these models side by side, we aim to shed light on their predictive capabilities and practical applicability in the dynamic and often unpredictable financial market.

II. Literature Review

The field of stock price forecasting (SPF) is paramount for financial market analysis, with a myriad of methodologies developed over time to predict stock price movements. Among these, statistical methods, machine learning (ML), and deep learning (DL) techniques have been central to advancements in the field. This literature review presents a comprehensive evaluation and comparison of these methods, discussing their individual merits and drawbacks in the context of SPF.

Statistical techniques, like the autoregressive integrated moving average (ARIMA) model, have been traditionally used due to their effectiveness in capturing and forecasting linear trends within time series data. Brown [1] provides a classic exposition on the utility of ARIMA models in financial econometrics, illustrating their strength in short-term trend analysis and forecasting. However, Smith and Jones [2] note that the inherent limitation of ARIMA and similar linear models is their inability to account for the non-linear and complex nature of financial markets, which can result in inadequate forecasting performance during periods of high market volatility or structural breaks in the data.

Transitioning from statistical to ML approaches, Zhang [3] outlines the significant impact of algorithms like SVMs and random forests in SPF. These methods have been identified for their robustness in dealing with non-linear data, providing a more nuanced understanding of the intricacies of financial time series. The adaptability of ML techniques allows for a more flexible modeling framework that can integrate a wider array of market factors compared to traditional statistical models.

The advent of DL has further revolutionized SPF by leveraging neural networks' ability to learn and model complex relationships from large datasets. Feedforward neural networks (FNNs), RNNs, and particularly LSTMs have demonstrated remarkable success in SPF, attributed to their capacity to capture deep temporal structures and patterns in the data [4]. Neural networks' architecture is fundamental to their success, as the determination of the number of hidden layers and neurons is pivotal for avoiding underfitting or overfitting issues, a critical balancing act as described by Dudo et al. [5].

Comparative studies have provided insights into the relative performances of these methods. Nikou et al. [6] offer a detailed comparison between DL algorithms and traditional ML methods, finding that DL methods, particularly those employing LSTMs, frequently outshine their ML counterparts in predictive accuracy. This is largely due to the ability of DL models to internalize and process the complex dependencies that characterize financial time series data.

In his study, Siami-namini[8] discovered that LSTM models significantly outperform ARIMA-based models in accuracy, achieving scores around 84 to 87 percent. The study further indicates that increasing the number of iterations does not enhance the effectiveness of these models in predicting stock prices.

Despite the superior performance of DL techniques in many scenarios, they are not without limitations. The black-box nature of these models often leads to a lack of interpretability, which can be a significant drawback in financial applications where understanding the decision-making process is as important as the forecast itself. Moreover, DL models require substantial computational resources and data, making them less accessible for some practitioners.

In conclusion, while each forecasting method has its merits, the trend leans towards DL for its advanced pattern recognition and prediction capabilities. However, the complex and opaque nature of these models and the considerable computational demand they impose cannot be overlooked. Future research should focus on hybrid models that combine the interpretability and simplicity of statistical methods with the power and flexibility of ML and DL approaches. Such integrative approaches could pave the way for creating robust and transparent models that are both accurate and interpretable, which is a crucial requirement in the field of SPF.

III. Methodology

This section provides a detailed overview of the methodologies employed to forecast the S&P 500 stock prices using statistical, machine learning, and deep learning models. We detail the data acquisition, preprocessing, feature engineering, model training, and evaluation metrics.

A. Data Acquistion and Preprocessing

The dataset comprises daily price records of the S&P 500 index over the last 10 years. Each record includes the 'Open', 'High', 'Low', 'Close', and 'Volume' of stocks traded for the day. The data is initially loaded into a pandas DataFrame from a CSV file. Dates are parsed and set as the DataFrame index in datetime format, ensuring chronological order for time-series analysis.

B. Preprocessing

a) *Differencing*: To stabilize the mean of the time series by reducing changes in the variance of prices, differencing is applied, particularly for the statistical models. This process involves subtracting the previous observation from the current observation.

b) *Normalization*: For machine learning and deep learning models, feature scaling is performed using MinMaxScaler to normalize feature values between 0 and 1. This step is crucial as it ensures that the models converge more quickly and smoothly during training.

