A Fusion of LSTM and GRU Neural Networks for Predicting the S&P 500 Index

Shijon Das*†1
*Department of Electrical and Computer Engineering, Idaho State University, Pocatello, ID, USA.
Email: [1]shijondas@isu.edu

***Abstract*—This study explores the domain of financial market predictive analytics through the development of a hybrid deep learning model aimed at forecasting the future values of the S&P 500 index, a key benchmark of U.S. equity market performance. The proposed model combines Long Short-Term Memory (LSTM) units with Gated Recurrent Units (GRU) to utilize their respective strengths in capturing long-term dependencies and adapting to less persistent features in time-series data. The dataset is evaluated against traditional LSTM and GRU models, and performance metrics such as Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE) are meticulously analyzed to quantify the predictive accuracy. The results indicate that our composite LSTM-GRU architecture achieves a striking balance between computational efficiency and forecasting accuracy, making it an effective tool for investors and analysts seeking to navigate the complexities of equity markets. This study not only contributes to the existing body of knowledge in financial market prediction but also demonstrates the practical implications of hybrid deep learning models in the domain of economic forecasting.**

***Index Terms*—S&P 500 index, LSTM, GRU, RMSE, MAE, MAPE.**

I. Introduction

Forecasting serves as a method for predicting future occurrences by analyzing historical data. This practice is widespread across various domains such as business, economics, environmental sciences, and financial sectors. Depending on the domain, forecasting tasks are generally grouped into:

- Short-term forecasting, which deals with projections of a few seconds to a few months into the future.
- Medium-term forecasting, which extends predictions to one or two years ahead.
- Long-term forecasting, which looks beyond the two-year mark.

A time series dataset consists of sequential observations recorded over time concerning a particular variable. In the context of this study, that variable is the stock price. Such data may be univariate, containing information about a single stock, or multivariate, comprising stock prices from multiple companies over time. Time series analysis aids in discerning existing patterns, trends, and cycles within the data. In stock markets, understanding whether the market is trending upward (bullish) or downward (bearish) in advance can lead to informed investment decisions. Furthermore, recognizing patterns can highlight the best-performing companies over certain periods, underlining the significance of time series analysis and forecasting in research fields.

979-8-3503-8532-8/24/$31.00

Existing stock price forecasting methodologies fall into three categories [1]:

- Fundamental Analysis, which predicts the value of a company's stock based on economic factors like sales, earnings, and profits, ideal for long-term forecasts.
- Technical Analysis, which forecasts future prices using historical stock prices. Methods such as the moving average, the unweighted mean of the previous n data points, are commonly employed for short-term forecasts.
- Time Series Forecasting, which encompasses: a.
  1) Linear Models, including AR, ARMA, ARIMA, and their variants [2] [3] [4]. These models employ predefined equations to fit a mathematical model to univariate time series. However, they fail to capture the underlying patterns in the data, making them unsuitable for recognizing the intricacies of multivariate datasets or the interconnectedness of different stocks.
  2) Non-linear Models, such as ARCH, GARCH, [3] TAR, and deep learning algorithms [5]. A study [6] investigated the relationship between stock price and volume for selected companies, emphasizing the role of deep learning algorithms in stock price prediction [7] [8].

Deep neural networks are quite adept at approximating non-linear functions, effectively mapping complex functions. Depending on the application, various architectures are deployed, like multi-layer perceptrons (MLP), Recursive Neural Networks (RNN), Long Short Term Memory (LSTM), Gated Recurrent Unit (GRU), and Convolutional Neural Networks (CNN) [9]. These networks have seen applications in diverse areas, including image processing, natural language processing, and time series analysis, showcasing their versatility.

Section II discusses the motivation behind pursuing this research, while Section III discusses the previous works done in this field of research. Section IV, presents the data set preparation and the preprocessing along with the implementation details of the ensemble model, and in Section V, the model's performance in comparison to other forecasting models is discussed.

II. Motivation

Being able to predict stock price with a high accuracy is both a vital financial challenge and a significant academic interest, driven by its profound implications for investment strategies, risk management, and economic policy planning. Traditional methods like fundamental and technical analyses provide valuable insights but often fail to capture complex market dynamics and non-linear relationships. Similarly, linear and non-linear time series models have made strides but typically fall short due to their inherent limitations when it comes to handling the volatile nature of financial markets.

The advent of deep learning, particularly through Long Short-Term Memory (LSTM) networks and Gated Recurrent Units (GRU), has revolutionized predictive analytics, yet these models still face challenges when employed separately. This study is motivated by the potential of enhancing predictive accuracy by integrating LSTM and GRU architectures, leveraging their combined strengths to address the complexities of financial data.

