A Comparative Analysis of Traditional and Machine Learning Methods in Forecasting the Stock Markets of China and the US

Shangshang Jin
Department of Art and Science, Johns Hopkins University, Washington, D.C., United States

***Abstract*—In the volatile and uncertain financial markets of the post-COVID-19 era, our study conducts a comparative analysis of traditional econometric models—specifically, the AutoRegressive Integrated Moving Average (ARIMA) and Holt's Linear Exponential Smoothing (Holt's LES)—against advanced machine learning techniques, including Support Vector Regression (SVR), Long Short-Term Memory (LSTM) networks, and Gated Recurrent Units (GRU). Focused on the daily stock prices of the S&P 500 and SSE Index, the study utilizes a suite of metrics such as R-squared, RMSE, MAPE, and MAE to evaluate the forecasting accuracy of these methodologies. This approach allows us to explore how each model fares in capturing the complex dynamics of stock market movements in major economies like the U.S. and China amidst ongoing market fluctuations instigated by the pandemic. The findings reveal that while traditional models like ARIMA demonstrate strong predictive accuracy over short-term horizons, LSTM networks excel in capturing complex, non-linear patterns in the data, showcasing superior performance over longer forecast horizons. This nuanced comparison highlights the strengths and limitations of each model, with LSTM emerging as the most effective in navigating the unpredictable dynamics of post-pandemic financial markets. Our results offer crucial insights into optimizing forecasting methodologies for stock price predictions, aiding investors, policymakers, and scholars in making informed decisions amidst ongoing market challenges.**

***Keywords*—Machine learning; Holt's LES; SVR; LSTM; GRU**

I. Introduction

The post-COVID-19 era has ushered in an era of heightened volatility and uncertainty in financial markets worldwide [1]. Particularly, the stock markets of China and the United States, two leading global economies, have garnered significant attention from investors, policymakers, and scholars alike. Precise forecasting of stock prices in these markets is crucial for informed decision-making and effective risk management. However, the challenge of accurate stock price prediction remains formidable due to the complex interplay of factors such as economic indicators, market sentiment, geopolitical events, and policy changes.

Our study adopts a two-pronged methodological approach. Initially, we leverage traditional econometric models, specifically the AutoRegressive Integrated Moving Average (ARIMA) model [2] and Holt's Linear Exponential Smoothing (Holt's LES) [3], known for their robustness in time series forecasting. These models, grounded in historical data patterns and statistical principles, offer a foundational understanding of stock price movements, emphasizing the linear aspects of financial time series. However, the intricate dynamics of post-pandemic markets—characterized by abrupt changes and non-linear patterns—necessitate a more adaptive and sophisticated analysis framework. Enter machine learning techniques: Support Vector Regression (SVR) [4], Long Short-Term Memory (LSTM) networks [5], and Gated Recurrent Units (GRU) [6]. These methods bring to the fore the capability to model complex, non-linear relationships and capture deep temporal dependencies, which are often missed by traditional models. By incorporating both traditional and machine learning methodologies, our study aims to harness the complementary strengths of each approach, ensuring a comprehensive and nuanced exploration of forecasting accuracy in the tumultuous environment of post-COVID-19 stock markets. This hybrid approach not only facilitates a direct comparison of predictive performances but also sheds light on the evolving nature of financial time series analysis in response to unprecedented market conditions.

In this study, we conduct a comparative analysis of the forecasting performance of both traditional and machine learning models on the daily stock prices of the S&P 500 Index in the United States and the SSE Index in China in the post-COVID-19 period. We assess the forecasting accuracy of ARIMA, Holt's LES, SVR, LSTM, and GRU models using evaluation metrics such as R-squared (R^2), Root Mean Square Error (RMSE), Mean absolute percentage error (MAPE), and Mean Absolute Error (MAE). By shedding light on the strengths and weaknesses of different forecasting approaches, this study seeks to contribute to the ongoing pursuit of effective stock market prediction tools in the post-COVID-19 era.

II. Related Work

Traditionally, stock price prediction has relied on econometric models and time-series analysis techniques like AutoRegressive Integrated Moving Average (ARIMA) and Linear Exponential Smoothing Model (LSE) . These models excel at capturing linear relationships and seasonality in the data. Nevertheless, their ability to handle the inherent complexities and non-linearities of stock market dynamics is limited [7].

Machine learning and deep learning techniques have emerged as promising alternatives for stock market prediction, offering the potential to capture intricate patterns and relationships in financial data [8], [9]. Methods such as Support

Vector Regression (SVR), Long Short-Term Memory networks (LSTM), and Gated Recurrent Units (GRU) can process large-scale datasets, recognize non-linear relationships, and learn from sequential information, making them well-suited for forecasting stock prices.

