Time Series Forecasting and Stock Price: A Study of Different Models

Md. Muhtasim Mahmud
School of Data and Sciences
Brac University
Dhaka, Bangladesh
md.muhtasim.mahmud@g.bracu.ac.bd

***Abstract*—The stock market is a complicated and volatile system that requires accurate forecasting techniques in order to make smart investing decisions. In this research, we explore the effectiveness of different advanced machine learning algorithms in improving the prediction accuracy of the widely used ARIMA model. We evaluate the performance of LSTM, Simple RNN, GRU, and FB Prophet models on the stock market dataset using three widely used measures, RMSE, MAPE and MAE. In terms of RMSE, MAPE and MAPE, our results suggest that on average, the LSTM, Simple RNN and GRU models have better performance compared to the other models, including the ARIMA and Prophet model. Our findings indicate that adding deep learning techniques into classical time series analysis can increase stock market forecasting accuracy dramatically.**

***Index Terms*—Autoregressive Integrated Moving Average(ARIMA), Recurrent Neural Network(RNN), Gated Recurrent Unit(GRU), Long Short Term Memory(LSTM), Root Mean Squared Error(RMSE), Mean Absolute Percentage Error(MAPE), Mean Absolute Error(MAE)**

I. INTRODUCTION

Financial models are being used by investment companies, hedge funds, and even private investors to better analyze market behavior and execute profitable trades and investments. In the form of past stock prices and firm performance data, there is a lot of information that may be processed by machine learning algorithms. Can machine learning be used to predict the price of stocks? Investors use data analysis to create informed assumptions. In order to make a prediction, they will read the news, research the background of the company, market trends, and other relevant information. It is unclear why prestigious companies like Morgan Stanley and Citigroup pay quantitative analysts to create prediction models given the current ideas, which hold that stock prices are completely random and unpredictable. The image that comes to me is of a trading floor full of men racing about yelling into phones while high on adrenaline. Today, it is more common to see rows of machine learning specialists quietly working in front of computers. On Wall Street, software now places about 70 percent of all orders. The age of algorithms is now upon us. This study first constructed Deep Learning models(LSTM, Simple RNN, GRU, FB Prophet) to enhance the ARIMA model's initial predictions. For tech giants like Apple, Google, Tesla, Microsoft,Amazon and Yahoo stock prices are anticipated.

II. RELATED WORKS

"Automatic Time Series Forecasting: The forecast Package for R" [1] by Hyndman et al.introduces the "forecast" R package to solve time series forecasting. This work requires predicting a time-dependent variable's future values given its historical values. The authors want to design a simple, accurate automated forecasting program. The "forecast" package offers tools and functions that automatically identify the appropriate forecasting approach, calculate model parameters, and output time series forecasts. Random forests, neural networks, exponential smoothing models, ARIMA models, state space models, non-probabilistic hybrid models like ES-RNN, and probabilistic hybrid models like Gaussian processes and linear state space models are included in the package. The "forecast" program is tested on several real-world time series datasets and compared to other popular forecasting methods. The scientists say the software beats other methods and produces accurate projections with minimal user input. Scalable and computationally efficient, the program allows large-scale forecasting. "Predict" software simplifies forecasting for beginners. Automating forecasting approach selection and model parameter estimation saves researchers and practitioners time and money. Due to its ease of use, the application is useful for time series forecasting. The "predict" package is a powerful and useful tool for researchers and practitioners, contributing to time series forecasting. It simplifies forecasting for laypeople and produces trustworthy results with little user input.

"Time-series forecasting with deep learning: a survey" [2] by Lim et al. provides a detailed overview of deep learning-based time-series forecasting. Time-series forecasting is addressed by predicting future values of a time-dependent variable based on its historical values. The authors reviewed existing academic and commercial deep learning-based time-series forecasting research to reach their goal. They examined RNNs, CNNs, LSTMs, autoencoders, and other time-series forecasting deep learning architectures. Deep learning time-series forecasting was compared to ARIMA and exponential smoothing. Deep learning performed well in many applications, especially with complex and nonlinear time-series data. Deep learning-based time-series forecasting has many challenges and unmet research topics, including interpretability, model choice, and scalability. They stressed feature engineering, data preparation, and large, diverse training and testing datasets.

Financial time series forecasting using deep learning is covered in ”Financial time series forecasting with deep learning: A systematic literature review: 2005-2019.” [3] by Sezer et al. The authors want to analyze the literature on this topic, highlighting the most relevant models and methodologies used and the pros and cons of deep learning for financial forecasting. The authors’ method involves a rigorous literature review of 2005–2019 deep learning research on financial time series forecasting. They explored deep learning designs and methodologies like RNNs, CNNs, and LSTM networks. Deep learning algorithms for financial time series forecasting were praised for their ability to recognize complex and asymmetrical financial data patterns. Deep learning algorithms performed well in stock price predicting, exchange rate forecasting, and credit risk assessment. The authors also noted overfitting, interpretability, and data quality challenges with deep learning for financial forecasting. They highlighted model selection and validation and financial forecasting domain-specific expertise and experience. The authors’ paper thoroughly analyzes deep learning approaches, contributing to financial time series forecasting. Their work highlights the promise of deep learning for managing complex financial data and the importance of purposeful experimentation and subject expertise for accurate and trustworthy estimates.

”Financial time series forecasting using support vector machines,” [4] by Kim et al. a popular machine learning technique for classification and regression, addresses the subject of financial time series forecasting. For financial forecasting, the authors compare SVMs against ARIMA and exponential smoothing. SVMs were applied to stock prices and currency rates to compare their performance to conventional time series models. They used several SVM kernels and parameter settings, evaluated their performance using MSE and MAPE, and reported their findings (MAPE). SVMs performed well for financial time series forecasting in many cases, according to the authors. They noted that SVMs are resistant to overfitting and other time series modeling issues and can be effective for modeling complex and nonlinear financial data relationships. The authors note that SVMs for financial forecasting are sensitive to parameter choices, need thorough feature engineering, and require data pretreatment. They underlined the importance of financial forecasting domain-specific knowledge and abilities and SVM parameter selection and fine-tuning for optimal performance. The authors’ research evaluates SVMs for this purpose, which advances financial time series forecasting. Their work highlights the importance of comprehensive testing and model selection to generate accurate and trustworthy forecasts.