This research aims to explore a hybrid deep learning framework that incorporates advanced techniques such as dropout, batch normalization, and data augmentation to develop a robust model capable of providing highly accurate forecasts. This research is conducted with the hypothesis that this approach will overcome the limitations of previous models, offering a more reliable and efficient tool for market prediction, thus significantly impacting financial decision-making and strategy development.

III. Related Work

Chen et al. explored the use of an LSTM model for predicting the returns on Chinese stocks, where historical data was organized into sequences of 30 days with ten learning features and labels for 3-day returns. The study reported an improvement in prediction accuracy from 14.3% to 27.2%, marking a significant improvement over random prediction methods [10]. Bao et al. utilized the Haar wavelet transformation for filtering noise from financial time series before employing stacked autoencoders for extracting deep features, subsequently using an LSTM to predict the closing prices of stock indices. They recorded an average R score under 88% on their LSTM models for the S&P 500 [11].

Roondiwala et al. applied an LSTM model to forecast closing prices using data from the NIFTY 50 index spanning 2011 to 2016. The model utilized only fundamental data such as open, high, low, and close prices and did not incorporate any macroeconomic or technical indicators [12]. Fischer and Krauss, analyzed the potential of LSTM networks for classifying and predicting directional movements of the S&P 500's constituent stocks between 1992 and 2015. The findings from their research indicated that LSTM networks could effectively decipher significant patterns from financial time series data and outperformed other models like random forests, standard deep networks, and logistic regression in terms of prediction accuracy and net daily returns after accounting for transaction costs [13].

Qiu et al. enhanced an LSTM-based model by integrating an attention mechanism to assess information in news articles that could influence price volatility. They pre-processed the data with wavelet transformations to denoise it before applying their attention-based LSTM framework to predict the opening prices of indices such as the S&P 500, Dow Jones, and Hang Seng, using only fundamental market data [14]. Lanbouri and Achchab, focused on high-frequency trading, using LSTM to predict the S&P 500 stock prices for the forthcoming 1, 5, and 10 minutes [15].

Yadav et al. implemented an LSTM model with various hidden layers to predict closing stock prices in the Indian market, explicitly removing trend and seasonality components from the data. Their studies concluded that a single-layer LSTM model was most effective in terms of prediction accuracy [16]. Kara et al. explored the use of support vector machines (SVM) and artificial neural networks to predict movements in the daily Istanbul Stock Exchange National 100 Index from 1997 to 2007, finding that the neural network model significantly outperformed the SVM model [17].

Karmiani et al. compared several prediction models, including LSTM, Backpropagation, SVM, and Kalman filter, on the stock prices of nine selected companies, identifying LSTM as the superior model in terms of prediction accuracy and consistency [18]. Yu and Yan combined the phase-space reconstruction method with LSTM to analyze various market environments such as the S&P 500 and DJIA. They reported that LSTM surpassed other models including Multilayer Perception, Support Vector Regressor, and ARIMA for S&P 500 data [19].

Gao et al. conducted a comparative analysis using four machine learning algorithms — Multilayer Perceptron, LSTM, Convolutional Neural Network, and Uncertainty-Aware Attention — to predict next day's stock prices. They considered a range of predictors like open and close prices, trading volume, and MACD for indices representing developed, less developed, and developing markets. Their findings suggested that the performance of the Uncertainty-Aware Attention model was marginally superior to the others, and that incorporating additional predictors such as the volatility index and unemployment rate could further enhance model performance [20].

IV. Methodology

A. Dataset

For this study, the S&P dataset from the year 2000 to around 2020 has been used. The S&P 500 dataset provides a comprehensive view of the historical performance and dynamics of the S&P 500 index, a benchmark index for the overall health of the US stock market. The dataset contains crucial market indicators such as Open, High, Low, Close, Adjusted Close prices, and trading Volume. Each feature in the dataset offers unique insights into the market's behavior and the thinking of the investors, crucial for understanding trends, volatility, and significant market events over time.

The dataset's richness allows for comprehensive research and analysis, offering insights into market trends, volatility patterns, and investor behavior. Fig. 1 shows the plot of every index with respect to date. Analyzing the plots of the index reveals several key observations. Firstly, there is a visible upward trajectory

