

Comparative Analysis of Machine Learning Models for Stock Price Prediction: Leveraging LSTM for Real-Time Forecasting

Bijay Gautam, Sanif Kandel, Manoj Shrestha, Shrawan Thakur*

Department of Computing, Softwarica College of IT and E-Commerce, Kathmandu, Nepal
Email: *bijay.gautam385@gmail.com

How to cite this paper: Gautam, B., Kandel, S., Shrestha, M. and Thakur, S. (2024) Comparative Analysis of Machine Learning Models for Stock Price Prediction: Leveraging LSTM for Real-Time Forecasting. *Journal of Computer and Communications*, 12, 52-80.

<https://doi.org/10.4236/jcc.2024.128004>

Received: June 2, 2024

Accepted: August 19, 2024

Published: August 22, 2024

Copyright © 2024 by author(s) and Scientific Research Publishing Inc.
This work is licensed under the Creative Commons Attribution-NonCommercial International License (CC BY-NC 4.0).
<http://creativecommons.org/licenses/by-nc/4.0/>



Open Access

Abstract

The research focuses on improving predictive accuracy in the financial sector through the exploration of machine learning algorithms for stock price prediction. The research follows an organized process combining Agile Scrum and the Obtain, Scrub, Explore, Model, and iNterpret (OSEMN) methodology. Six machine learning models, namely Linear Forecast, Naive Forecast, Simple Moving Average with weekly window (SMA 5), Simple Moving Average with monthly window (SMA 20), Autoregressive Integrated Moving Average (ARIMA), and Long Short-Term Memory (LSTM), are compared and evaluated through Mean Absolute Error (MAE), with the LSTM model performing the best, showcasing its potential for practical financial applications. A Django web application “Predict It” is developed to implement the LSTM model. Ethical concerns related to predictive modeling in finance are addressed. Data quality, algorithm choice, feature engineering, and preprocessing techniques are emphasized for better model performance. The research acknowledges limitations and suggests future research directions, aiming to equip investors and financial professionals with reliable predictive models for dynamic markets.

Keywords

Stock Price Prediction, Machine Learning, LSTM, ARIMA, Mean Squared Error

1. Introduction

Investors engage in the buying and selling of ownership interests, securities, and company shares on the stock market, a dynamic and evolving marketplace that is a key component of the larger equity capital markets, facilitating the issuance

and trading of shares in publicly listed companies. Companies go public to raise capital for expansion or debt repayment by issuing stocks on stock exchanges, allowing investors to become shareholders and benefit from the company's growth and profits. The stock market's significance has led to a strong interest in accurate market predictions and trend evaluations, helping participants make informed decisions. This interest has grown, attracting individual traders, market participants, and data analysts focusing on machine learning and artificial intelligence, all aiming to gain insights into market movements and make well-informed financial decisions [1].

This research, which is being conducted against the backdrop of a growing interest in the topic, aims to investigate and comprehend the intricate dynamics of stock markets. Trading on the stock market and investing in the stock market both come with certain inherent risks since share values are affected by a wide variety of factors. These factors include the performance of a company's revenue and sales, as well as external influences such as governmental regulations, microeconomic indicators, and the ever-changing relationship between supply and demand. Because of the complexity of these factors, it is necessary to conduct extensive research to design and develop software and programs that make use of advanced learning techniques such as artificial intelligence, machine learning, deep learning, and neural networks [2]. In this way, these forecasts prove to be of tremendous value because they reduce the likelihood of monetary loss while simultaneously increasing the possibility of monetary gain because of the investment process. Traders can use these algorithms to engage in algorithmic trading strategies, stock forecasting, and the establishment of their trading setups with target prices and stop-loss levels. This research aims to enhance predictive accuracy in the financial sector by exploring the application of machine learning algorithms for stock price prediction. Accurate stock price forecasting is critical for investors, financial analysts, and other market participants as it informs investment strategies and decision-making processes. To achieve this, the study employs a structured approach that integrates Agile Scrum methodologies with the OSEMN (Obtain, Scrub, Explore, Model, and iNterpret) framework as shown in **Figure 1**.

This combination ensures a flexible and iterative development process while maintaining a systematic and comprehensive approach to data analysis and model development. The research evaluates the performance of six distinct algorithms: Linear Forecast, Naive Forecast, Simple Moving Average (SMA) with windows of 5 and 20, ARIMA (Autoregressive Integrated Moving Average), and Long Short-Term Memory (LSTM) with a 30-day prediction horizon. These algorithms are assessed using the Mean Absolute Error (MAE) metric, providing a robust measure of their predictive accuracy [3]. By comparing these models, the research seeks to identify the most effective machine learning techniques for stock price prediction, ultimately contributing to more reliable and actionable financial insights.