In ”Non-linear Financial Time Series Forecasting - Application to the Bel 20 Stock Market Index,” [5] by Lendasse et al. the Brussels Stock Exchange’s benchmark index is predicted. The authors intend to build a model that can accurately predict the index’s change over time. ANNs and SVMs are used to predict Bel 20 index movements by the authors. They compare these non-linear models to linear models like ARIMA models. The authors evaluate their algorithms using MSE, MAE, DA, and trading simulations. They do sensitivity analyses to determine how model parameters affect model performance. The authors claim that their SVMs outperform linear models in forecasting accuracy and directional correctness. SVMs capture non-linear correlations better than ANNs. Their models can be used for trading simulations and the SVM model outperforms linear models in return on investment. The authors found that SVMs can predict financial time series data. They underline the importance of taking non-linear relationships in data into consideration and show how their methodology can produce more accurate estimates and better trading results.

In ”Optimizing LSTM for time series prediction in the Indian stock market,” [6] by Yadav et al. the Long Short-Term Memory (LSTM) model is used to anticipate Indian stock market index movements. The authors intend to construct an LSTM model that can accurately predict NSE Nifty 50 index closing prices. The authors alter hyperparameters including the amount of neurons, learning rate, and epochs to improve the LSTM model’s forecasting accuracy. They compare the LSTM model to support vector regression and autoregressive integrated moving average (ARIMA) (SVR). The authors evaluate their models (DA) using MAE, RMSE, and directional correctness. They simulate their LSTM model’s trading profitability. The authors’ modified LSTM model surpasses competitors in forecasting accuracy with a lower MAE, RMSE, and DA. They also find that their LSTM model generates profitable trading signals and has the best cumulative return. The LSTM model’s ability to understand data’s complex non-linear correlations improves its performance. The authors’ study shows that LSTM models may be used for time series prediction in the Indian stock market and that hyperparameter adjustments improve forecasting accuracy. They establish their upgraded LSTM model works and suggest stock market traders and investors use it.

”Financial time series forecasting with machine learning techniques: A survey” [7] by Krollner et al. provides a basic review of the various machine learning methodologies used for financial time series forecasting and its pros and cons. Financial time series including stock prices, exchange rates, and commodity prices are notoriously difficult to predict due to their complexity and unpredictability. Because conventional time series models like ARIMA and exponential smoothing may not always represent financial data’s non-linear and intricate relationships, they suggest machine learning alternatives. The authors review machine learning-based financial time series forecasting literature. They classify ensemble methods, neural networks, support vector machines, and linear regression. The authors evaluate the strategies using mean absolute error, mean squared error, and directional accuracy. They discuss each technique's pros and cons and financial applications. The authors say machine learning algorithms can manage large volumes of data and capture non-linear relationships, making them attractive financial time series forecasters. They warn against overfitting and data snooping biases and note that parameter selection and data quality may affect machine learning algorithm success. The authors' research analyzes the different machine learning methods used to forecast financial time series and shows their pros and cons in the financial business.

"Fuzzy transfer learning in time series forecasting for stock market prices" [8] by Pal et al. discusses forecasting stock market prices using transfer learning and fuzzy logic. Due to the stock market's complexity and volatility, investors and financial institutions need reliable stock price forecasts. Traditional time series forecasting methods may not accurately capture stock market movements with highly coupled and nonlinear data. Transfer learning with deep learning and fuzzy logic is the authors' solution. After pre-processing and removing noise with fuzzy logic, they use a deep learning model like a recurrent neural network for time series forecasting. Transfer learning trains the deep learning model on comparable stocks or markets before fine-tuning it on the target stock or market. The authors test their technique using stock market data. They compare their transfer learning model to ARIMA and exponential smoothing time series forecasting algorithms. Their fuzzy transfer learning model is more accurate and resilient, according to the authors. Their technology captures the complex and nonlinear stock market data relationships and is less vulnerable to noise and outliers. Transfer learning and fuzzy logic improve stock market value time series forecasting, as shown by the authors' work.

"Stock Prediction Based on Optimized LSTM and GRU Models" [9] by Gao et al. uses deep learning to predict stock prices. The authors suggested a hybrid RNN model that combines long short-term memory (LSTM) and gated recurrent unit (GRU) models. The authors optimized LSTM and GRU models by modifying hyperparameters like learning rate and batch size and choosing the best number of neurons and layers. A hybrid model was created from the improved LSTM and GRU models to improve stock price forecasts. The authors compared their model to single LSTM and GRU models using historical stock price data from three Chinese companies. They assessed prediction accuracy using performance indicators like MAE, MSE, and RMSE. The hybrid LSTM-GRU model beat the solo LSTM and GRU models in MAE, MSE, and RMSE. Their model outperformed other state-of-the-art models in prediction accuracy. The authors found that the hybrid LSTM-GRU model can forecast stock values and be used in finance.

"Research on the Feasibility of Applying GRU and Attention Mechanism Combined with Technical Indicators in Stock Trading Strategies" [10] by Lee et al. examines the use of Gated Recurrent Unit (GRU) and Attention Mechanism in stock trading strategies. The authors build stock trading techniques. They propose a deep learning model that predicts stock prices and generates trading signals using GRU and Attention Mechanism with technical indicators. They preprocess data to obtain technical indicators like MACD and RSI. The GRU and Attention Mechanism layers of the deep learning model receive the preprocessed data. The authors say their algorithm outperformed technical indicator-based trading strategies. The approach generated larger returns and reduced risk than traditional tactics. The model's hyperparameters, such as the number of technical indicators and the historical data window, are sensitivity-analyzed in the study. The appropriate hyperparameters depend on the stock traded and the time period. The research shows that deep learning models that mix GRU and Attention Mechanism with technical indicators can be used for stock trading strategies and emphasizes hyperparameter selection.