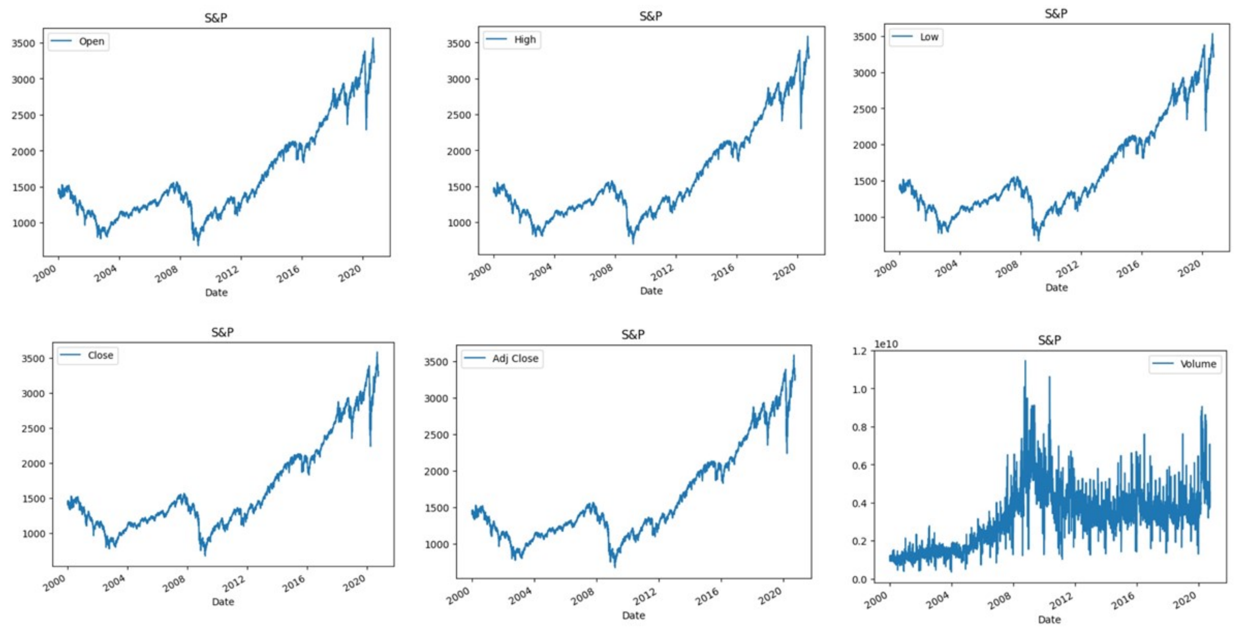

Fig. 1. Figure showing plots of every index with respect to date.

in the index's value over the years, reflecting the long-term growth and resilience of the market despite periodic downturns. Major economic events such as the global financial crisis in 2008 and the COVID-19 pandemic around 2020 are evident in the data, showcasing the market's response to external shocks and subsequent recovery phases. Additionally, fluctuations in trading volume highlight periods of heightened market activity and investor participation, often coinciding with significant market events or periods of uncertainty.

Fig. 2 is a histogram showing the distribution for the Closing Price. Looking at the histogram, it seems that the close prices are not normally distributed. Rather, there are peaks at certain intervals which could suggest some form of data grouping or binning. It appears that the most common close price range falls between roughly 1000 to 1500, as indicated by the tallest bar. The distribution is not symmetrical.There are several peaks and the data is somewhat spread out, indicating variability in the closing prices. There's also a noticeable gap in the frequency between price ranges of around 2000 to 2500, where there are significantly fewer occurrences. The shape of the histogram suggests that there may be multiple modes - local high points in the distribution of close prices - which could imply that the asset has had periods of stability around certain price levels.

The organisation of this dataset is important as it forms the basis of the predictive model used in this study. Through this study we are not only looking at the S&P 500's history, but we're also trying to predict what might happen in the future. We want to build a model that's better than the ones in the existing literature.

B. Data Preprocessing

The first part of the data preprocessing involved organizing our data in a way that makes it easy to follow the timeline, using the 'Date' column as the main reference point. This helps us keep track of when each piece of information was recorded and ensures there are no gaps in our data.

Looking at the data visually using a correlation heatmap as shown in Fig. 3, we saw that the 'Open', 'High', 'Low', 'Close', and 'Adj Close' features all have a strong positive relationship with each other. This makes sense because they all track the price movements of the S&P 500 index, with slight adjustments in the 'Adj Close' feature for things like dividends and stock splits.

On the other hand, the 'Volume' column, which represents the number of shares traded, had a weaker positive relationship with the price columns. This means that while there is some connection between trading volume and price movements, it's not as strong as the relationship between the price columns themselves.

The most important preprocessing step was normalization. This involves adjusting the scale of our data to make it more consistent and easier for our models to understand. We used a technique called MinMaxScaler to scale our data so that all the features fall within a range of 0 to 1. This helps our models learn more effectively and ensures that no single feature has too much influence on the final predictions.