For instance, Gülmez [9] explored machine learning models for stock market prediction, underscoring the effectiveness of the LSTM model with two dropout layers. The study noted the potential for performance improvement by optimizing hyperparameters such as the number of neurons, batch size, and epoch count. Gülmez's research also employed the Support Vector Regression (SVR) model, optimizing hyperparameters via grid search with Scikit-learn's library. Employing 10-fold cross-validation and RMSE as a loss measurement, the study highlighted that hyperparameter tuning significantly impacts the SVR's forecasting performance.

Md et al. [10] introduced a novel Multi-Layer Sequential Long Short-Term Memory (MLS LSTM) model for stock price prediction, utilizing Samsung stock data from 2016 to 2021. Comprising three vanilla LSTM layers and a dense layer, the MLS LSTM model exhibited high accuracy (95.9% and 98.1%) and a low average error percentage (2.18%) on the testing dataset. The study revealed that multi-layered LSTMs outperform single-layered LSTMs, with added layers enhancing accuracy.

In another study, Yu et al. [11] proposed a predictive model for stock price index realized volatility (RV) based on optimized variational mode decomposition (VMD), deep learning models, including LSTM and GRU, and the Q-learning algorithm. The model was applied to the RV sequences of the SSEC, SPX, and FTSE indices. The VMD method decomposed the RV sequences into intrinsic mode functions (IMFs), which were then predicted using the LSTM and GRU models. Q-learning determined the optimal model weights for an integrated approach. Performance evaluation using MAE, MSE, HMAE, HMSE, and MDM demonstrated the model's superior performance over comparison models in both emerging and developed markets.

Recent literature shows that machine learning methods, including SVR, LSTM, and GRU, have gained popularity due to their ability to tackle non-linear problems and learn complex patterns in large-scale datasets [12]. These models can capture complex relationships in financial data and enhance prediction accuracy. However, they come with drawbacks such as high computational requirements, risk of overfitting, and reduced interpretability [13], [14]. Additionally, machine learning models often require meticulous hyperparameter tuning, which can be time-consuming and computationally intensive [15].

III. Methodology

In this study, we examine the predictive performance of both traditional statistical methods and machine learning techniques for forecasting the stock price. We compare the ARIMA model and the ETS model from the traditional methods against the SVR, LSTM networks, and GRU networks from the machine learning approaches. Our analysis focuses on one-step-ahead out-of-sample forecasting.

A. AutoRegressive Integrated Moving Average (ARIM)

The ARIMA model, commonly recognized as the Box-Jenkins model, is a fundamental tool in time-series forecasting. It merges autoregressive (AR) and moving average (MA) components to effectively model stationary time series with minimal parameters. In contrast to pure AR and MA models, the ARIMA model introduces a differencing (I) component to ensure stationarity of the series [2]. By blending these three components—autoregressive, differencing, and moving average—the ARIMA model offers a holistic approach to capturing temporal dependencies in data [16].

It is expressed as follows:

$$(1-\sum_{i=1}^{p}\phi_i L^i)(1-L)^d X_t = (1+\sum_{i=1}^{q}\theta_i L^i)\epsilon_t \quad (1)$$

Here, ϕ_i denotes the AR parameters, θ_i the MA parameters, d is the order of differencing, L is the lag operator, X_t is the time series value at time t, and ϵ_t signifies the white noise error term.

B. Holt's Linear Exponential Smoothing Model (Holt's LES)

Holt's Linear Exponential Smoothing model, also known as the Holt's Linear model, is a time-series forecasting method that captures the linear trend and level components in the data. It is particularly useful for datasets with trends but no seasonal patterns. The model uses two smoothing equations to estimate the level and trend components, respectively [17].

Let y_t be the observed value at time t, l_t be the estimated level at time t, and b_t be the estimated trend at time t. The smoothing equations are given by:

$$\begin{aligned} l_t &= \alpha y_t + (1-\alpha)(l_{t-1}+b_{t-1}) \\ b_t &= \beta(l_t - l_{t-1}) + (1-\beta)b_{t-1} \end{aligned} \quad (2)$$

Here, y_t is the observed value, l_t the estimated level, and b_t the estimated trend at time t.α and β are smoothing parameters between 0 and 1. The h-period ahead forecast is:

$$\hat{y}_{t+h} = l_t + h \cdot b_t \quad (3)$$

In this study, the optimal values of α and β a re determined by minimizing the Mean Squared Error (MSE) of the model on the training data.

C. Support Vector Regression (SVR)

SVR is a machine learning algorithm for regression analysis. It extends the concept of Support Vector Machines (SVM) used for classification tasks to the regression context. SVR aims to find a hyperplane that best fits the data points while maximizing the margin from the closest data points (support vectors) (Liu, Wang and Gu, 2021).

Given a training dataset $D=\{(x_1,y_1),\dots,(x_n,y_n)\}$, SVR aims to find a function $f(x)=w\cdot x+b$ that approximates the relationship between the input features x and the target variable y.