"A Comparative Study of LSTM and DNN for Stock Market Forecasting" [11] by Shah et al. compares Long Short-Term Memory (LSTM) and Deep Neural Network (DNN) models for stock price forecasting. Financial markets are complex and fickle, making stock price predictions difficult. LSTM and DNN models are compared for accuracy and computational efficiency. They train each model on historical stock price data to anticipate future prices. They evaluate each model using MAE and RMSE measures. The authors report that the LSTM model outperformed the DNN model in terms of accuracy, achieving lower MAE and RMSE values. DNN outperformed LSTM in computational efficiency. The models' hyperparameters, such as the number of hidden layers and neurons per layer, were also sensitivity analyzed. The research discusses the accuracy-efficiency tradeoff of LSTM and DNN models for stock market forecasting. The results imply that the DNN model may be better for computational efficiency and the LSTM model for accuracy.

"Stock market forecasting using recurrent neural network" [12] by Gao et al. develops an RNN-based deep learning model for stock market forecasting. Due to financial market volatility, projecting stock values is difficult. They present a deep learning model using RNNs, which can detect temporal connections in sequential data. They preprocess data to extract opening and closing prices, volume, and technical indicators. The RNN model, which has several LSTM layers, receives the preprocessed data. The authors say their model outperformed ARIMA and GARCH models. The model predicted stock values and generated profitable trading signals. The model's hyperparameters, such as the number of LSTM layers and the historical data window, are sensitivity-analyzed in the paper. The appropriate hyperparameters depend on the stock traded and the time period. The research shows that RNN-based deep learning models can forecast stock markets and emphasizes hyperparameter selection. These models may help traders and investors predict stock prices and provide effective trading signals.

”Time Series Forecasting Model for Supermarket Sales using FB-Prophet” [13] by Jha et al. tries to use the FB-Prophet algorithm to anticipate supermarket sales. Predicting supermarket sales, a common retail issue, can affect inventory, staffing, and profitability. They suggest using Facebook’s open-source time series forecasting method, FB-Prophet. They preprocess sales data for seasonality and other tendencies. After preprocessing, the FB-Prophet algorithm fits a time series model and anticipates sales. The authors say their model outperformed ARIMA and exponential smoothing. The model successfully predicted grocery sales, improving inventory management and personnel. The model’s hyperparameters, such as the prior scale parameter and seasonality parameters, are sensitively analyzed in the study. The ideal hyperparameters depend on the supermarket and product. The research shows that the FB-Prophet algorithm is effective for retail time series forecasting and emphasizes the importance of seasonality and other sales data trends. The results show that store managers and other retail professionals may use such models to forecast sales and optimize operations.

”Financial Time Series Forecasting Using Prophet” [14] by Yusof et al. develops a stock price prediction model using the FB-Prophet algorithm. Financial markets are dynamic and fickle, making stock price prediction difficult. They offer an FB-Prophet-based forecasting model that has performed well in time series forecasting for several industries. Preprocessing financial data, including daily stock prices and trading volume, accounts for seasonality and other tendencies. After preprocessing, the FB-Prophet algorithm fits a time series model and anticipates stock prices. The authors say their model outperformed ARIMA. The model predicted stock values and generated profitable trading signals. The model’s hyperparameters, such as the prior scale parameter and seasonality parameters, are sensitively analyzed in the study. The appropriate hyperparameters depend on the stock traded and the time period. The study shows that the FB-Prophet algorithm can anticipate financial time series and emphasizes the importance of seasonality and other trends in financial data. These models may help traders and investors predict stock prices and provide effective trading signals.

”Forecasting Time Series Data Using ARIMA and Facebook Prophet Models” [15] by Sivaramakrishnan et al. compares the performance of the two models in predicting time series data. Time series forecasting and its limitations owing to complexity and variability are first discussed by the authors. They next explain their ARIMA and Prophet models. ARIMA is a

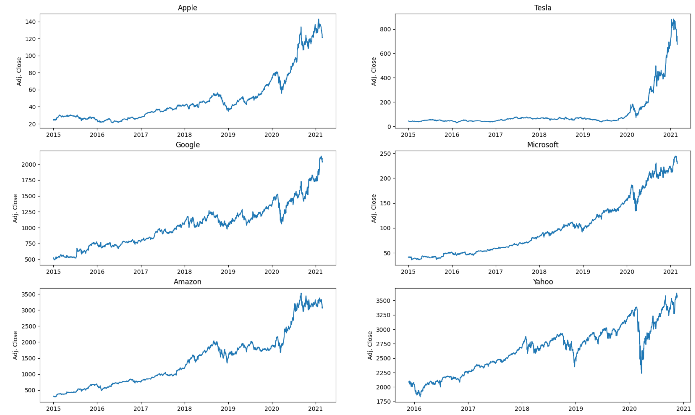

Fig. 1. Data trend of Adj. Close.

famous model that predicts by identifying patterns and trends in historical data. Its simplicity and efficacy make it a popular time series analysis model. Prophet, a newer Facebook model, accounts for seasonality, trends, and other external factors that may affect time series data. It can accommodate big datasets and incorporate variable seasons and patterns. They assess model performance using MAE, MSE, and RMSE. Prophet outperforms ARIMA in most temperature predictions. Prophet’s ability to handle seasonality and trends makes it an effective time series forecasting tool. However, ARIMA’s simplicity and use make it valuable. The study compares two common time series forecasting algorithms and shows how each predicts real-world data.

III. Research Methodology

A. Datasets and Pre-Processing

In this project, we analyzed six stock price datasets of the following companies: Apple(9800), Tesla(2686), Google(4162), Microsoft(8890), Amazon(5989) and Yahoo(1825). The datasets were containing daily stock price data from past to uptill to 2021. Before conducting our analysis, we performed data preprocessing on the datasets to ensure that they were in the appropriate format for our analysis. Specifically, we changed the datatype of the rows from object to date and from object to floats. Then we dropped the rows having data prior to the date 2015-01-01, so that we can train our model with comparatively newer datasets. Each dataset contains the following attributes:
Date: The date of the stock price.
Open: The opening price of the stock on that day.
High: The highest price the stock reached on that day.
Low: The lowest price the stock reached on that day.
Close: The closing price of the stock on that day.
Volume: The number of shares that were traded on that day.
Adj. Close: The closing price of the stock adjusted for any corporate actions such as stock splits, dividends, or rights offerings.