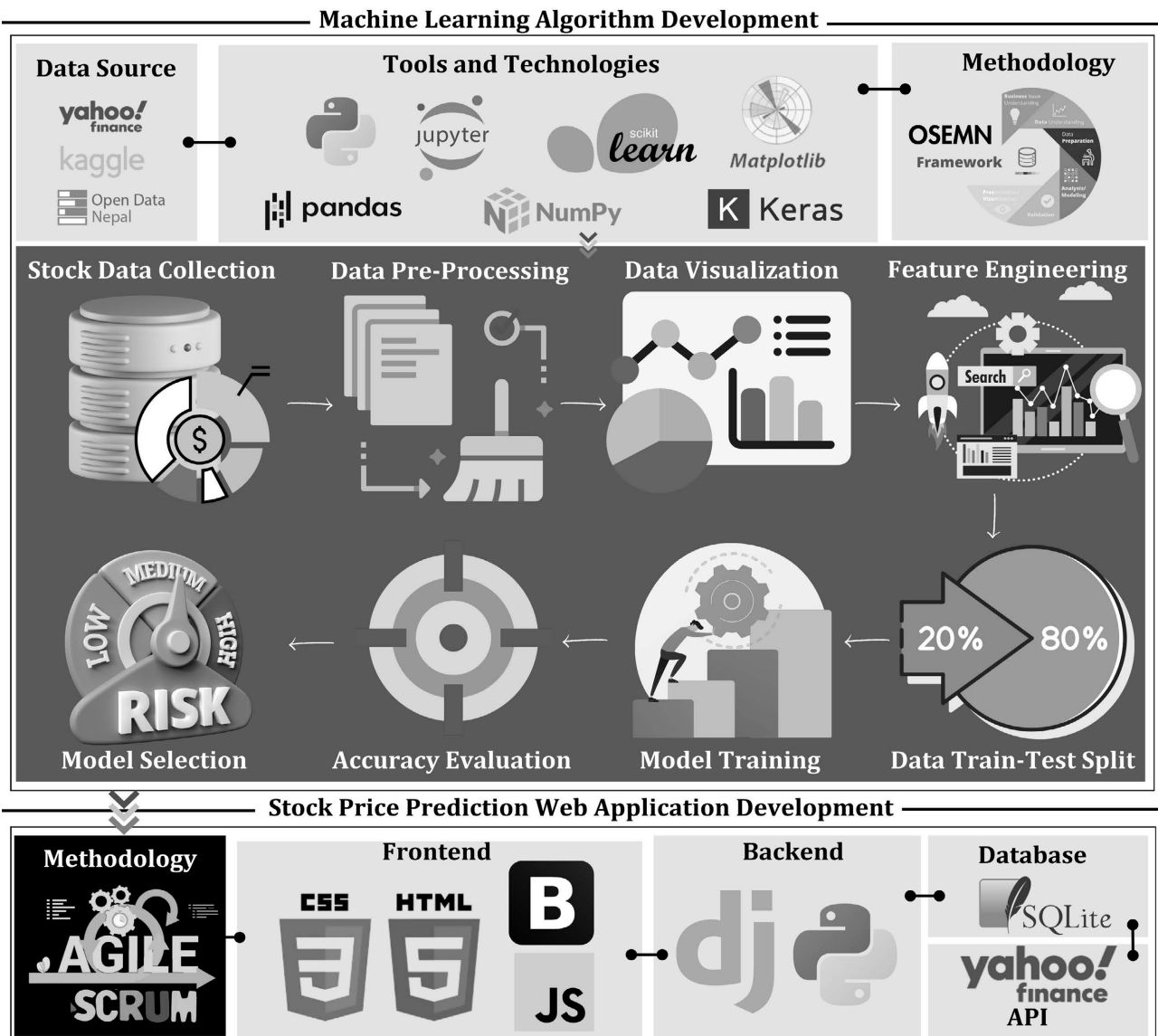


Figure 1. Integration of methodologies for development.

1.1. Justification

1.1.1. Problem Statement

In the volatile stock market, traders and financial organizations have many obstacles in predicting stock values. This complex financial landscape makes accurate predictions difficult. Market instability and uncertainty make stock price predictions challenging. Economic, political, technical, and investor mood all affect the stock market. Stock price changes are complicated by these various and unpredictable influences. Financial data is enormous and diversified, complicating the issue. Market indicators, news stories, social media sentiment, business financial statements, and historical price data fill the stock market. Data comes in many types and structures, making integration and processing difficult. Meaningful insights and powerful prediction models depend on data quality and

consistency [4]. However, the sheer volume and variety of financial data make it difficult to sort through the noise and discover relevant information, leaving traders desiring effective methods and tools to make sense of it. The complex and non-linear interactions between stock price variables are another difficulty. In the stock market, variables interact non-linearly, affecting stock prices. Such nuanced linkages challenge the basic assumptions of typical linear models, requiring more advanced methods to identify these hidden patterns [5].

Another challenge is determining trade entry and departure sites. Trading at the right time maximizes profits and minimizes losses. Price fluctuations and external factors make entry and exit positions challenging for traders. Traders can't confidently and precisely execute transactions without appropriate tools and methods. Trading in fast-changing markets requires real-time data analysis. Traders need real-time data and insights to make smart judgments. Traditional manual analysis methods may not process and analyze real-time data quickly, putting traders at a disadvantage in the dynamic market. Market participants endure manual prediction errors. Manual predictions are biased and inaccurate due to the stock market's complexity and many influences. Predictions based on gut impressions or biases might lead to inconsistent results and ineffective trading tactics [6]. Without dependable analytical tools and objective procedures, traders may struggle to anticipate and maintain profits. Investment bias should be considered when predicting stock values. Preconceived beliefs or emotional biases may influence traders' financial decisions.

1.1.2. Proposed Solution

Using advanced machine learning methods, this research project aims to address the complexities of stock market analysis and enhance predictive accuracy. By analyzing historical price data, the research seeks to uncover underlying patterns and trends, providing vital insights into market behavior. The heterogeneity and complexity of financial data, which includes fundamental indicators and social media sentiment, present significant challenges [7]. To overcome these obstacles, the study employs robust data processing techniques, such as Long Short-Term Memory (LSTM) models, which can isolate significant signals from background noise. This research evaluates six distinct models: Linear Forecast, which applies a straightforward linear approach to future price prediction; Naive Forecast, which assumes that future prices will be the same as the most recent prices; Simple Moving Average (SMA) with windows of 5 and 20, which smoothens price data to identify trends; ARIMA (Autoregressive Integrated Moving Average), which is a comprehensive time series model combining autoregression, differencing, and moving average components; and LSTM (Long Short-Term Memory) with a 30-day prediction horizon, known for capturing temporal correlations and complex patterns in sequential data. These models are assessed using the Mean Absolute Error (MAE) metric to determine their predictive accuracy. By identifying the key variables driving stock price fluctuations and auto-

inating calculations, this research aims to provide traders with crucial information to inform their investment decisions, allowing them to focus on other strategic aspects of their investment methods.

The typical process for creating a machine learning model to predict stock prices includes several stages: gathering historical data through an API, pre-processing the data, developing a forecasting model, and assessing the model's performance. During pre-processing, zero values are removed, duplicates are eliminated, and features are scaled. Key features are then selected, and valid data is chosen for predicting or forecasting stock prices [8]. This article examines various popular machine learning and deep learning algorithms, such as linear regression, moving average, naive Bayes, ARIMA (autoregressive integrated moving average), and LSTM (long short-term memory). The mean absolute error (MAE) is used to evaluate the performance of the regression or price forecasting models.

1.2. Research Aim and Objective

The research aims to build six distinct machine learning models leveraging historical stock prices and assess their predictive accuracy using the Mean Absolute Error (MAE) metric. Through a comprehensive literature review, existing studies, research papers, and scholarly articles will be explored to identify methodologies and evaluate their effectiveness in predicting stock prices. By addressing limitations and proposing solutions for algorithmic prediction, the study seeks to enhance the efficacy of these models. Clean and structured datasets will be prepared to facilitate model training and evaluation. Subsequently, the most accurate model identified will be implemented into a user-friendly web application, forming the foundation of an accurate full-stack price prediction system.

1.3. Research Questions

- 1) How does the accuracy of different machine learning algorithms compare in predicting stock prices?
- 2) What are the key factors that significantly influence the performance of machine learning algorithms in stock price prediction?
- 3) What are the ethical considerations and dilemmas associated with using machine learning algorithms for stock price prediction?