To test our models, we split the dataset into two parts: a training set and a test set. The training set contained the first 3000 data points, allowing the model to learn from a large

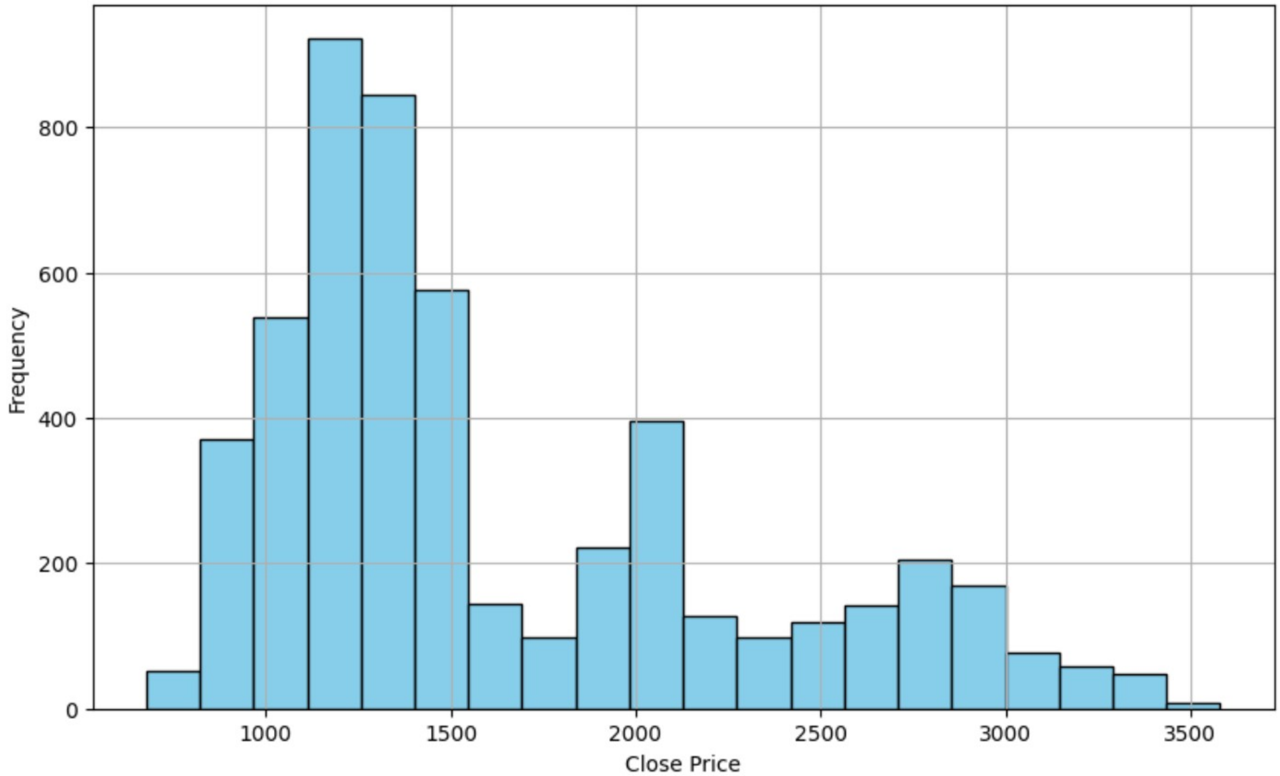

Fig. 2. Histogram showing the distribution of Closing Price

amount of historical data. Amongst the 3000 instances, 500 were used for the validation set. The test set, starting from data point 3500, was used to evaluate how well the model performs on unseen.

C. Model Specification

This section contains an overview of the deep learning models which are applied to the dataset. The machine learning models chosen for this study are LSTM and GRU.

D. Long Short Term Memory (LSTM)

We started off with LSTM architectures for our prediction task.The first LSTM model - Model 1 - is a straightforward stack of LSTM layers, beginning with a 100-unit layer designed to read the sequences of our time series data. Following that, we stack another 100-unit LSTM layer to further dissect the patterns. We finish with a 50-unit LSTM layer that distills all this information down and preps it to predict a single value: our next day's stock price. We hope this model, with its simplicity, could capture the essence of market movements.The next model -Model-2- introduced Dropout and Batch Normalization after each LSTM layer. Dropout prevents it from getting too fixated on the training data and helping it generalize better. With these additions, we aim to craft a model that's not only smart but also adaptable and robust. Model 3 takes Model 2's architecture but adds some strategic tweaks to optimize learning. We introduced ReduceLROnPlateau and EarlyStopping fro keeping an eye on the model's learning process. If the model hits a learning plateau, ReduceLROnPlateau gives it a gentle nudge by adjusting the learning rate, helping it to learn more efficiently. EarlyStopping is our safety net, ensuring that if learning stalls, we don't waste time and resources. Finally, Model 4 takes the robust framework of Model 3 and enriches it with data augmentation. By injecting noise, shifting sequences, and scaling, we mimic the unpredictability of the stock market, aiming to build a model that's unshaken by the market's capricious nature. The hypothesis behind the addition of hyperparamters is rooted in the belief that the stock market is not just complex, but also noisy and non-linear. Each modification to the models is an attempt to capture these characteristics better. The layered LSTM structure aims to decode the time series data, while the regularization techniques and learning optimizers are there to make the model resilient and efficient. Lastly, data augmentation ensures the model is not thrown off by unexpected volatility. Table I shows the comparision of the 4 LSTM architectures.