SVR introduces the ε-insensitive loss function, meaning that the error is only considered if it exceeds a certain threshold ε. The SVR objective is to minimize the cost function:

$$L(w,b)=\frac{1}{2}\parallel w\parallel^2+C\sum_{i=1}^{n}(\xi_i+\xi_i^*) \tag{4}$$

subject to the constraints:

$$\begin{aligned} &y_i-w\cdot x_i-b\le\epsilon+\xi_i\\ &w\cdot x_i+b-y_i\le\epsilon+\xi_i^*\\ &\xi_i,\xi_i^*\ge 0\end{aligned} \tag{5}$$

where, w denotes the weight vector and b is the bias term. C is the regularization parameter that controls the trade-off between maximizing the margin and minimizing the error. The slack variables, ξ_i and ξ_i^*, handle instances that are difficult to separate perfectly.

In this study, we use the Radial Basis Function (RBF) kernel, which is defined as:

$$K(x,z)=\exp(-\gamma\parallel x-z\parallel^2) \tag{6}$$

D. Long Short-Term Memory (LSTM)

LSTM networks, introduced by Hochreiter and Schmidhuber [5], are a specialized variant of recurrent neural networks (RNN) meticulously engineered to address sequence prediction challenges. Their distinctive architecture, which facilitates the retention of patterns over extended durations, renders LSTMs especially proficient for time series modeling. In the context of this study, we harness the capabilities of the LSTM network for our forecasting endeavors.

LSTM networks consist of memory cells that are regulated by three gates: forget, input, and output gates. These gates determine how information flows through the memory cells.

Forget Gate:

$$f_t=\sigma\left(W_f\cdot[h_{t-1},x_t]+b_f\right) \tag{7}$$

Input Gate:

$$\begin{aligned} i_t&=\sigma(W_i\cdot[h_{t-1},x_t]+b_i)\\ \tilde{C}_t&=\tanh(W_C\cdot[h_{t-1},x_t]+b_C)\end{aligned} \tag{8}$$

Update of Cell State:

$$C_t=f_t\times C_{t-1}+i_t\times\tilde{C}_t \tag{9}$$

Output Gate:

$$\begin{aligned} o_t&=\sigma(W_o[h_{t-1},x_t]+b_o)\\ h_t&=o_t\times\tanh(C_t)\end{aligned} \tag{10}$$

where, σ represents the sigmoid function, W and b are the weight matrices and biases for each gate, respectively, x_t is the input at time t, and h_t is the output.

E. Gated Recurrent Unit (GRU)

Introduced by [18], GRUs are a streamlined variant of the RNN designed to adeptly capture long-term sequence dependencies. Functioning as a simplified version of LSTMs, GRUs are characterized by two pivotal gates: the update gate and the reset gate. The update gate is instrumental in determining the proportion of the preceding hidden state that should be relayed to the subsequent state. Concurrently, the reset gate ascertains the extent to which the prior hidden state is disregarded. The computations for the GRU model are as follows:

$$\begin{aligned} r_t&=\sigma(W_r\cdot[h_{t-1},x_t]+b_r)\\ z_t&=\sigma(W_z\cdot[h_{t-1},x_t]+b_z)\\ \tilde{h}_t&=\tanh(W\cdot[r_t\odot h_{t-1},x_t]+b)\\ h_t&=(1-z_t)\odot h_{t-1}+z_t\odot\tilde{h}_t\end{aligned} \tag{11}$$

where, r_t and z_t are the reset and update gates at time t respectively, σ denotes the sigmoid activation function, W and b are the weight matrices and bias vectors, $\odot$ represents element-wise multiplication, and h_t is the hidden state at time t.

F. Grid Search Hyperparameter Tuning

In this study, optimizing hyperparameters becomes paramount to ensure the robustness of machine learning models. As demonstrated in Fig. 1, our approach harnesses a comprehensive grid search to navigate the vast hyperparameter space. For the SVR model, adjustments are made to the regularization parameter, gamma, and epsilon values. Meanwhile, the LSTM's performance is fine-tuned considering the number of units, dropout rate, and batch size. On the other hand, the GRU model sees alterations in its units, batch size, and number of epochs. This methodical approach, anchored in a three-dimensional exploration, seeks to refine our forecasting tools, aligning them with the sophisticated dynamics of today's financial markets.

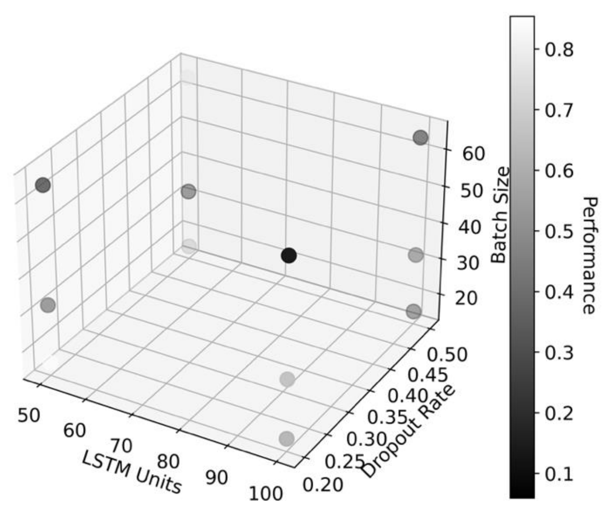

Fig. 1. 3D visualization of grid search.