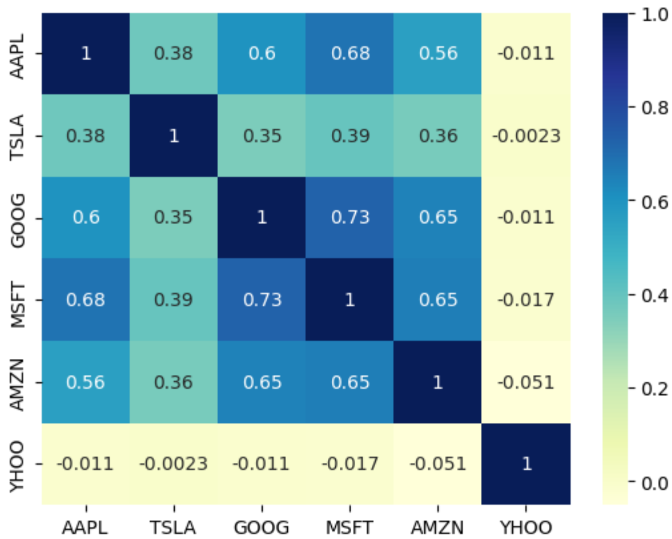

Fig. 2. Dataset Heat-map

B. Model Architecture

ARIMA: Autoregressive Integrated Moving Average (ARIMA) is a prominent time series modeling technique that predicts future values based on past data. Autoregression, differencing, and moving average make up the ARIMA model.
Autoregression (AR): AR represents the link between an observation and its own past values. AR models presume that time series values are a function of past values.
Differencing (I): To keep data steady, this component subtracts consecutive time series observations. Modeling stationary data is easier because its mean and variance remain constant.
Moving Average (MA): This component models the link between an observation and a residual error from a moving average model applied to lagged time series observations. MA models presume that a time series' value depends on its prior errors.
p, d, and q determine the ARIMA model's component order. The p, d, and q parameters indicate the autoregressive, differencing, and moving average components, respectively.
In conclusion, the ARIMA model is a powerful time series analysis tool that can discover trends and patterns and accurately predict future values. [15]

LSTM: Long Short-Term Memory (LSTM) is a recurrent neural network (RNN) architecture used to analyze sequential data like time series data. LSTM networks use memory cells and multiple gating methods to capture long-term dependencies in data, unlike typical RNNs, which suffer from the vanishing gradient problem.LSTM architecture includes:
Memory Cell: The memory cell stores and regulates network information. LSTMs update or forget their cell state based on input and previous state.
Input Gate: The input gate governs how much input and prior state information is added to the memory cell. A sigmoid activation function that outputs 0–1 controls the gate.
Forget Gate: The forget gate governs how much prior state information is erased from the memory cell. A sigmoid activation function that outputs 0–1 controls the gate.
Output Gate: This gate controls how much memory cell data is output. A sigmoid activation function that outputs 0–1 controls the gate.
Candidate State: The memory cell can store a new candidate state. -1 to 1 tanh activation functions govern the candidate state.
For sequential data processing, the LSTM architecture stores and manipulates information over time. A memory cell stores information and many gating mechanisms control information flow in the network. Time series forecasting, speech recognition, and natural language processing benefit from the LSTM design. [6]

Simple RNN: Simple recurrent neural networks (RNNs) are used to analyze sequential data like time series data and natural language. A single hidden layer of neurons connects the input and output layers in simple RNNs.
Each data point is fed successively into the Simple RNN architecture. At each time step, a weight matrix multiplies the input data and adds it to the previous output. The current time step output is produced by passing this output through a non-linear activation function like the sigmoid or hyperbolic tangent function. The network's next time step input is the current time step's output.
The Simple RNN architecture's biggest drawback is the vanishing gradient problem, which makes training on large data sequences problematic. The vanishing gradient problem happens when the error function gradients with respect to network weights become exceedingly small as they propagate back in time across the network. [12]

Gated Recurrent Unit: GRU (Gated Recurrent Unit) RNNs avoid the vanishing gradient problem of ordinary RNNs. GRU, like LSTM, has less parameters and trains faster.
GRU gates govern network information flow. The input gate controls how much new information enters the network, the forget gate controls how much old information is preserved, and the output gate controls how much information is sent to the next timestep.
A new gate, the "update gate," controls how much of the old state should be changed with the new input in GRU. GRUs can choose retain or reject information based on context, making them more versatile than RNNs.
GRU design is ideal for sequence modeling applications including language translation, audio recognition, and text production. It outperforms other state-of-the-art RNN models in several applications. [10]

FB Prophet: Prophet is Facebook Core Data Science's time series forecasting model. Business forecasting uses it because it handles time series data with numerous seasonality and trend

variations.
The Prophet model breaks time series into trend, seasonality, and holidays. Trend and seasonality describe the time series' direction and fluctuations, respectively. Holidays and promotional initiatives can affect time series.
Prophet fits models using Bayesian probabilistic models and user-specified parameters. A piecewise linear model and Fourier series model trend changes and seasonality, respectively.
Prophet also detects outliers and lets users set trend changepoints. The model also optimizes hyperparameters with a built-in technique.
Prophet is a versatile and powerful model architecture that excels at time series forecasting. Business forecasting applications include its automatic outlier detection, hyperparameter adjustment, and capacity to manage numerous seasonality and trend variations. [14]

C. Model Tweaking and Inclusion

The (p, d, q) parameter in the ARIMA model was initially set to (1, 1, 0). In order to improve our result, we changed the default parameter and used (1, 2, 2), (2, 2, 2), (3, 2, 2) and (5, 2, 2).
Additionally, we moved forward with the implementation of four distinct time series forecasting models: LSTM, Simple RNN, GRU, and FB Prophet. The Simple RNN model has two Recurrent layers and five additional dense layers. Similarly, the GRU model implementation has two Gated Recurrent layers and four extra dense layers, whereas the LSTM model has two Long Term Dependency layers and four additional dense layers. We attempted these models with "relu" activation and Adam Optimizer.
Based on their root mean squared error (RMSE), mean absolute percentage error (MAPE), and mean absolute error (MAE), each model was tested using the datasets mentioned earlier.
The results were compared to identify the model that performed the best for the given dataset after the models had been trained and evaluated. The best-performing model was the one with the lowest RMSE, MAPE, and MAE values.