1) Gated Recurrent Unit (GRU): The GRU model architecture excels at capturing sequential data patterns. Unlike traditional RNNs, which suffer from issues like the vanishing gradient problem, GRUs are designed to better retain long-term dependencies in data sequences. The architecture begins with a GRU layer containing 100 units, enabling it to analyze sequential data while retaining memory of past information. To prevent overfitting, a dropout layer is introduced, randomly dropping a fraction of connections during training. Additionally, batch normalization is applied to stabilize the learning process

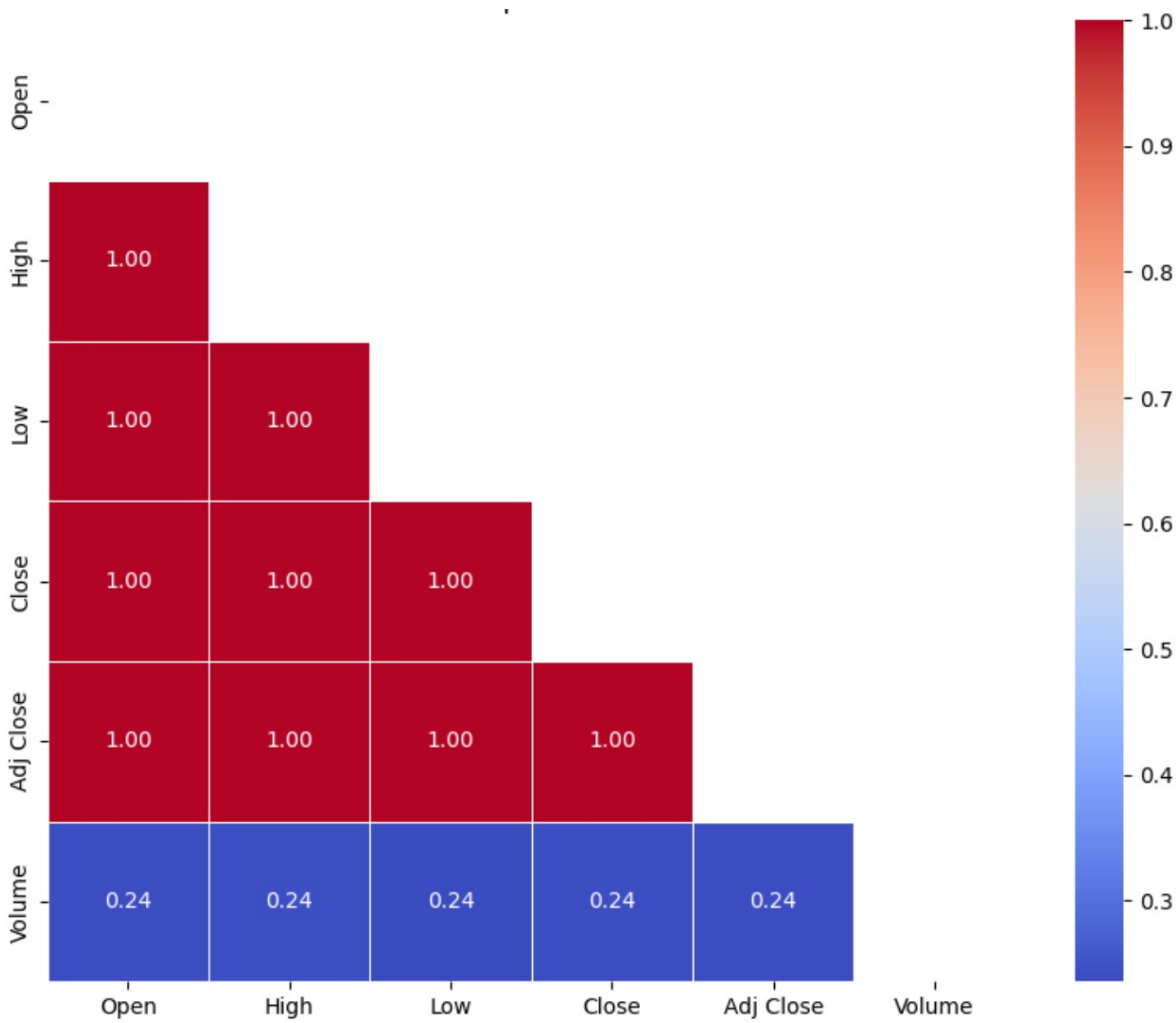

Fig. 3. Histogram showing the distribution of Closing Price

by standardizing the inputs. Following this, another GRU layer with 100 units is added, further enhancing the model's ability to capture intricate temporal patterns. Similar dropout and batch normalization techniques are applied to maintain model robustness and stability. Finally, a third GRU layer with 50 units refines the feature extraction process, extracting relevant information for predicting the target variable. The addition of a dense layer at the end consolidates the extracted features into a single output - the predicted stock price. By incorporating GRU alongside LSTM, we aim to leverage the strengths of both architectures, allowing our model to capture a wide range of temporal dynamics and improve overall prediction accuracy. Additionally, the use of data augmentation techniques, such as random noise injection, time series shifting, and scaling, further enhances the model's resilience to noise and variability in the data, ultimately leading to more robust predictions.