G. Evaluation Metrics

To evaluate the predictive performance of the models, followed by [19] and [20], we utilize several well-established metrics: R-squared (R^2), Root Mean Square Error (RMSE), Mean Absolute Percentage Error (MAPE) and Mean Absolute Error (MAE). Each metric provides a different perspective on the quality of the predictions.

R^2 : R^2 measures the proportion of the variance in the dependent variable that is predictable from the independent variable. It ranges from 0 to 1, with 1 indicating perfect prediction. It is calculated as follows:

$$R^2 = 1 - \frac{\sum (y_t - \hat{y}_t)^2}{\sum (y_t - \bar{y})^2} \tag{12}$$

RMSE (Root Mean Square Error): RMSE represents the square root of the second sample moment of the differences between predicted and observed values or the quadratic mean of these differences. It is interpreted as the standard deviation of the unexplained variance:

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n} (y_t - \hat{y}_t)^2} \tag{13}$$

The Mean Absolute Percentage Error (MAPE): MAPE provides an easy-to-interpret measure of the average prediction error in percentage terms. It is especially useful when comparing the performance of different models on the same dataset.

The MAPE is calculated as follows:

$$\text{MAPE} = \frac{100}{n}\sum_{t=1}^{n} \left|\frac{y_t - \hat{y}_t}{y_t}\right| \tag{14}$$

Mean Absolute Error (MAE): MAE measures the average of the absolute differences between the predicted and observed values. It provides an idea of the magnitude of the error, without considering the direction. Lower MAE values indicate a better fit to the data. It is calculated as:

$$MAE = \frac{1}{n}\sum_{t=1}^{n} |y_t - \hat{y}_t| \tag{15}$$

where, y_t is the actual value at time $t, \hat{y}_t$ is the predicted value at time , and $\bar{y}$ represents the mean of the actual values.

H. Forecasting Algorithm

Our methodology, detailed in Fig. 2, commenced by dividing the data into training and testing subsets. Depending on the chosen model-traditional techniques like ARIMA and Holt's LES or more contemporary machine learning approaches-appropriate parameter optimization processes were undertaken, with the latter employing a grid search. Using a rolling window framework, we executed forecasts for three distinct time horizons: H=1, 10, and 30 days. Utilizing the rolling window approach, each forecast integrated the most recent observation from the testing set into the training dataset. This method allowed our models to consistently update and adapt based on the newest economic data available. Once the end of the testing data was reached, we compared the performance of the various models under different forecasting time horizons using key metrics such as R^2, RMSE, MAE and MAPE.

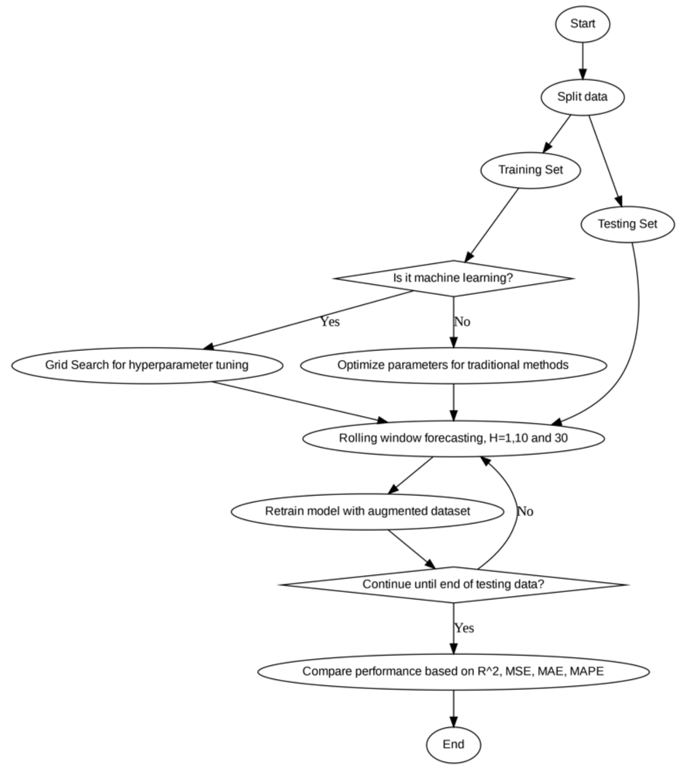

Fig. 2. Flowchart of algorithm.