2. Materials and Methods

2.1. Materials

2.1.1. Data Collection

The research used different datasets for different purposes. Historical financial data for NABIL bank's stock was obtained from the NpseAlpha platform, while real-time and historical data for various stocks were fetched using the Yahoo Finance API. NABIL dataset was primarily used for training and testing the models while Yahoo Finance API was used to fetch real-time stock data for web ap-

plications.

2.1.2. NepseAlpha

NepseAlpha consists of all the stock data listed in NEPSE which is the stock index of Nepal. The study uses the NABIL dataset which is a banking stock for the model training and testing purpose [9]. The dataset contains date, open price, high price, low price, close price, percent change, and volume for the stock as sample shown in **Table 1**.

Table 1. Data table.

Date	Open	High	Low	Close	Percent Change	Volume
2013-07-22	1910.0	1915.0	1885.0	1885.0	-1.31	1031
2013-07-23	1885.0	1920.0	1870.0	1920.0	1.86	898
2013-07-24	1920.0	1900.0	1843.0	1890.0	-1.56	2886
2013-07-25	1890.0	1905.0	1880.0	1889.0	-0.05	1962

2.1.3. Yahoo Finance

As there is no API that allows real-time data integration of NEPSE, Yahoo Finance was used to fetch real-time stock data for foreign stocks for the web application [10]. It consists of date, open price, high price, low price, close price, volume, dividends, and stock splits as sample shown in **Table 2**.

Table 2. Data table.

Date	Open	High	Low	Close	Volume	Dividends	Stock Splits
2023-07-24	193.410004	194.910004	192.250000	192.750000	45377800	0.0	0.0
2023-07-25	193.330002	194.440002	192.919998	193.619995	37283200	0.0	0.0
2023-07-26	193.669998	195.639999	193.320007	194.500000	47471900	0.0	0.0
2023-07-27	196.020004	197.199997	192.550003	193.220001	47460200	0.0	0.0
2023-07-28	194.669998	196.630005	194.139999	195.830002	48254600	0.0	0.0

2.2. Methods

2.2.1. Data Processing and Analysis

The project utilized various tools and technologies to process and analyze data for stock price prediction. Python was used for data collection and storage, with its versatility and libraries making it ideal for data manipulation and analysis tasks. NumPy was used for data manipulation and analysis, while Pandas was used for data cleaning and preprocessing. The structured and cleaned data were prepared for further analysis stages [11]. For feature engineering, Python libraries such as NumPy and Pandas were again instrumental in creating new features from the raw data. Features like day return, log day return, and weekday were included as shown in **Table 3**.

Table 3. Data table.

Date	Open	High	Low	Close	Percent Change	Volume	Day Return	Log Day Return	Weekday
2013-07-23	1885.0	1920.0	1870.0	1920.0	1.86	898	0.0186	0.0184	Tuesday
2013-07-24	1920.0	1900.0	1843.0	1890.0	-1.56	2886	-0.0156	-0.0157	Wednesday
2013-07-25	1890.0	1918.0	1880.0	1905.0	0.85	2279	0.0085	0.0084	Thursday
2013-07-29	1905.0	1910.0	1890.0	1900.0	-0.26	2562	-0.0026	-0.0026	Monday

The exploratory data analysis (EDA) process was carried out using Python’s data visualization libraries, Matplotlib, and Seaborn. These libraries allowed for the creation of informative visualizations, enabling a deeper understanding of the data distribution, and identifying potential correlations between features and stock price movements as shown in **Figures 2-4**.

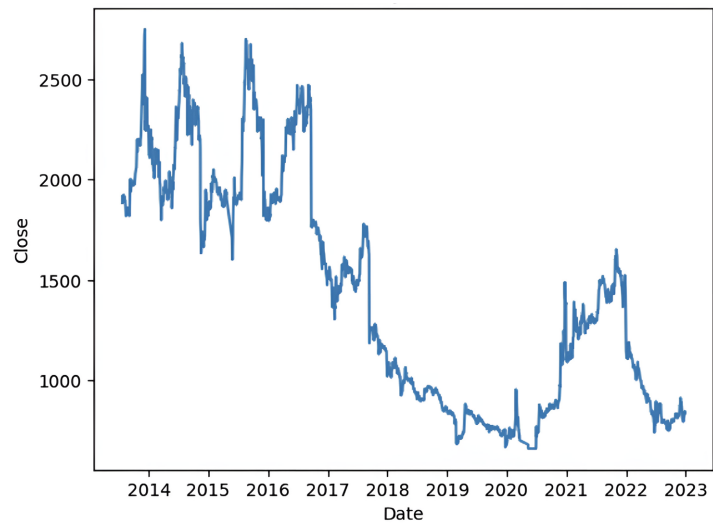


Figure 2. NABIL dataset daily close price.

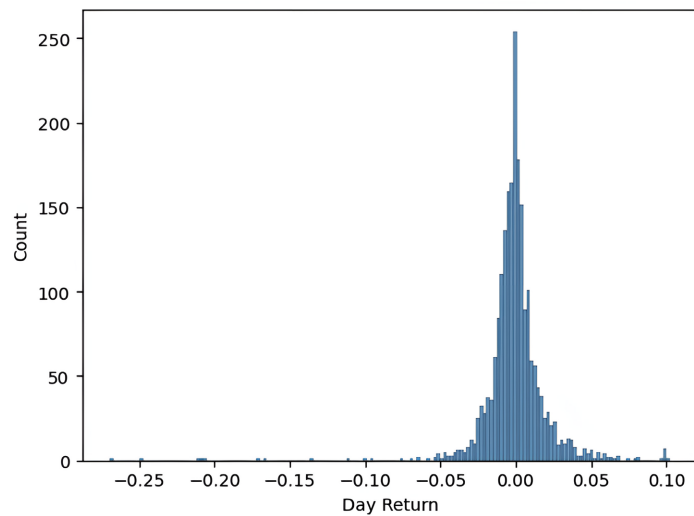


Figure 3. NABIL dataset daily return distribution.

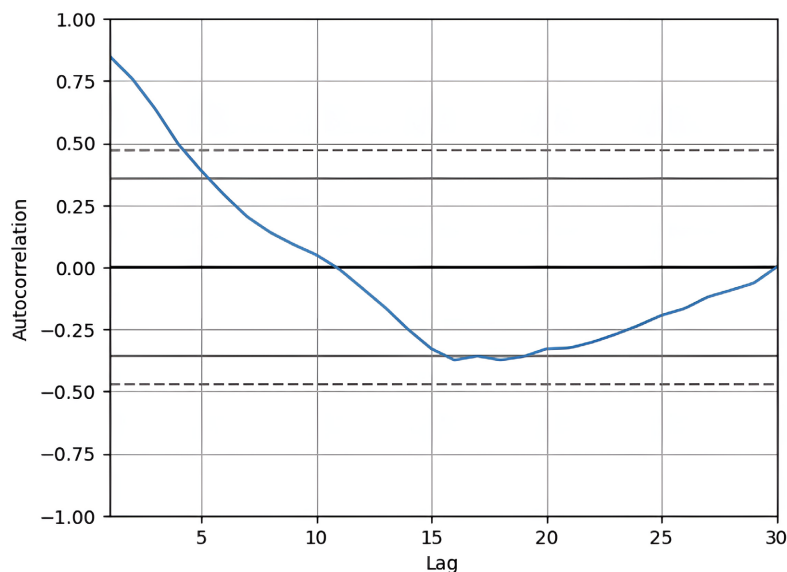


Figure 4. Auto-correlation plot for NABIL dataset.