IV. Results

We used six datasets from six distinct IT businesses, as we discussed earlier. We discover the following results after training and testing our five models using our datasets-

A. Apple

Our findings show that, using the provided dataset, the ARIMA, LSTM, Simple RNN, GRU, and FB Prophet models were all able to accurately anticipate the stock price of Apple company.
Based on the evaluation metrics, GRU and Simple RNN had the best performance among the models. The Simple RNN model's RMSE was 4.94, its MAPE was 0.0324, and its MAE was 4.05 while the GRU model's were 4.93, 0.306, and 3.84

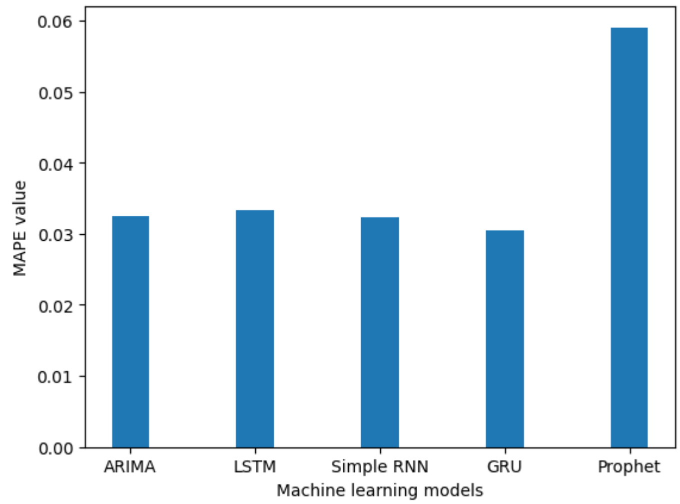

Fig. 3. MAPE representation of Apple dataset

respectively.
With an RMSE of 5.14, a MAPE of 0.0334, and an MAE of 4.18, the LSTM model likewise performed admirably. The ARIMA model was marginally less accurate than the other models, with an RMSE of 5.20, a MAPE of 0.0324, and an MAE of 4.06.
The FB Prophet model was the least accurate for this dataset because it had the greatest RMSE, MAPE, and MAE values of any model.

B. Tesla

Our findings show that, using the provided dataset, the ARIMA, LSTM, Simple RNN, GRU, and FB Prophet models were all capable of predicting the stock price of Tesla company.
Based on the evaluation metrics, GRU and Simple RNN had the best performance among the models. The Simple RNN model's RMSE was 49.43, its MAPE was 0.0563, and its MAE was 35.46, compared to the GRU model's RMSE of 46.41, MAPE of 0.0535, and MAE of 34.51.
With an RMSE of 64.23, a MAPE of 0.0782, and an MAE of 50.63, the LSTM model likewise did well. The ARIMA model was less accurate than the other models, with an RMSE of 102.60, a MAPE of 0.1264, and an MAE of 83.19.
The FB Prophet model was the least accurate for this dataset because it had the greatest RMSE, MAPE, and MAE values of any model.

C. Google

Our findings show that, using the provided dataset, the ARIMA, LSTM, Simple RNN, GRU, and FB Prophet models were all able to predict the stock price of Google.
Based on the evaluation metrics, GRU and LSTM had the best performance among the models. The LSTM model had an RMSE of 62.02, a MAPE of 0.0245, and an MAE of 46.37 compared to the GRU model's RMSE of 59.22, 0.0228 MAPE,