IV. NUMERICAL RESULTS

A. Data Description

This study evaluates the daily performance of two major stock market indices, the S&P 500 and the Shanghai Stock Exchange (SSE) Composite Index, spanning December 31, 2012, to December 31, 2022. These indices were chosen due to their importance in representing overall stock market performance in the United States and China, respectively, and their influence on global financial markets. Both are market-capitalization-weighted, capturing broad market movements efficiently.

We obtain daily closing prices of the indices from the Yahoo Finance API, a publicly accessible and reliable data source extensively used in financial research. The data, adjusted for splits and dividends, provide an accurate representation of the indices' performance over the period.

The data are partitioned into training and testing sets. The training set, consisting of data before 2020, is used to calibrate forecasting models, while the testing set, from January 1, 2020, to December 31, 2022, evaluates their out-of-sample performance.

Table I summarizes the descriptive statistics of the daily closing prices for both indices over the study period. The S&P 500 index, with a mean of 2742.1700 and standard deviation of 873.0140, traded between 1426.1900 and 4796.5600. The SSE index had a mean of 3017.0600, standard deviation of 527.9180, and prices between 1950.0100 and 5166.3500. The S&P 500 displays a negative kurtosis of -0.6494 and positive skewness of 0.6749, suggesting a less peaked and right-skewed distribution. The SSE index, with a kurtosis of 0.8312 and near-zero skewness of 0.0525, indicates a more peaked and symmetric distribution. There are 2519 and 2428 observations for the S&P 500 and SSE indices, respectively. Fig. 3 and Fig. 4 depict the time series of daily closing prices for the S&P 500 and SSE indices.

TABLE I. DESCRIPTIVE STATISTICS FOR THE S&P 500 AND SSE INDICES

Index	S&P500	SSE
Mean	2742.1700	3017.0600
Std	873.0140	527.9180
Minimum	1426.1900	1950.0100
Maximum	4796.5600	5166.3500
Kurtosis	-0.6494	0.8312
Skewness	0.6749	0.0525
Count	2519	2428

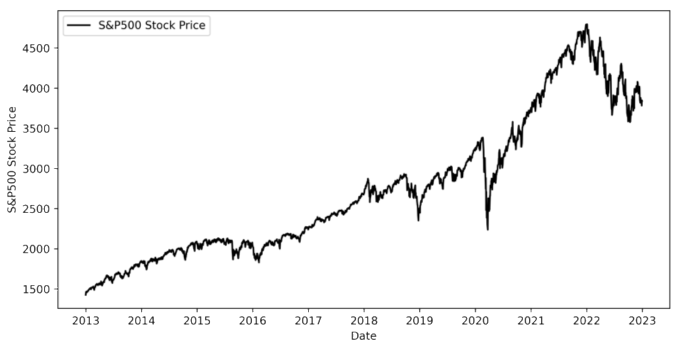

Fig. 3. S&P 500 index price.

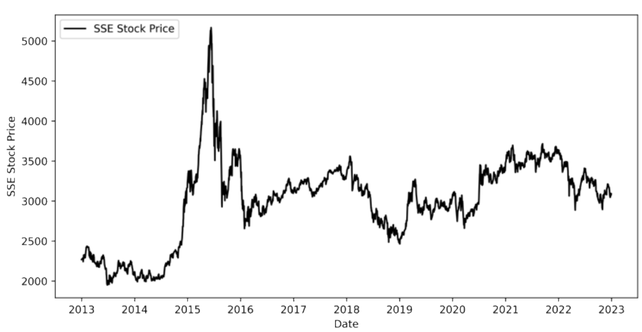

Fig. 4. SSE index price.

B. Determination of Parameters of Traditional Methods

For the ARIMA models, we use an automatic order selection method that seeks to minimize the Akaike Information Criterion (AIC). The ARIMA model parameters include the order of the autoregressive (AR) and moving average (MA) components, as well as the degree of differencing. Given the daily frequency of the data, we focus on non-seasonal models. Through this procedure, we identify ARIMA (2,1,0) as the best model for the SSE index, while ARIMA(1,1,1) with an intercept is chosen for the S&P 500 index.

For the Holt's LES models, an optimization procedure is applied to estimate the smoothing parameters for the level and trend components. The models are fitted to the training data of both indices. For the SSE index, the estimated smoothing level is approximately 0.995, and the smoothing trend is about 0.0237. For the S&P 500 index, the respective values are approximately 0.907 and 0.0212. The initial level and trend for both models are estimated based on the training data.