The combination of these tools and technologies ensured a comprehensive and efficient data phase. By leveraging the power of Python and its libraries, the project was able to process and analyze large volumes of financial data from NpseAlpha and Yahoo Finance API effectively. The resulting clean and structured dataset, enriched with engineered features and insights from EDA, laid a solid groundwork for the subsequent stages of machine learning model development and web application creation [12].

2.2.2. Building Models

In the model-building phase of the project, the primary objective was to develop, evaluate, and compare six distinct machine learning algorithms for stock price prediction. These algorithms, namely Naive, Simple Moving Average (SMA 5 with a 5 day trading week window), Simple Moving Average (SMA 20 with a 20 day trading month window), Linear Regression, Autoregressive Integrated Moving Average (ARIMA) model, and Long Short-Term Memory (LSTM) model were carefully selected based on their suitability for time-series prediction tasks [13]. Python's powerful libraries, Scikit-learn, TensorFlow, and Keras, were used to implement and train machine learning algorithms. Scikit-learn provided an extensive collection of machine learning models, while TensorFlow and Keras offered a robust platform for developing and configuring deep learning models like LSTM. The machine learning process began with preparing a dataset divided into training and testing sets. Each algorithm was meticulously configured by setting appropriate hyperparameters, such as window size, epochs, learning rates, and activation functions, to ensure accurate predictions and capture underlying patterns in stock price data [14]. The Mean Absolute Error (MAE) metric was used to evaluate the performance of each algorithm. An iterative and interactive approach was adopted during the experimentation process using Jupy-

ter Notebook, allowing for easy modification of hyperparameters and rapid comparison of the algorithm's performance on the dataset.

The machine learning models were primarily tested on the NABIL dataset. Testing the algorithms on multiple datasets provided valuable insights into their generalizability and performance across different stocks and market conditions. After a comprehensive evaluation, the best-performing model was selected for further refinement and integration into the web application for stock price prediction. Techniques like hyperparameter tuning and cross-validation were applied to optimize the model's performance and ensure its ability to generalize well to unseen stock price data [15]. The rigorous evaluation and comparison of the machine learning algorithms provided valuable insights into their strengths and weaknesses for stock price prediction. The selected best-performing model served as the foundation for the successful integration of the machine-learning component into the web application.

2.2.3. Naive Model

In our research, we delve into the efficacy of the naive forecast method within our predictive modeling framework. With our dataset spanning nearly 2200 daily observations, we meticulously partition it into training, validation, and test sets to evaluate model performance. The naive forecast, aptly named for its straightforward approach of using the previous day's price as the prediction for the next day, proves surprisingly effective owing to inherent auto-correlation [16]. This method capitalizes on the close linkage between tomorrow's market open price and today's closing price.

2.2.4. Simple Moving Average Model

In our study, we embark on an exploration of the Simple Moving Average (SMA) models, specifically the SMA 5 and SMA 20 variants, within the context of financial data analysis. With our dataset comprising nearly 2200 daily observations, we meticulously partition it into training, validation, and test sets to assess the performance of our models. The SMA, a foundational tool in technical analysis, serves to smooth out data and discern underlying trends by calculating the average price or value over a specific period. Unlike the more sophisticated Exponential Moving Average (EMA), which assigns varying weights to each time step, SMA treats all data points equally within the specified window [17]. Our analysis extends to both the SMA 5 and SMA 20 models, each tailored to capture different temporal trends. While the SMA 5 focuses on weekly fluctuations, reflecting the short-term dynamics of the market, the SMA 20 provides insights into monthly trends, offering a broader perspective on price movements. These moving averages not only facilitate trend identification but also serve as vital tools for generating trading signals. For instance, the crossover of the SMA 5 over the SMA 20 signifies a recent uptrend in prices, providing traders with valuable momentum-based insights for decision-making. The major parameters implemented are mentioned in **Table 4**.

Table 4. Simple Moving Average (SMA) model parameters.

Model	Input Size	Data Frequency
SMA 5	5	Daily
SMA 20	20	Daily

2.2.5. Linear Forecast

Linear forecasting in machine learning involves predicting future values based on past data using a simplified linear relationship. This method is akin to the familiar equation $y = mx + b$, where y represents the predicted output, x is the input, m is the slope (weight), and b is the intercept (bias). The core principle behind linear forecasting is to compute the output as the sum of the products of input features and their corresponding weights, plus a bias term. Mathematically, this is expressed as:

$$y = \sum_{i=1}^n w_i x_i + b$$

where w_i are the weights, x_i are the input features, b is the bias term, and n is the number of input features [18].

To apply linear forecasting in the context of stock price prediction, we first divide historical data into sequences of input-output pairs. Each input sequence comprises past stock prices, while the corresponding output is the next day's price. These sequences are then used to train the linear forecasting model, adjusting the weights and biases to minimize the difference between predicted and actual values [19]. In our scenario, we aim to forecast future stock prices using machine learning. To do this, we train the model to predict the next step based on the previous 30 steps. This translates to creating a dataset consisting of 20-step windows for training, as each 20 steps represents one trading month, considering that markets are operational Sunday through Thursday, totaling 20 days a week. The major parameters implemented are mentioned in **Table 5**.

Table 5. Parameters for linear forecast model.

Parameter	Value
Learning Rate	1e-3
Input Size	20
Hidden Size	1
Number of Layers	1
Number of Epochs	500

2.2.6. ARIMA

To begin with, we assess the stationarity of our data. We conducted the Augmented Dickey-Fuller test for this purpose as shown in **Figure 5** and **Table 6**. The obtained p-value as exceeds the significance level of 0.05 and the ADF statistic surpasses all critical values, we fail to reject the null hypothesis, indicating

non-stationarity in the time series.



Figure 5. Dickey-Fuller test plot.

Table 6. Results of dickey-fuller test.