2) LSTM- GRU Network Model Construction: For predicting the stock prices, we adopt a novel approach by integrating two proven deep learning architectures: LSTM, and GRU. This model fusion is made to capture not just the short-term dependencies but also the long and complex temporal relationships inherent in financial time series data.

In the model, the first LSTM layer is equipped with 100 units, designed to recognize patterns over extended sequences, making it ideal for the initial interpretation of the time series data. To mitigate overfitting, a Dropout layer with a rate of 0.2 follows, pruning some connections randomly to ensure the model's robustness. Batch Normalization is then employed, stabilizing the learning process by normalizing the input layer by re-centering and rescaling.

Building on this, we introduce a GRU layer, also with 100 units. GRUs, being computationally efficient alternatives to LSTMs, offer the model additional prowess in capturing dependencies without the burden of excessive parameters. This is paired with another Dropout and Batch Normalization, preserving the regularization theme and ensuring consistent

TABLE I
COMPARISON OF LSTM MODEL ARCHITECTURES

Model	Architecture
Model 1	LSTM(100, return_sequences=True) LSTM(100, return_sequences=True) LSTM(50, return_sequences=False) Dense(1)
Model 2	LSTM(100, return_sequences=True) Dropout(0.2) BatchNormalization() LSTM(100, return_sequences=True) Dropout(0.2) BatchNormalization() LSTM(50, return_sequences=False) Dropout(0.2) BatchNormalization() Dense(1)
Model 3	LSTM(100, return_sequences=True) Dropout(0.2) BatchNormalization() LSTM(100, return_sequences=True) Dropout(0.2) BatchNormalization() LSTM(50, return_sequences=False) Dropout(0.2) BatchNormalization() Dense(1)
Model 4	LSTM(100, return_sequences=True) Dropout(0.2) BatchNormalization() LSTM(100, return_sequences=True) Dropout(0.2) BatchNormalization() LSTM(50, return_sequences=False) Dropout(0.2) BatchNormalization() Dense(1)

training.

Another LSTM layer is added with 50 units for achieving further depth. This layer continues the work of pattern detection, fine-tuning the model's ability to identify subtler trends and nuances in the data, a crucial feature in forecasting complex financial indices like the S&P 500.

The architecture concludes with a second GRU layer of 50 units to finalize the feature extraction process, followed by yet another Dropout and Batch Normalization sequence. The final Dense layer boils down the intricate features extracted by the recurrent layers into a single output — the predicted stock price.

To optimize our model, we incorporate a ReduceLROnPlateau, which adjusts the learning rate when the validation loss plateaus, ensuring efficient and dynamic learning. EarlyStopping is also employed as a form of computational prudence, halting the training if the validation loss fails to improve, thus preventing wasted epochs and potential overfitting.

This LSTM-GRU hybrid architecture is compiled with the Adam optimizer, a reliable choice for fast and effective backpropagation through time. The combined strength of LSTM and GRU, with their complementary capabilities, equips our model with a profound understanding of temporal dynamics, which is believed to enhance accuracy in stock market predictions.

Lastly, to enhance the robustness of our model against the noise inherent in financial data, we implement a data augmentation strategy. By introducing random noise, shifting time series, and scaling data points, we ensure that our model is trained on a dataset reflective of real-world variability, strengthening its predictive power in a volatile market.

V. PERFORMANCE EVALUATION

For this study, three evaluation metrics have been used to measure the performance of the model. Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and Root Mean Squared Error (RMSE) are the three evaluation metrics used. MAE measures the average absolute difference between the predicted values and the true values. The calculation method is shown in Eq. ((1)). MAPE calculates the percentage difference between predicted and true values using Eq. ((2)). RMSE uses Eq. ((3)) to measure the square root of the average squared difference between predicted and true values. MAPE calculates the percentage difference between predicted and true values.