C. Determination of Optimal Hyperparameters

In this study, we employ a grid search approach to optimize the hyperparameters for the SVR, LSTM, and GRU models across different time horizons (H=1/10/30). For the SVR model, we consider three hyperparameters: the regularization parameter (C), gamma (γ), and epsilon (ε). For the LSTM and GRU models, the hyperparameters assessed include the number of units, dropout rate, and batch size. The selection of these hyperparameters is crucial as they directly affect the models' forecasting performance. We evaluate various hyperparameter combinations using a training dataset to identify the best-performing models, which are subsequently tested on a separate test dataset. We use a radial basis function (RBF) kernel for the SVR model. For the LSTM and GRU models, we compile them using the Adam optimizer and mean squared error (MSE) loss function, which is well-suited for regression tasks like stock price prediction. This hyperparameter optimization process is conducted across different forecasting horizons to evaluate the models' suitability for both short-term and long-term forecasting. The Hyperparameters are given in Table II.

TABLE II. HYPERPARAMETER SETTINGS FOR MACHINE LEARNING MODELS ACROSS VARIOUS TIME HORIZONS (H=1/10/30)

Model	Name of Parameter	S&P 500	SSE
SVR	Regularization parameter	10/10/10	10/10/10
	Gamma	0.1/0.1/0.1	0.1/0.1/0.1
	Epsilon	0.1/0.1/0.1	0.1/0.1/0.1
LSTM	Units	100/100/50	100/100/100
	Drop out	0.2/0.2/0.2	0.2/0.5/0.5
	Batch size	16/16/64	16/32/16
GRU	Units	70/70/30	50/50/50
	Batch size	16/64/32	16/64/16
	epochs	30/70/50	50/50/50

D. Comparison and Analysis

Table III presents the evaluation results for forecasting the S&P 500 index using various models: ARIMA, Holt's LES, SVR, LSTM, and GRU. The performance of each model is assessed across three-time horizons: 1-day, 10-day, and 30-day.

For the 1-day time horizon, the ARIMA model stands out, achieving an R^2 of 99.04%, MAPE of 1.06%, RMSE of 53.93, and MAE of 38.78. The Holt's LES model closely follows, with similar metrics. Both SVR and GRU models exhibit strong performance, with R^2 values exceeding 98%. Notably, the LSTM model shows the lowest R^2 of 94.70% and the highest MAPE of 2.71%.

TABLE III. EVALUATION OF S&P 500 INDEX FORECASTING ACROSS DIFFERENT TIME HORIZONS

Models	R^2	MAPE	RMSE	MAE
Time Horizon =1				
ARIMA	**99.04%**	**1.06%**	**53.93**	**38.78**
Holt's LES	99.02%	1.06%	54.48	38.62
SVR	98.95%	1.18%	56.31	43.16
LSTM	94.70%	2.71%	126.80	108.05
GRU	98.16%	1.60%	74.72	61.29
Time Horizon =10				
ARIMA	95.52%	2.27%	116.65	83.72
Holt's LES	94.93%	2.34%	124.00	87.07
SVR	94.18%	2.77%	132.87	103.20
LSTM	91.89	3.27%	156.82	131.59
GRU	94.18%	2.73%	132.84	100.79
Time Horizon = 30				
ARIMA	83.94%	4.35%	220.74	157.45
Holt's LES	75.48%	5.02%	272.73	181.37
SVR	81.06%	5.21%	239.69	189.03
LSTM	85.80%	4.42%	207.60	176.05
GRU	80.21%	5.34%	245.04	194.03

In the 10-day horizon, the ARIMA model again leads with an R^2 of 95.52% and the lowest MAPE of 2.27%. Holt's LES, SVR, and GRU models all report R^2 values above 94% and MAPE values under 3%. The LSTM model lags, with the lowest R^2 of 91.89% and the highest MAPE of 3.27%.

For the 30-day horizon, the LSTM model surprisingly achieves the highest R^2 of 85.80%, but with a relatively high MAPE of 4.42%. The ARIMA model follows with an R^2 of 83.94% and the lowest MAPE of 4.35%. The Holt's LES model's performance diminishes, recording the lowest R^2 of 75.48% and a higher MAPE of 5.02%. SVR and GRU models display similar R^2 values around 80% and MAPE values above 5%.

In summary, the ARIMA model consistently performs well across all time horizons, exhibiting the highest R^2 and the lowest MAPE for the 1-day and 10-day horizons. While the LSTM model underperforms in shorter horizons, it surprisingly has the highest R^2 for the 30-day horizon. The Holt's LES model performs well for shorter horizons but declines for the 30-day horizon. SVR and GRU models show moderate performance across all horizons.

Table IV presents the evaluation results of forecasting the SSE index. For the 1-day horizon, all models display strong performance, with R^2 values exceeding 97%. The SVR model leads with an R^2 of 97.81% and the lowest MAPE of 0.80%. ARIMA and Holt's LES models both achieve R^2 values around 97.78% and similar MAPE values of 0.81%. LSTM and GRU models also perform well, with R^2 values above 97.5% and MAPE values under 0.85%.