Test Statistic	-1.411910
p-value	0.576553
#Lags Used	4
Number of Observations Used	2140
Critical Value (1%)	-3.433409
Critical Value (5%)	-2.862892
Critical Value (10%)	-2.567489

Consequently, we proceed to convert the data to a stationary format using differencing, subtracting the value at time $(t - 1)$ from the value at time (t) to compute the daily return. Once again, we perform the ADF test as shown in Figure 6 and Table 7, and the p-value is less than 0.05 and the ADF statistic is lower than the critical values, we reject the null hypothesis, confirming the stationarity of the time series [20].

Moving forward, we delve into auto-correlation and partial auto-correlation analysis, pivotal in time series forecasting. Auto-correlation gauges the correlation between observations at time t (P_t) and those at time $t - 1$ (P_{t-1}), while partial auto-correlation assesses the correlation between time t (P_t) and time $t - k$ (P_{t-k}), with k representing any number of lags. These plots in Figure 7 and Figure 8, though similar in appearance, serve distinct purposes.

The first lag in the auto-correlation plot dictates the choice between an auto-regressive (AR) or moving average (MA) model [21]. Given a negative first lag

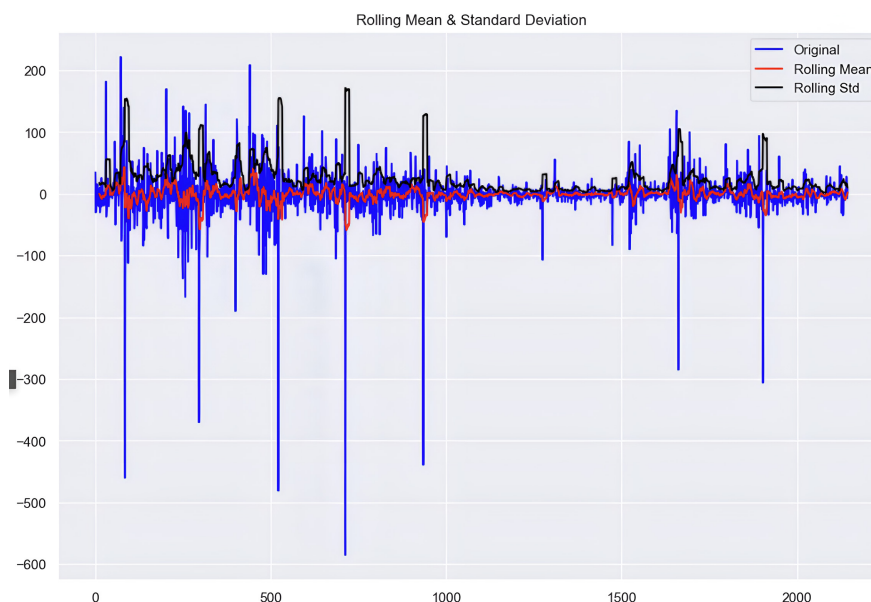


Figure 6. Dickey-Fuller test plot after differencing.

Table 7. Results of Dickey-Fuller test after differencing.

Test Statistic	-24.823159
p-value	0.000000
#Lags Used	3
Number of Observations Used	2140
Critical Value (1%)	-3.433409
Critical Value (5%)	-2.862892
Critical Value (10%)	-2.567489

and a subsequent positive second lag, we opt for the 1st lag as a moving average point. Conversely, the substantial drop-off at lag one in the partial auto-correlation plot prompts an AR factor of 1. Experimentation with lag selection is encouraged, with attention to the Akaike Information Criterion (AIC) to gauge model performance [22]. The ARIMA model, requiring input parameters of “p” for the AR term, “d” for differencing, and “q” for the MA term, emerges as a vital forecasting tool. Our data analysis identifies the optimal model order as (1,1,1). However, exploration of alternative parameter combinations is welcomed to minimize AIC values and enhance model accuracy [23]. The training process proves relatively swift, facilitating efficient model iteration and refinement.

Transitioning to model implementation and forecasting, we deploy the ARIMA model with a preferred order of (4,2,0). This step entails a notable computation time, necessitating caution. To streamline subsequent analyses, we retain model predictions through a magic method, obviating the need for repeated execution of time-consuming computations. The major parameters implemented are mentioned in **Table 8**.

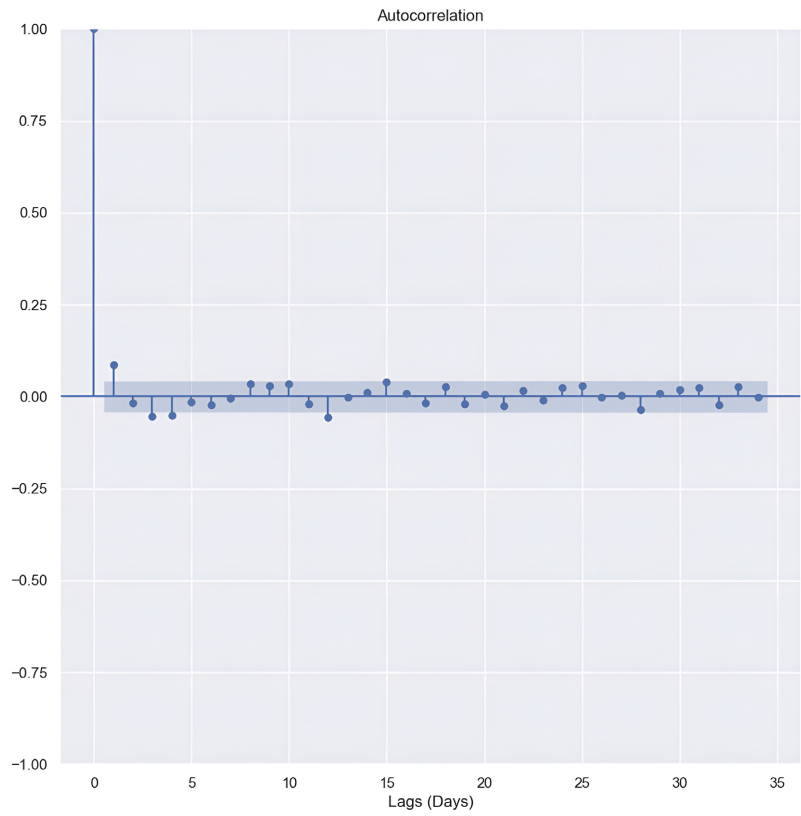


Figure 7. Autocorrelation.