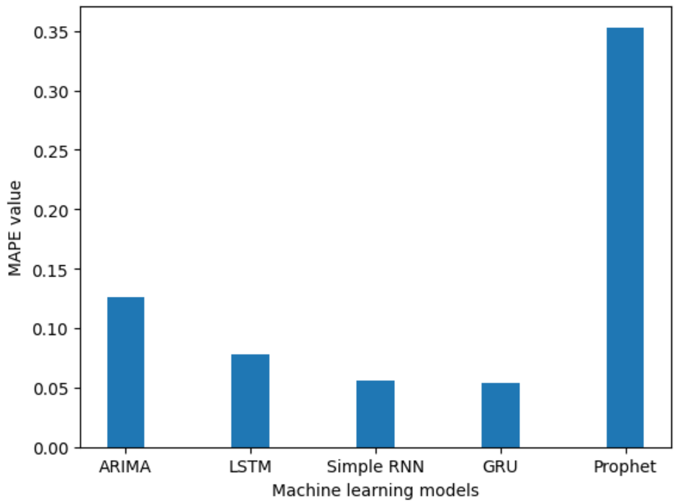

Fig. 4. MAPE representation of Tesla dataset

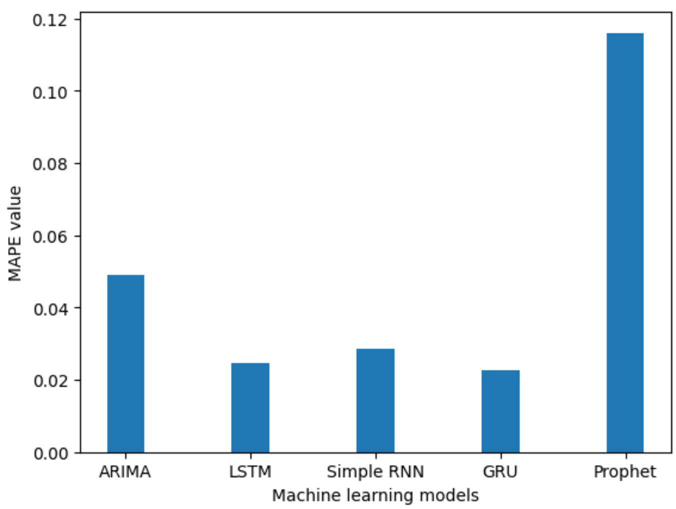

Fig. 5. MAPE representation of Google dataset

and 43.43 MAE.
With an RMSE of 74.11, a MAPE of 0.0285, and an MAE of 54.36, the Simple RNN model likewise fared reasonably well. The ARIMA model was less accurate than the deep learning models, with an RMSE of 116.70, a MAPE of 0.0491, and an MAE of 93.85.
The FB Prophet model was the least accurate for this dataset because it had the greatest RMSE, MAPE, and MAE values of any model.

D. Microsoft

Our findings show that, using the provided dataset, the ARIMA, LSTM, Simple RNN, GRU, and FB Prophet models were all able to predict the stock price of Microsoft company. Based on the evaluation metrics, LSTM and GRU had the best performance among the models. The RMSE of the LSTM model was 6.04, the MAPE was 0.02, and the MAE was 4.54

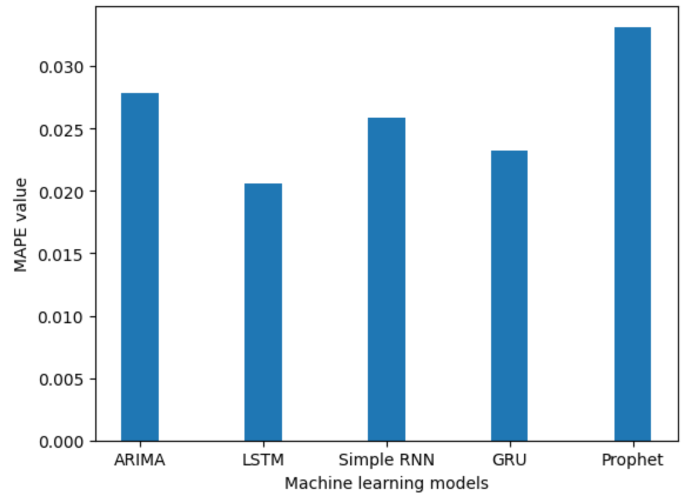

Fig. 6. MAPE representation of Microsoft dataset

while the RMSE of the GRU model was 6.54, the MAPE was 0.02, and the MAE was 5.10.
With an RMSE of 7.10, a MAPE of 0.03 and an MAE of 5.68, the Simple RNN model likewise performed admirably. The ARIMA model was less accurate than the other models, with an RMSE of 7.32, a MAPE of 0.03, and an MAE of 6.12.
The FB Prophet model was the least accurate for this dataset because it had the greatest RMSE, MAPE, and MAE values of any model.

E. Amazon

Our findings show that, using the provided dataset, the ARIMA, LSTM, Simple RNN, GRU, and FB Prophet models were all able to predict the stock price of Amazon company.
Based on the evaluation metrics, Simple RNN and LSTM had the best performance among the models. The LSTM model had an RMSE of 87.47, a MAPE of 0.0225, and an MAE of 71.84 compared to the Simple RNN model's RMSE of 84.64, 0.0210, and 66.78.
With an RMSE of 92.08, a MAPE of 0.0223, and an MAE of 70.90, the GRU model likewise performed admirably. The ARIMA model was less accurate than the other models, with an RMSE of 113.16, a MAPE of 0.0250, and an MAE of 79.71.
The FB Prophet model was the least accurate for this dataset because it had the greatest RMSE, MAPE, and MAE values of any model.

F. Yahoo

Our findings show that, using the provided dataset, the ARIMA, LSTM, Simple RNN, GRU, and FB Prophet models were all able to predict the stock price of Yahoo company.
Based on the evaluation metrics, Simple RNN and LSTM had the best performance among the models. The LSTM model has an RMSE of 170.88, a MAPE of 0.0365, and an MAE of 109.57 compared to the Simple RNN model's RMSE of

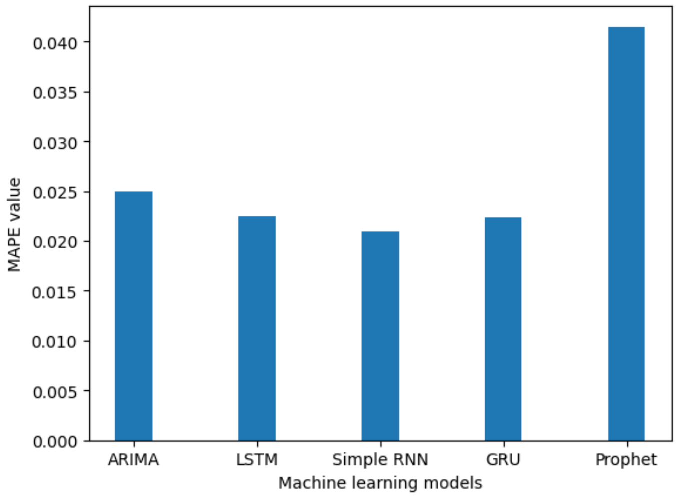

Fig. 7. MAPE representation of Amazon dataset

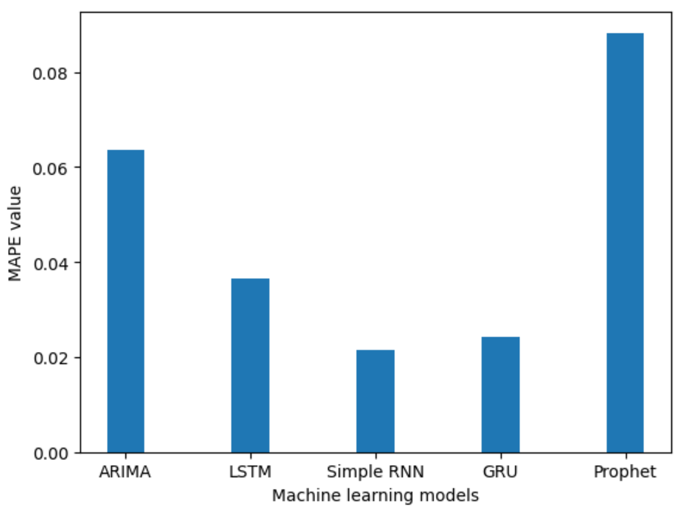

Fig. 8. MAPE representation of Yahoo dataset

95.36, 0.0215, and 64.42.