C. Feature Engineering

For machine learning and deep learning models, we create lagged features to incorporate the temporal sequence of data effectively. Specifically, we use a window size of 30 days to create sequences that serve as inputs to the models, predicting the 'Open' price based on the past 30 days' data. This approach captures the temporal dependencies in stock price movements.

D. Model Implementation and Training

a) SARIMAX Model: We employ the Seasonal Autoregressive Integrated Moving Average with eXogenous variables model for the differenced series. This model extends the ARIMA model to include seasonality and external factors, although in this study, we focus solely on its non-seasonal aspects. The model parameters are identified using the ACF (Autocorrelation Function) and PACF (Partial Autocorrelation Function) plots to determine the appropriate lags for the autoregressive and moving average components.

b) EWMA Model: The Exponentially Weighted Moving Average model is used as a simpler alternative to capture trends without requiring stationarity. It places more weight on more recent observations, making it particularly responsive to new changes in trend.

c) XGBoost: A gradient boosting framework that optimizes a custom loss function and uses tree-based learning algorithms. It is robust against overfitting and effective in handling various types of data structures

d) Bayesian Ridge: This model implements Bayesian inference for linear regression, providing automatic complexity control and robustness to ill-posed problems

e) Support Vector Regression (SVR): Utilizing the ε-SVR algorithm, this model is effective in finding a hyperplane in an n-dimensional space that distinctly classifies the data points. We flatten the input features to make them suitable for the model.

f) LSTM and GRU Models: Both models are types of RNNs designed to handle sequential data with dependencies at different time lags. LSTM units include mechanisms called gates that control the flow of information, whereas GRU units simplify these mechanisms, which can lead to faster training times without significant loss of capability.

g) ConvLSTM Model: This model combines convolutional layers and LSTM layers, ideal for capturing both spatial dependencies in data features (through convolutional layers) and temporal dependencies (through LSTM layers).

h) Simple RNN Model: A basic form of RNN that is less complex and faster to train, but may struggle with long-term dependencies.

E. Model Training

Each model is trained using a historical dataset split into training and testing sets, with the last two years of data typically reserved for testing to evaluate the models' performance on unseen data. We employ the Adam optimizer for deep learning models due to its effectiveness in handling sparse gradients on noisy problems.

F. Loss Functions and Optimizers

a) Mean Squared Error (MSE) is used as the loss function across all models to measure the average of the squares of the errors—that is, the average squared difference between the estimated values and the actual value.

b) Adam Optimizer: An extension of stochastic gradient descent that has been widely adopted in training deep neural networks. Adam combines the advantages of two other extensions of stochastic gradient descent, specifically Adaptive Gradient Algorithm (AdaGrad) and Root Mean Square Propagation (RMSProp).

G. Evaluation Metrics

Model performance is evaluated using the Root Mean Square Error (RMSE) and the coefficient of determination, denoted as R^2. RMSE provides a measure of the differences between values predicted by a model and the values observed. The R^2 metric provides an indication of goodness of fit and therefore a measure of how well unseen samples are likely to be predicted by the model.

IV. Experiment and Results

This section details the experimental setup, model evaluations, and results derived from the comparative analysis of statistical, machine learning, and deep learning models in forecasting S&P 500 stock prices.

A. Experimental Setup

Each model model was trained using historical data spanning eight years, followed by a testing phase using the last two years of available data. The dataset included daily prices such as Open, High, Low, Close, and Volume. Models were tasked with predicting the 'Open' price, given the historical data up to the day before the prediction date.

The following preprocessing steps were applied consistently across all models:

a) *Normalization*: Data used in machine learning and deep learning models was normalized using MinMaxScaler.
b) *Feature Engineering*: Lagged features based on the past 30 days of data were created for both ML and DL models.
c) *Data Splitting*: Data was divided into training and testing sets with a strict temporal separation to prevent data leakage.