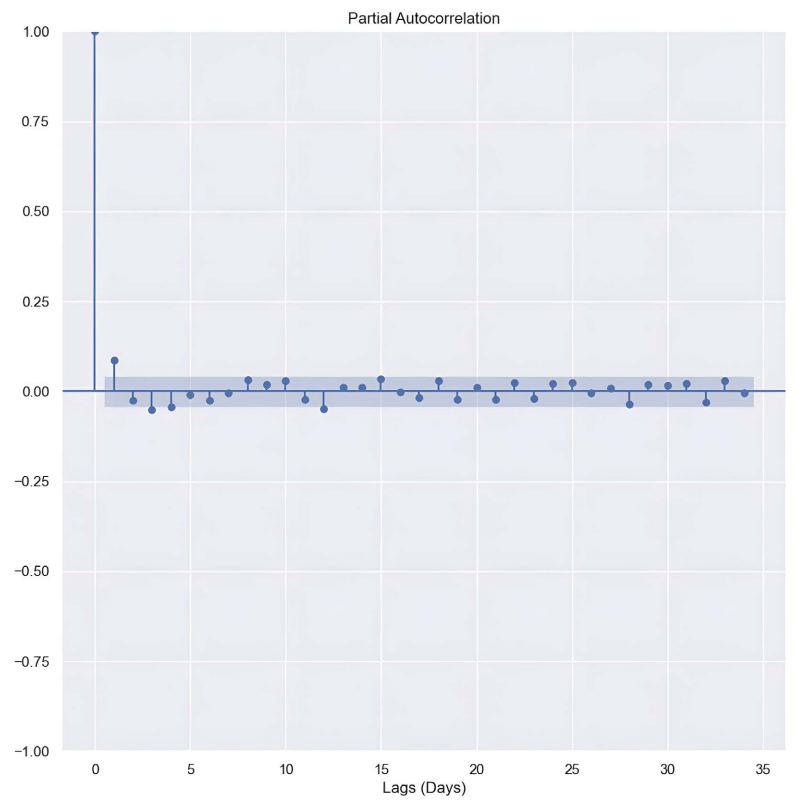


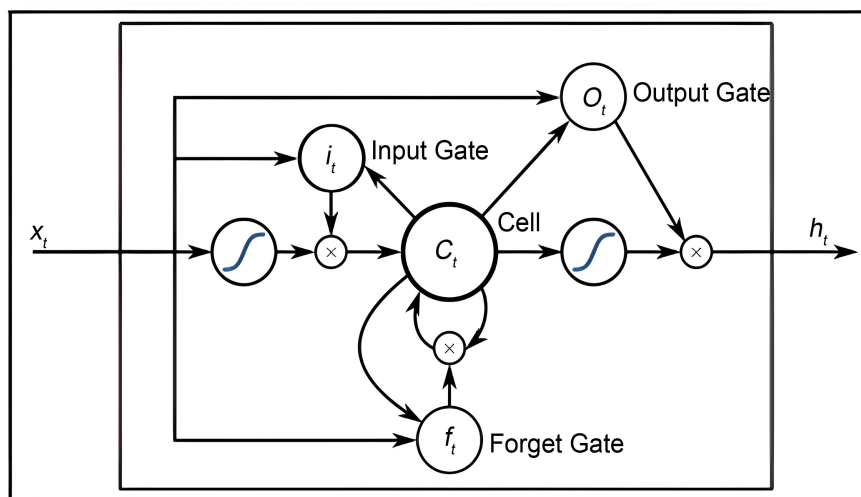
Figure 8. Partial autocorrelation.

Table 8. ARIMA model parameters.

Parameter	Value
Order of AR (p)	4
Order of Differencing (d)	2
Order of MA (q)	0
Input Size	1
Data Frequency	Daily

2.2.7. LSTM

In our pursuit of forecasting stock price movements, we turn to the powerful Long Short-Term Memory (LSTM) network. Renowned for its adeptness in handling sequential data and capturing long-term dependencies, LSTM emerges as a formidable choice. Its architecture, characterized by improved memory cells and intricate mechanisms such as forget gates, lends it a distinct advantage in stock price prediction tasks. Unlike traditional feedforward neural networks, LSTM's architecture as shown in **Figure 9** incorporates memory cells that allow it to retain information over extended periods. These cells are equipped with forget gates, which regulate the flow of information and play a crucial role in determining what information is relevant for prediction [25].

**Figure 9.** LSTM Architecture [24].

By selectively remembering past observations and disregarding noise, LSTM can extract meaningful patterns from time series data. Furthermore, LSTM's stateful nature enables it to maintain information across time steps, facilitating the capture of temporal dynamics and subtle nuances in the data [26]. This stateful behavior is particularly beneficial in financial forecasting, where historical context often plays a pivotal role in shaping future trends.

In our implementation, we employ a two-layer LSTM model, with each layer comprising 100 units. This architecture allows for the extraction of hierarchical

features from the input data, enabling the model to capture both short-term fluctuations and long-term trends. Additionally, we utilize Huber loss as the optimization criterion and the Nadam optimizer to train the model efficiently. To enhance training stability and prevent overfitting, we incorporate various training strategies, including learning rate scheduling, early stopping, and model checkpointing [27]. Learning rate schedule dynamically adjusts the learning rate during training, ensuring optimal convergence. Early stopping prevents the model from continuing to train once its performance on a validation set starts to degrade, while model checkpointing saves the best-performing model weights during training [28]. Through meticulous training and evaluation, our LSTM model endeavors to provide accurate and reliable forecasts of stock price movements. The major parameters implemented are mentioned in **Table 9**.

Table 9. Model parameters.

Parameter	Value
Window Size	30
Input Size	1
Number of LSTM Units per Layer	100
Number of Layers	2
Learning Rate	$1e^{-4}$
Maximum Epochs	500
Batch Size	1
Early Stopping Patience	50

2.3. Backend Development

In the backend development phase of the Predict It Django web application, the focus was on building the server-side components that handle data processing, storage, and communication with the front end as shown in **Figure 10**.

Django, a high-level Python web framework, was chosen for its robustness, scalability, and rapid development capabilities. The backend of the web application played a critical role in handling user requests, retrieving stock data, and serving the predictions generated by the machine learning model. SQLite3, a lightweight and serverless relational database management system, was utilized to manage the storage of historical stock price data and other relevant information required for the prediction process. SQLite3 was selected for its simplicity, portability, and suitability for smaller-scale applications like the one developed in this project.