With an RMSE of 105.11, a MAPE of 0.0242, and an MAE of 74.65, the GRU model likewise performed admirably. The ARIMA model was less accurate than the other models, with an RMSE of 255.78, a MAPE of 0.0636, and an MAE of 184.47.

The FB Prophet model was the least accurate for this dataset because it had the greatest RMSE, MAPE, and MAE values of any model.

G. Discussion

We can see from the results section that, out of the five models, the LSTM, Simple RNN, and GRU models gave the results that were the most accurate, whilst the ARIMA and FB Prophet models produced the results that were the least accurate. As a result, of the five models, ARIMA and FB Prophet performed the worst on our datasets.

For instance, the Yahoo dataset was the noisiest dataset since

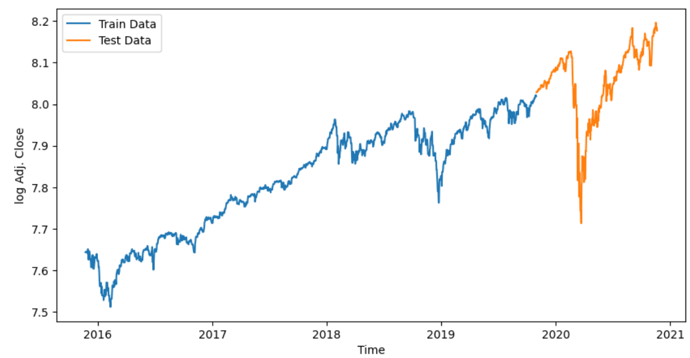

Fig. 9. Data-trend of Yahoo dataset

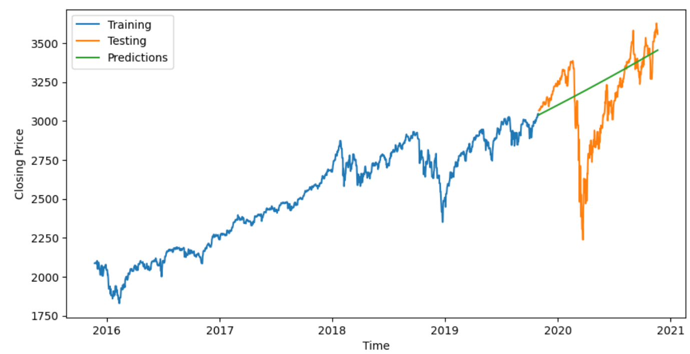

Fig. 10. Output of ARIMA model for Yahoo dataset

the Adj. Close values showed sharp increase and decrease trends. The results from the LSTM, Simple RNN, and GRU models were quite good for this dataset. However, ARIMA and FB Prophet were significantly underperforming. Therefore, we can conclude that RNN-based ML models are more appropriate for our datasets.

V. Conclusion

Based on the project's findings, it can be said that deep learning models like LSTM, Simple RNN, and GRU are superior to models like ARIMA and FB Prophet in forecasting stock values. All six stock price datasets for Apple, Tesla,

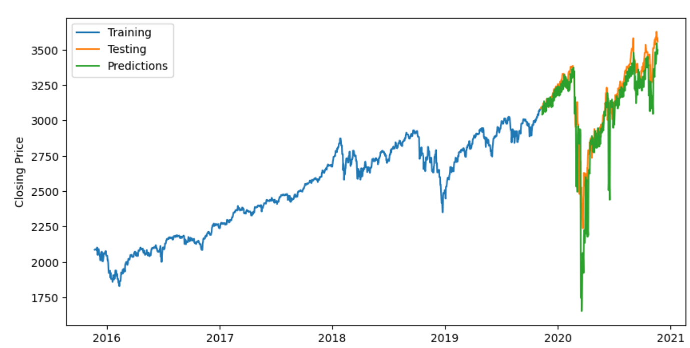

Fig. 11. Output of LSTM model for Yahoo dataset

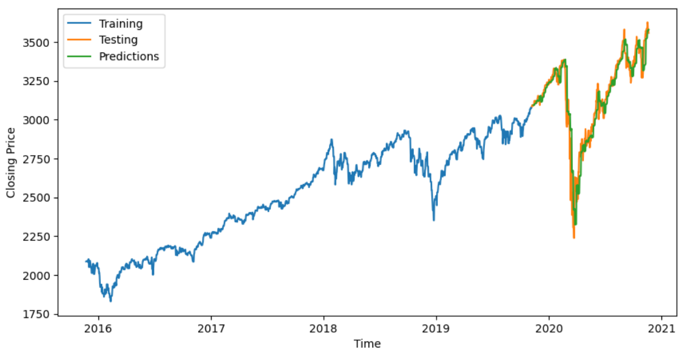

Fig. 12. Output of Simple RNN model for Yahoo dataset

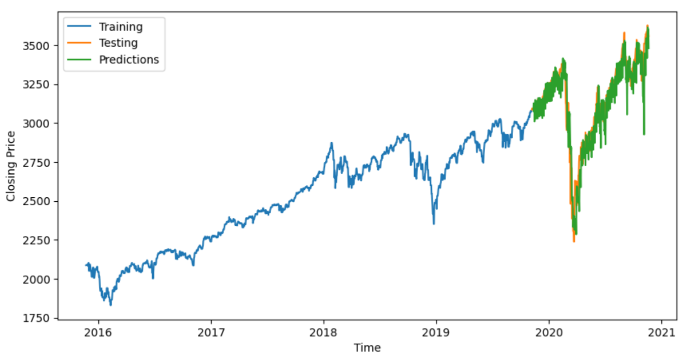

Fig. 13. Output of GRU model for Yahoo dataset

Google, Microsoft, Amazon, and Yahoo had consistent results. The Simple RNN model was discovered to be the most accurate deep learning model, with the lowest RMSE, MAPE, and MAE values for the majority of datasets. For the datasets, it was also discovered that the LSTM and GRU models were quite accurate. The ARIMA and FB Prophet models, on the other hand, fared poorly, demonstrating that they are inadequate for our datasets.