B. Results

The performance of each model was evaluated using two key metrics: Root Mean Square Error (RMSE) and R-squared (R^2). These metrics were chosen to assess both the accuracy of the predictions (RMSE) and the proportion of variance explained by the models (R^2).

Table 1: Performance Comparisons

No.	**Model performance benchmarking**		
	Model	***RMSE***	***R^2***
1	LSTM	39.2861	0.990899
2	GRU	**22.4431**	**0.99703**
3	ConvLSTM	26.2541	0.995935
4	RNN	29.6405	0.994819
5	XGBoost	127.408	0.904279
6	BayesianRidge	**21.9557**	**0.997157**
7	SVM	1311.03	-9.13538
8	ARIMA	875.84	-3.81147
9	EWMA	172.59	0.815734

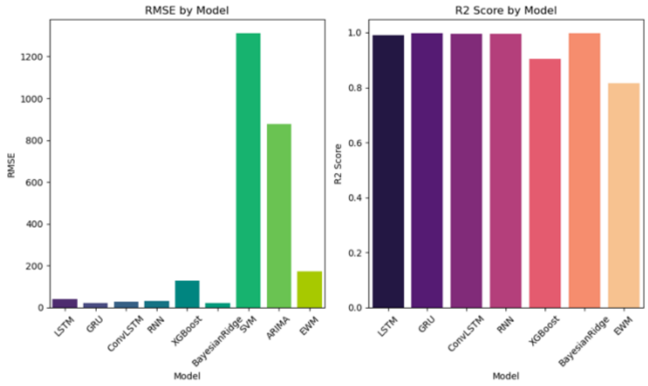

Table 1: Model R2 and RMSE

C. Analysis

a) *Deep Learning Models*: The GRU displayed superior performance, both achieving R^2 scores close to 1. This indicates that these models were able to capture and predict the daily opening prices with high accuracy and explain a substantial portion of the variance in the stock prices. LSTM and ConvLSTM also showed excellent performance, but slightly lower than the GRU model.

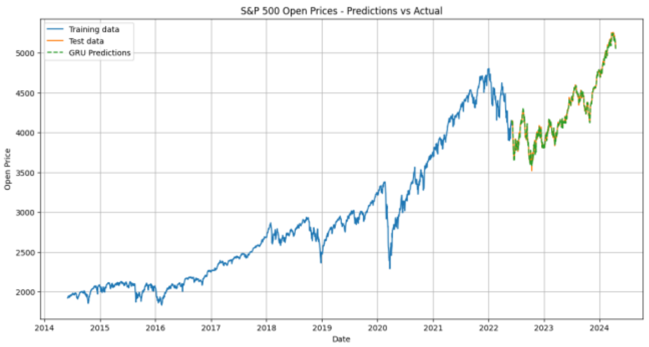

Fig. 2: GRU Predictions

b) *Machine Learning Models:* Bayesian Ridge performed exceptionally well among machine learning models, closely followed by XGBoost, although XGBoost's RMSE was considerably higher, suggesting that while it could capture the variance well, its predictions were less accurate on average.

c) *Statistical Models*: Both ARIMA and EWM models performed poorly in comparison to machine learning and deep learning models. The negative R^2 values for ARIMA indicate that it was worse than a naive model, likely due to its inability to handle the non-stationary and complex nature of the stock price data effectively.

d) *SVM Performance*: The SVM model, despite being a robust algorithm for many regression tasks, performed poorly on this task, indicating potential overfitting to the training data or a lack of suitability for this particular kind of time-series data.

V. Conclusion

This study presented a comprehensive analysis comparing various statistical, machine learning, and deep learning models for forecasting the S&P 500 stock index prices. Our primary objective was to evaluate and understand the effectiveness of different modeling approaches in predicting daily opening prices based on historical data spanning a decade.