The Yahoo Finance API is a valuable data source for web applications, allowing the backend to fetch real-time and historical stock price data for various companies. This data is essential for training machine learning models and providing accurate predictions to users. By integrating the API into the backend, the web application can access a wide range of financial data without extensive

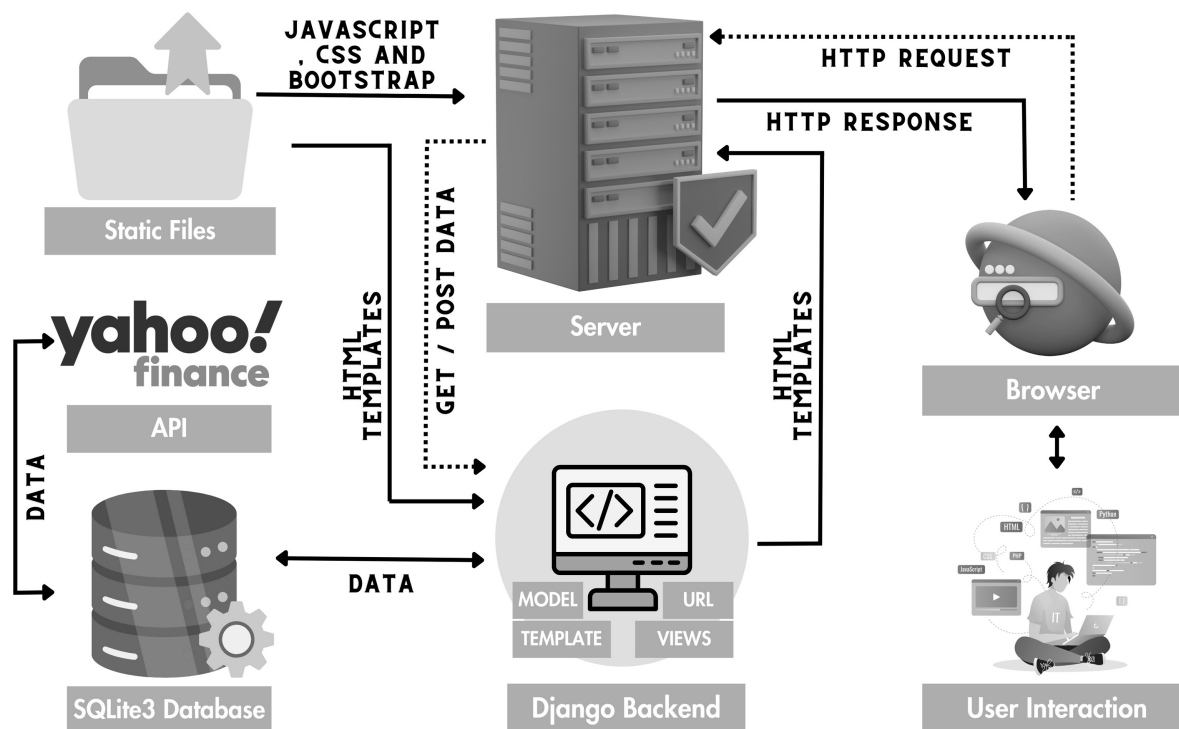


Figure 10. Django web application workflow.

data collection and storage. The backend handles user requests for stock price predictions and historical data visualization. When a user enters a specific stock symbol, the backend checks if the data is available in the local database (SQLite3) and if not, fetches the necessary historical data using the Yahoo Finance API. The model then generates a stock price prediction based on the provided data, which is then sent back to the front end for display. RESTful APIs were implemented to ensure smooth communication between the front-end and back-end. The backend was thoroughly tested to ensure functionality and reliability, with various test cases executed to validate its ability to handle user requests, fetch data from the Yahoo Finance API, and provide accurate predictions. By implementing an efficient backend in Django, the web application effectively integrates the machine learning model's predictions with real-time financial data, providing users with valuable insights and empowering them to make informed investment decisions.

2.4. Frontend Development

The frontend development phase of the Predict It Django web application aimed to create an intuitive and user-friendly interface for users to interact with the machine learning model's predictions and visualize historical stock price data. HTML provided the structural foundation, defining the layout and content of the web pages, while CSS enhanced the visual appeal and consistency, ensuring a visually pleasing design throughout the application. JavaScript brought interactivity and dynamic functionality, enabling features like form validation, re-

al-time updates, and asynchronous requests to the backend. The integration of Bootstrap ensured responsiveness, adapting the application's layout to different devices. Interactive charts generated using Matplotlib and Seaborn empowered users to explore historical stock performance and make informed decisions. The careful combination of these technologies resulted in a seamless user experience, allowing easy access to machine learning predictions and historical data, thus providing users with a powerful tool for financial analysis and decision-making.

3. Results and Discussion

3.1. Research Question One: Accuracy of Different Models

The purpose of this study was to evaluate and contrast the accuracy and efficiency of six different machine-learning algorithms for predicting stock prices. These algorithms were: Naive, Simple Moving Average (SMA 5), Simple Moving Average (SMA 20), Autoregressive Integrated Moving Average (ARIMA) model, linear regression, and Long Short-Term Memory (LSTM) model. Mean Absolute Error, or MAE is a reliable metric that was calculated to evaluate the predictive performance of each algorithm. The study chose Mean Absolute Error (MAE) as the evaluation metric for stock price prediction due to its straightforwardness, simplicity, and alignment with the objective. MAE quantifies the average prediction error without considering direction, making it suitable for stock price prediction tasks with extreme fluctuations and market dynamics. This allowed for an accurate comparison of each algorithm's predictive capabilities.

3.1.1. Naive Forecast

It was discovered that the MAE of the Naive algorithm, which was used as a fundamental reference model, was 13.17, which is a value that is on the lower end. **Figure 11** shows the full view of the entire training period. The Naive algorithm is a straightforward method for predicting future stock prices because it bases its projections solely on the most recent observation that is available. The naive model uses the price from the day before as its prediction of tomorrow's price. This is done without taking any historical patterns or trends into consideration in any way. Although the model's simplicity contributed to its low MAE, this same factor also hindered the model's ability to consider complex market dynamics.

3.1.2. SMA

Moving on to the Simple Moving Average (SMA) models as shown in **Figure 12**, which both had window sizes of 5 and 20 days, it was discovered that they had higher MAE values when compared to the Naive model. This was the case even though they both had the same number of days in the window. Simple moving averages take the average of the price over a certain period while exponential applies a weight factor to the average that decreases over time. SMA models function by first calculating an average of stock prices over a given period and then presenting this result as a more accurate depiction of price trends. This process