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|\hat{y}_i - y_i| \tag{1}$$

$$\mathrm{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{|\hat{y}_i - y_i|}{|y_i|}\right) \times 100 \tag{2}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2} \tag{3}$$

TABLE II
PERFORMANCE METRICS OF LSTM MODELS

Model	RMSE	MAE	MAPE (%)
Model 1	0.379	0.362	274.47
Model 2	0.065	0.053	50.65
Model 3	0.130	0.117	99.90
Model 4	0.055	0.043	35.09

Table II shows the results of the 4 LSTM models. Model 1 has a straightforward LSTM architecture with multiple layers but no regularization or advanced training strategies. The metrics indicate that while it can capture some patterns in the data, it may not be dealing well with overfitting or capturing more complex patterns, as indicated by the relatively high error rates. Model 2 introduces dropout and batch normalization, which are techniques designed to reduce overfitting and improve model generalization. These additions seem to have a substantial impact on performance, with all three metrics showing significant improvement over Model 1. Model 3 further builds on Model 2 by incorporating learning rate reductions on plateaus and early stopping, which are techniques to refine the learning process. Surprisingly, Model 3's metrics are worse than Model 2's. This may be due to the influence of the specific dataset and how these additional training strategies interacted with the data, possibly stopping the training prematurely or reducing the learning rate too much. Model 4 extends Model 3's structure by adding data augmentation, which helps to introduce more variability in the training data and can lead to better generalization. This model achieves the best performance across all three metrics, suggesting that the combination of LSTM layers with dropout, batch normalization, learning rate adjustments, early stopping, and data augmentation provides the best predictive accuracy for stock prices.

In conclusion, based on the provided metrics, Model 4 outperforms the others, with the lowest RMSE, MAE, and MAPE values. This suggests that its architecture and training strategy are the most effective at capturing the underlying patterns in the stock price data and predicting future prices. The incorporation of data augmentation likely played a significant role in achieving this, by preventing overfitting and allowing the model to learn a more robust representation of the data.

TABLE III
PERFORMANCE METRICS OF GRU MODEL VS. BEST LSTM MODEL

Model	**RMSE**	**MAE**	**MAPE (%)**
GRU Model	0.060	0.049	34.60
Best LSTM Model	0.055	0.043	35.09

Table III shows the performace of the GRU in comaprison to the best LSTM architecture which was of Model - 4. When compared to the LSTM models, the GRU model exhibits a competitive performance, with its RMSE, MAE, and MAPE being very close to those of the best-performing LSTM model (Model 4). This suggests that GRUs are quite effective for this particular prediction task. Given the slightly higher RMSE and MAE but a slightly better MAPE, the GRU model seems to be on par with the LSTM in terms of accuracy but may have some advantages in terms of computational efficiency due to its simpler architecture. The GRU model slightly lags behind Model 4 in terms of RMSE and MAE, but surpasses it with a lower MAPE. This suggests that while Model 4 was slightly better at minimizing errors on average, the GRU model may have been more consistent in its predictions relative to the actual values.

The GRU model, despite its architectural simplicity compared to the LSTMs, holds its ground well, with performance metrics that are comparable to those of the more complex LSTM models. The GRU model's slightly higher RMSE and MAE compared to Model 4 suggest that it may not predict as closely to the actual values on average. However, its lower MAPE indicates that when it comes to relative errors, it may be more accurate proportionally, especially in scenarios where the price values vary significantly. The close performance between the GRU and the best LSTM model (Model 4) suggests that for this dataset and task, the additional complexity of LSTMs may not provide a significant advantage over GRUs.

Table IV shows how the combined LSTM - GRU model performed in comparision to the best LSTM model and the GRU architecture. The combined LSTM and GRU model offers a blend of the two types of recurrent units in an effort to capitalize on the strengths of both. By examining the performance metrics, we can draw a comparison to the best performing LSTM (Model 4) and the GRU model.

TABLE IV
PERFORMANCE METRICS OF COMBINED LSTM AND GRU MODEL, BEST LSTM MODEL, AND GRU MODEL

Model	**RMSE**	**MAE**	**MAPE (%)**
Combined LSTM and GRU Model	0.056	0.045	34.84
Best LSTM Model (Model 4)	0.055	0.043	35.09
GRU Model	0.060	0.049	34.60

In comparison to the best LSTM model, the combined architecture has a slightly higher RMSE and MAE. This could suggest that while the combination of LSTM and GRU layers introduces more complexity, it does not necessarily translate into better predictive accuracy for this particular dataset. However, the combined model does show a marginal improvement in MAPE, which indicates that the errors are slightly lower in proportion to the actual values they predict.

When we turn to the GRU model, we see that the combined model has better RMSE and MAE metrics, implying that it was closer to the actual stock prices on average. Although the difference in MAPE is small, it suggests that the combined model was slightly more consistent in the percentage terms across all the predictions it made.

The differences in performance between these models are relatively minor and would need to be weighed against other factors such as training time, computational efficiency, and model complexity. The combined model might be more computationally expensive due to the added complexity of utilizing both LSTM and GRU layers, and yet the improvement in performance might not justify this extra cost. It's also possible that the slight improvements seen with the combined model could be within the margin of error and might not be statistically significant.