TABLE IV. EVALUATION OF SSE INDEX FORECASTING ACROSS DIFFERENT TIME HORIZONS

Models	R^2	MAPE	RMSE	MAE
Time Horizon =1				
ARIMA	97.78%	0.81%	36.25	26.34
Holt's LES	97.72%	0.81%	36.31	26.25
SVR	97.81%	0.80%	35.97	25.98
LSTM	97.49%	0.85%	38.54	27.72
GRU	97.65%	0.83%	37.27	27.19
Time Horizon =10				
ARIMA	90.16%	1.65%	76.34	53.31
Holt's LES	88.84%	1.78%	81.26	57.34
SVR	90.48%	1.56%	75.05	50.38
LSTM	94.74%	1.27%	55.82	40.89
GRU	92.98%	2.91%	145.90	113.95
Time Horizon = 30				
ARIMA	77.34%	2.73%	115.69	88.71
Holt's LES	72.21%	2.95%	128.34	95.70
SVR	77.27%	2.54%	116.00	82.29
LSTM	94.78%	1.25%	55.61	40.98
GRU	58.89%	3.80%	156.00	123.29

In the 10-day horizon, the LSTM model stands out with the highest R^2 of 94.74% and the lowest MAPE of 1.27%. SVR closely follows with an R^2 of 90.48% and a low MAPE of 1.56%. ARIMA and Holt's LES models both report R^2 values around 90% and MAPE values under 1.8%. The GRU model exhibits a solid R^2 of 92.98% but the highest MAPE of 2.91%.

For the 30-day horizon, the LSTM model clearly dominates with an R^2 of 94.78% and the lowest MAPE of 1.25%. The ARIMA and SVR models perform similarly, both achieving R^2 values around 77% and MAPE values under 2.75%. The Holt's LES model lags, with an R^2 of 72.21% and a higher MAPE of 2.95%. The GRU model shows the lowest performance with an R^2 of 58.89% and the highest MAPE of 3.80%.

In summary, the LSTM model consistently performs well across all time horizons, especially for the 30-day horizon, where it excels. The ARIMA and SVR models display similar moderate performance across all horizons. The Holt's LES model's performance declines for longer horizons. The GRU model exhibits strong performance in the 10-day horizon but struggles in the 30-day horizon.

V. CONCLUSION

The task of predicting stock market indices is essential for risk management, portfolio allocation, and derivative pricing, all of which contribute to stabilizing the financial market order. In this study, we compared the performance of several predictive models—ARIMA, Holt's LES, SVR, LSTM, and

GRU—across different time horizons (1-day, 10-day, and 30-day) for two prominent stock indices: the S&P 500 and the SSE. The models were evaluated based on four metrics: R-squared (R^2), Mean Absolute Percentage Error (MAPE), Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE).

Our empirical results indicate that:

- LSTM consistently performs well across all time horizons for both indices, especially in the 30-day horizon. It outperforms the other models in terms of both R^2 and MAPE. This result can be attributed to the model's ability to capture long-term dependencies in the data and its inherent adaptability in learning complex nonlinear relationships.

- ARIMA and SVR models display moderate performance across all time horizons for both indices, showcasing their robustness and applicability. The ARIMA model benefits from its ability to account for time trends, seasonality, and autoregressive behaviors. On the other hand, the SVR model leverages its capacity to model nonlinear relationships by using kernel functions.

- Holt's LES model performs well for the 1-day and 10-day horizons but struggles for longer horizons. The model's declining performance is attributed to its primary reliance on short-term trends, which may not capture more complex behaviors over longer time horizons.

- The GRU model performs well for shorter horizons but faces difficulties in the 30-day horizon. This could be due to the challenges posed by long-term dependencies in the data. GRU, similar to LSTM, is designed to address such challenges, but our results suggest that LSTM may be better suited for this particular dataset.

- Through extensive experimentation, we confirmed the robustness and applicability of our findings. For both indices, the results were consistent across different time horizons and evaluation metrics, confirming the validity of our conclusions.

In summary, our study provides valuable insights for investors and market analysts. The results can be used to enhance trading strategies, optimize portfolio allocations, and improve risk management approaches. Regulators may also benefit from these insights by identifying market anomalies and intervening when necessary to ensure financial market stability.

Despite the contributions of this study, we acknowledge that multivariate prediction was not considered. In future research, we will incorporate additional factors closely related to the stock indices' movements, such as macroeconomic indicators and sentiment analysis, to enhance the accuracy of our predictions. Incorporating these factors will not only improve the forecasting accuracy but also contribute to a deeper understanding of the underlying relationships that drive stock market dynamics. Besides, the methodologies and insights gained from this study hold the potential for broader applications beyond the S&P 500 and SSE indices. For instance, these models could be adapted to forecast emerging market indices, where volatility and data irregularities present unique challenges.

REFERENCES

[1] D. Zhang, M. Hu, and Q. Ji, 'Financial markets under the global pandemic of COVID-19', Finance Research Letters, vol. 36, p. 101528, Oct. 2020, doi: 10.1016/j.frl.2020.101528.