It is crucial to remember that stock prices are notorious for their volatility and erratic behavior, which can make precise forecasting difficult. As a result, it is advised that the project's

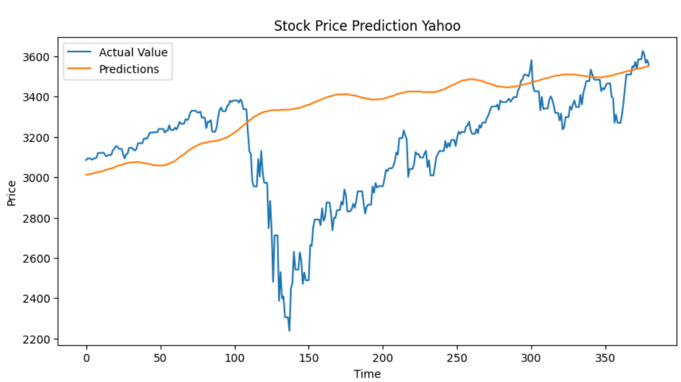

Fig. 14. Output of FB Prophet model for Yahoo dataset

outcomes be evaluated cautiously because they might not necessarily be representative of performance in the future.

In terms of potential future developments, adding additional pertinent attributes to the dataset might be one method to raise the models' accuracy. Examples of additional information that could be added to the models to help enhance their accuracy include news sentiment, market trends, and financial indicators.

Convolutional neural networks (CNNs) and other deep learning models and methods may potentially be worth investigating. Finally, more thorough testing on a wider range of stock prices over longer time periods could assist confirm the models' efficacy.

ACKNOWLEDGMENT

Without the assistance and support of numerous people, this study project would not have been able to be finished. We really appreciate the contributions made by everyone who helped make this initiative a success.

We appreciate Mr. Meem Arafat Manab sir and Mr. Jayanta Jyoti Mondal sir's important contributions and assistance during the study process. Their knowledge and expertise were crucial in determining the project's course.

Moreover, we would want to express our gratitude to all of the study participants who kindly gave of their time, experiences, and insights. Their openness to participate in our research was crucial to its success, and we are sincerely appreciative of their involvement.

REFERENCES

[1] R. J. Hyndman and Y. Khandakar, "Automatic time series forecasting: the forecast package for r," *Journal of statistical software*, vol. 27, pp. 1–22, 2008.
[2] B. Lim and S. Zohren, "Time-series forecasting with deep learning: a survey," *Philosophical Transactions of the Royal Society A*, vol. 379, no. 2194, p. 20200209, 2021.
[3] O. B. Sezer, M. U. Gudelek, and A. M. Ozbayoglu, "Financial time series forecasting with deep learning: A systematic literature review: 2005–2019," *Applied soft computing*, vol. 90, p. 106181, 2020.
[4] K.-j. Kim, "Financial time series forecasting using support vector machines," *Neurocomputing*, vol. 55, no. 1-2, pp. 307–319, 2003.
[5] A. Lendasse, E. de Bodt, V. Wertz, and M. Verleysen, "Non-linear financial time series forecasting-application to the bel 20 stock market index," *European Journal of Economic and Social Systems*, vol. 14, no. 1, pp. 81–91, 2000.
[6] A. Yadav, C. Jha, and A. Sharan, "Optimizing lstm for time series prediction in indian stock market," *Procedia Computer Science*, vol. 167, pp. 2091–2100, 2020.
[7] B. Krollner, B. J. Vanstone, G. R. Finnie *et al.*, "Financial time series forecasting with machine learning techniques: a survey." in *ESANN*, 2010.
[8] S. S. Pal and S. Kar, "Fuzzy transfer learning in time series forecasting for stock market prices," *Soft Computing*, vol. 26, no. 14, pp. 6941–6952, 2022.
[9] Y. Gao, R. Wang, and E. Zhou, "Stock prediction based on optimized lstm and gru models," *Scientific Programming*, vol. 2021, pp. 1–8, 2021.
[10] M.-C. Lee, "Research on the feasibility of applying gru and attention mechanism combined with technical indicators in stock trading strategies," *Applied Sciences*, vol. 12, no. 3, p. 1007, 2022.
[11] D. Shah, W. Campbell, and F. H. Zulkernine, "A comparative study of lstm and dnn for stock market forecasting," in *2018 IEEE international conference on big data (big data)*. IEEE, 2018, pp. 4148–4155.
[12] Q. Gao, "Stock market forecasting using recurrent neural network," Ph.D. dissertation, University of Missouri–Columbia, 2016.

[13] B. K. Jha and S. Pande, "Time series forecasting model for supermarket sales using fb-prophet," in *2021 5th International Conference on Computing Methodologies and Communication (ICCMC)*. IEEE, 2021, pp. 547–554.
[14] U. K. Yusof, M. N. A. Khalid, A. Hussain, and H. Shamsudin, "Financial time series forecasting using prophet," in *Innovative Systems for Intelligent Health Informatics: Data Science, Health Informatics, Intelligent Systems, Smart Computing*. Springer, 2021, pp. 485–495.
[15] S. Sivaramakrishnan, T. F. Fernandez, R. Babukarthik, and S. Premalatha, "Forecasting time series data using arima and facebook prophet models," in *Big data management in Sensing*. River Publishers, 2022, pp. 47–59.