In essence, choosing between these models would depend on the specific requirements and constraints of the deployment environment. If computational resources are limited, the simpler GRU model might be preferred. If the goal is to slightly reduce percentage errors and resources are abundant, then the combined model could be the better choice. However, in many practical applications, the best LSTM model might offer a good balance between complexity and performance, especially given its lower RMSE and MAE compared to the GRU and the very close MAPE to the combined model.

VI. CONCLUSION

To conclude, the superior performance of the combined LSTM and GRU architecture improves stock price predictions. LSTMs catch long-term patterns, while GRUs enhance efficiency and grasp intricate data relationships. The proposed fusion model beats individual LSTM and GRU models, as seen in lower error rates. By combining LSTM and GRU strengths, the combined model provides a robust tool for predicting stock prices, aiding investors and analysts in making better decisions.

As we refine our approach, we aim to further enhance prediction accuracy, benefitting stock trading strategies.

References

[1] A. V. Devadoss and T. A. A. Ligori, "Forecasting of stock prices using multi layer perceptron," *Int J Comput Algorithm*, vol. 2, pp. 440–449, 2013.

[2] J. G. De Gooijer and R. J. Hyndman, "25 years of time series forecasting," *International journal of forecasting*, vol. 22, no. 3, pp. 443–473, 2006.

[3] V. K. Menon, N. C. Vasireddy, S. A. Jami, V. T. N. Pedamallu, V. Sureshkumar, and K. Soman, "Bulk price forecasting using spark over nse data set," in *International Conference on Data Mining and Big Data*. Springer, 2016, pp. 137–146.

[4] G. E. P. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time series analysis: forecasting and control*. John Wiley Sons, 2015.

[5] G. Batres-Estrada, "Deep learning for multivariate financial time series," Technical Report, Stockholm, Tech. Rep., May 2015.

[6] P. Abinaya, V. S. Kumar, P. Balasubramanian, and V. K. Menon, "Measuring stock price and trading volume causality among nifty50 stocks: The toda yamamoto method," in *Advances in Computing, Communications and Informatics (ICACCI), 2016 International Conference on*. IEEE, 2016, pp. 1886–1890.

[7] J. Heaton, N. Polson, and J. Witte, "Deep learning in finance," *arXiv preprint arXiv:1602.06561*, 2016.

[8] H. Jia, "Investigation into the effectiveness of long short term memory networks for stock price prediction," *arXiv preprint arXiv:1603.07893*, 2016.

[9] Y. Bengio, I. J. Goodfellow, and A. Courville, "Deep learning," *Nature*, vol. 521, pp. 436–444, 2015.

[10] K. Chen, Y. Zhou, and F. Dai, "A lstm-based method for stock returns prediction: A case study of china stock market," in *2015 IEEE International Conference on Big Data (Big Data)*, 2015, pp. 2823–2824.

[11] W. Bao, J. Yue, and Y. Rao, "A deep learning framework for financial time series using stacked autoencoders and long-short term memory," *PLoS One*, vol. 12, no. 7, pp. 1–24, 2017.

[12] M. Roondiwala, H. Patel, and S. Varma, "Predicting stock prices using lstm," *International Journal of Science and Research (IJSR)*, vol. 6, no. 4, pp. 1754–1756, 2017.

[13] 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.

[14] J. Qiu, B. Wang, and C. Zhou, "Forecasting stock prices with long-short term memory neural network based on attention mechanism," *PLoS One*, vol. 15, no. 1, pp. 1–15, 2020.

[15] Z. Lanbouri and S. Achchab, "Stock market prediction on high frequency data using long-short term memory," *Procedia Computer Science*, vol. 175, pp. 603–608, 2020.

[16] A. Yadav, C. K. Jha, and A. Sharan, "Optimizing lstm for time series prediction in indian stock market," *Procedia Computer Science*, vol. 167, pp. 2091–2100, 2020.

[17] Y. Kara, M. Acar Boyacioglu, and Ö. Kaan Baykan, "Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the istanbul stock exchange," *Expert Systems with Applications*, vol. 38, no. 5, pp. 5311–5319, 2011.

[18] D. Karmiani, R. Kazi, A. Nambisan, A. Shah, and V. Kamble, "Comparison of predictive algorithms: Backpropagation, svm, lstm and kalman filter for stock market," in *2019 Amity International Conference on Artificial Intelligence (AICAI)*, 2019, pp. 228–234.

[19] P. Yu and X. Yan, "Stock price prediction based on deep neural networks," *Neural Computing and Applications*, vol. 32, pp. 1609–1628, 2019.

[20] P. Gao, R. Zhang, and X. Yang, "The application of stock index price prediction with neural network," *Mathematical and Computational Applications*, vol. 25, no. 3, p. 53, 2020.