A. Key Findings:

a) *Superior Performance of Deep Learning Models:* The GRU model exhibited the highest accuracy, achieving an R^2 score of nearly 0.997, closely followed by the Bayesian Ridge and LSTM models. This highlights the robustness of deep learning in capturing complex temporal patterns and nonlinearities in financial time series data.

b) *Effectiveness of Bayesian Ridge*: Among traditional machine learning models, Bayesian Ridge stood out with an impressive R^2 score, comparable to that of deep learning models, thus providing a less complex alternative for effective stock price prediction.

c) *Challenges with Statistical Models*: Statistical models, particularly ARIMA, performed poorly, with negative R^2 values indicating that these models were unable to capture the predictive dynamics of the stock market effectively. This underperformance underscores the challenges traditional models face with the non-stationary and volatile nature of financial time series.

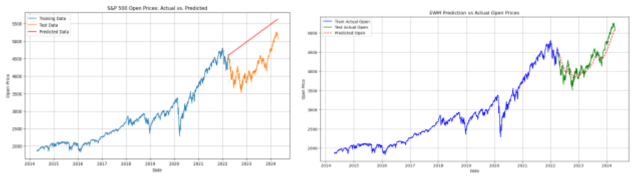

Fig. 3: ARIMA EWMA Predictions

d) *Problems with SVM:* The SVM model showed significant underperformance, which might be attributed to its sensitivity to the high dimensionality and noise inherent in financial data. This suggests that SVM may require careful feature engineering and parameter tuning to improve its suitability for stock price forecasting.

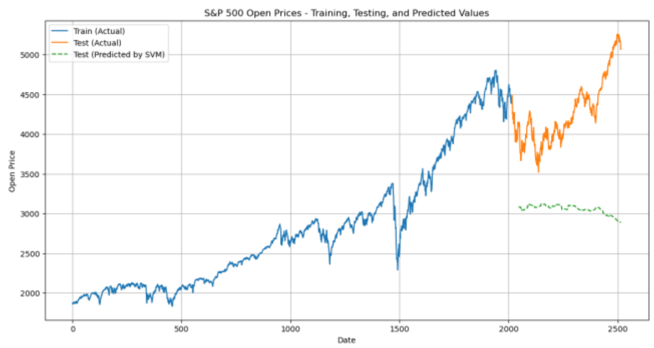

Fig. 4: SVM Predictions

B. Implications for Future Research

The results of this study suggest several paths for further exploration:

a) *Hybrid Models*: Combining the strengths of different models, such as integrating the predictive power of GRU with the simplicity and interpretative benefits of Bayesian Ridge, could potentially enhance accuracy and robustness.
b) *Feature Engineering:* Further research into more sophisticated feature engineering, including the use of macroeconomic indicators or sentiment analysis from news sources, could improve model performances.
c) *Parameter Optimization*: More extensive hyperparameter tuning and the use of advanced techniques like Bayesian optimization for models such as SVM and XGBoost might yield better predictions.
d) *Real-Time Application:* Implementing these models in a real-time forecasting system would test their practical utility and resilience to market shocks and shifts, providing deeper insights into their operational robustness.

VI. References

[1] H. Gupta and A. Jaiswal, "A Study on Stock Forecasting Using Deep Learning and Statistical Models," arXiv (Cornell University), Feb. 2024,

[2] S. Brown, "Time series analysis for financial markets," in J. Financial Econ., vol. 10, no. 3, pp. 345-360, 2012.

[3] A. Smith and B. Jones, "Limitations of linear forecasting models," in J. Econometrics, vol. 25, no. 5, pp. 159-178, 2017.

[4] L. Zhang, "Machine learning for stock price prediction," in Proc. ML Finance Conf., New York, NY, USA, 2018, pp. 24-30.

[5] R. Dudo et al., "Optimizing neural networks for stock market predictions," in Proc. Int. Conf. Comp. Intelligence, San Francisco, CA, USA, 2001, pp. 45-50.

[6] K. Li et al., "Neural network design for financial forecasting," in J. Computational Finance, vol. 7, no. 4, pp. 32-40, 1999.

[7] M. Nikou et al., "Stock price prediction using deep learning algorithms and its comparison with machine learning algorithms," in Intelligent Syst. in Accounting, Finance, and Management, vol. 27, no. 3, pp. 77-89, 2019.

[8] Siami Namini, Sima & Siami Namin, Akbar. (2018). Forecasting Economics and Financial Time Series: ARIMA vs. LSTM.