[2] G. E. P. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, Time Series Analysis: Forecasting and Control. John Wiley & Sons, 2015.

[3] C. C. Holt, 'Forecasting seasonals and trends by exponentially weighted moving averages', International Journal of Forecasting, vol. 20, no. 1, pp. 5–10, Jan. 2004, doi: 10.1016/j.ijforecast.2003.09.015.

[4] V. Vapnik, The Nature of Statistical Learning Theory. Springer Science & Business Media, 2013.

[5] S. Hochreiter and J. Schmidhuber, 'Long Short-Term Memory', Neural Computation, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.

[6] K. Cho et al., 'Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation'. arXiv, Sep. 02, 2014. doi: 10.48550/arXiv.1406.1078.

[7] A. A. Adebiyi, A. O. Adewumi, and C. K. Ayo, 'Comparison of ARIMA and Artificial Neural Networks Models for Stock Price Prediction', Journal of Applied Mathematics, vol. 2014, pp. 1–7, 2014, doi: 10.1155/2014/614342.

[8] M. Beniwal, A. Singh, and N. Kumar, 'Forecasting long-term stock prices of global indices: A forward-validating Genetic Algorithm optimization approach for Support Vector Regression', Applied Soft Computing, vol. 145, p. 110566, Sep. 2023, doi: 10.1016/j.asoc.2023.110566.

[9] B. Gülmez, 'Stock price prediction with optimized deep LSTM network with artificial rabbits optimization algorithm', Expert Systems with Applications, vol. 227, p. 120346, Oct. 2023, doi: 10.1016/j.eswa.2023.120346.

[10] A. Q. Md et al., 'Novel optimization approach for stock price forecasting using multi-layered sequential LSTM', Applied Soft Computing, vol. 134, p. 109830, Feb. 2023, doi: 10.1016/j.asoc.2022.109830.

[11] Y. Yu, Y. Lin, X. Hou, and X. Zhang, 'Novel optimization approach for realized volatility forecast of stock price index based on deep reinforcement learning model', Expert Systems with Applications, vol. 233, p. 120880, Dec. 2023, doi: 10.1016/j.eswa.2023.120880.

[12] K. E. ArunKumar, D. V. Kalaga, Ch. Mohan Sai Kumar, M. Kawaji, and T. M. Brenza, 'Comparative analysis of Gated Recurrent Units (GRU), long Short-Term memory (LSTM) cells, autoregressive Integrated moving average (ARIMA), seasonal autoregressive Integrated moving average (SARIMA) for forecasting COVID-19 trends', Alexandria Engineering Journal, vol. 61, no. 10, pp. 7585–7603, Oct. 2022, doi: 10.1016/j.aej.2022.01.011.

[13] A. Kumar, M. Paprzycki, and V. K. Gunjan, Eds., ICDSMLA 2019: Proceedings of the 1st International Conference on Data Science, Machine Learning and Applications, vol. 601. in Lecture Notes in Electrical Engineering, vol. 601. Singapore: Springer Singapore, 2020. doi: 10.1007/978-981-15-1420-3.

[14] P. Soni, Y. Tewari, and D. Krishnan, 'Machine Learning Approaches in Stock Price Prediction: A Systematic Review', J. Phys.: Conf. Ser., vol. 2161, no. 1, p. 012065, Jan. 2022, doi: 10.1088/1742-6596/2161/1/012065.

[15] D. Kobiela, D. Krefta, W. Król, and P. Weichbroth, 'ARIMA vs LSTM on NASDAQ stock exchange data', Procedia Computer Science, vol. 207, pp. 3836–3845, 2022, doi: 10.1016/j.procs.2022.09.445.

[16] M. Bilgili and E. Pinar, 'Gross electricity consumption forecasting using LSTM and SARIMA approaches: A case study of Türkiye', Energy, vol. 284, p. 128575, Dec. 2023, doi: 10.1016/j.energy.2023.128575.

[17] C. Lim and M. McAleer, 'Forecasting tourist arrivals', Annals of Tourism Research, vol. 28, no. 4, pp. 965–977, Jan. 2001, doi: 10.1016/S0160-7383(01)00006-8.

[18] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, 'Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling'. arXiv, Dec. 11, 2014. doi: 10.48550/arXiv.1412.3555.

[19] S. Paudel, A. Pudasaini, R. K. Shrestha, and E. Kharel, 'Compressive strength of concrete material using machine learning techniques', Cleaner Engineering and Technology, vol. 15, p. 100661, Aug. 2023, doi: 10.1016/j.clet.2023.100661.

[20] S. A. Rajakarunakaran et al., 'Prediction of strength and analysis in self-compacting concrete using machine learning based regression techniques', Advances in Engineering Software, vol. 173, p. 103267, Nov. 2022, doi: 10.1016/j.advengsoft.2022.103267.