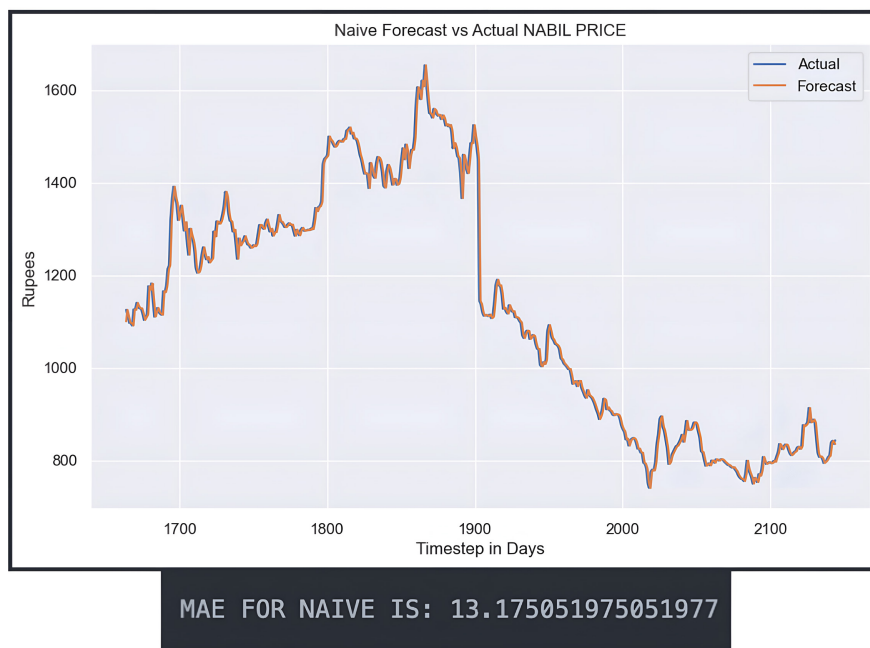
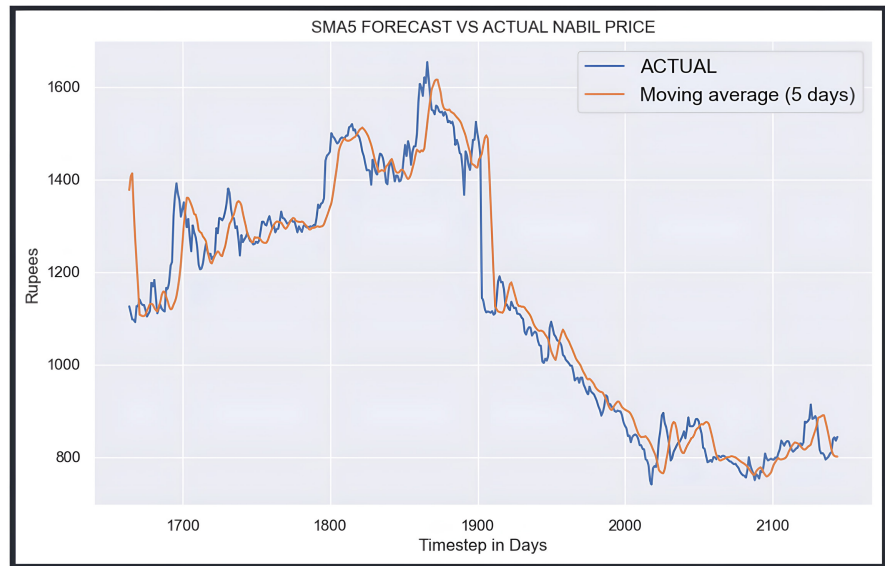


Figure 11. Naive forecast with MAE.

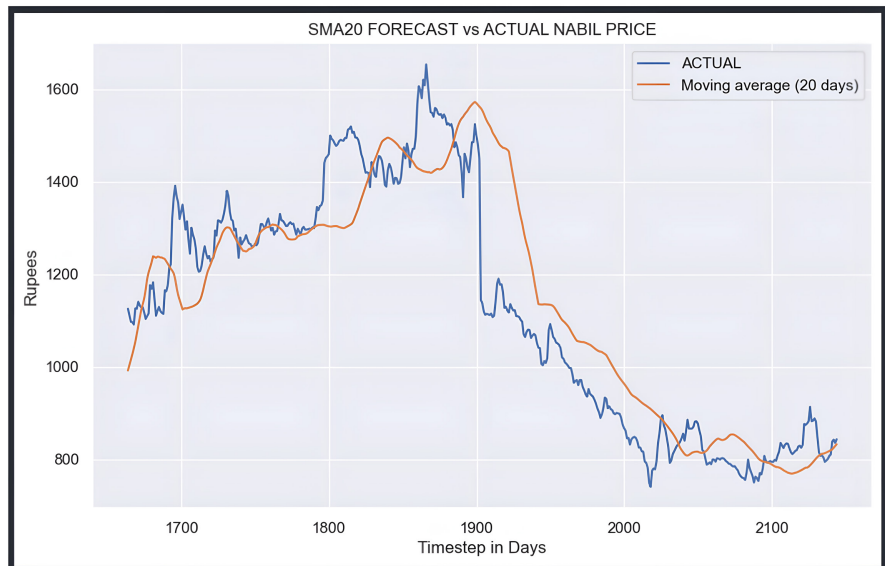
takes place over a predetermined time [29]. SMA models tend to lag current market conditions, which results in reduced prediction accuracy; however, this simplicity poses limitations when confronted with sudden market changes or volatile periods. This is because SMA models tend to be excessively simplistic. We can imply that SMA 20 is a better indication of a trend rather than a good predictor. It is also important to consider that there are only 5 trading days in a week so SMA 20 and SMA 5 are one trading month and one trading week respectively. The mean absolute error for SMA 5 was calculated to be 45.33, whereas the mean absolute error for SMA 20 was 88.50, demonstrating an even greater disparity between the two.

3.1.3. Linear Forecast

The linear regression model performed moderately well in predicting stock prices, with a mean absolute error (MAE) of 18.24 as shown in **Figure 13**. The goal of linear regression, a straightforward statistical technique, is to create a linear relationship between the input features and the target variable, in this case, the stock price. The linear model is implemented using a single dense layer in Keras for time series forecasting. A dense layer is just a regular layer of neurons in a neural network. Each neuron receives input from all the neurons in the previous layer, thus densely connected. The layer has a weight matrix W , a bias vector b , and the activations of the previous layer a [30]. Although the MAE value of 18.24 shows a respectable level of prediction accuracy, it is still within the average range of other evaluated algorithms. A straightforward yet understandable tool for prediction tasks, the linear regression model captures linear relationships between the chosen financial indicators and stock prices. Like any linear model, it might have trouble capturing intricate nonlinear dependencies



MAE FOR SMA5 IS: 45.33717255717256



MAE FOR SMA20 IS: 88.50874220374219

Figure 12. SMA forecast with MAE.

and patterns in the data. Despite its average performance, linear regression can be a helpful starting point model for predicting stock prices because it sheds light on the broad trend and direction of stock movements.

3.1.4. ARIMA Forecast

The Autoregressive Integrated Moving Average (ARIMA) model, created especially for time-series analysis, was able to predict stock prices with a moderate mean absolute error (MAE) value of 20.006 as shown in **Figure 14**. The acronym ARIMA, which stands for autoregression, differencing, and moving average, is a

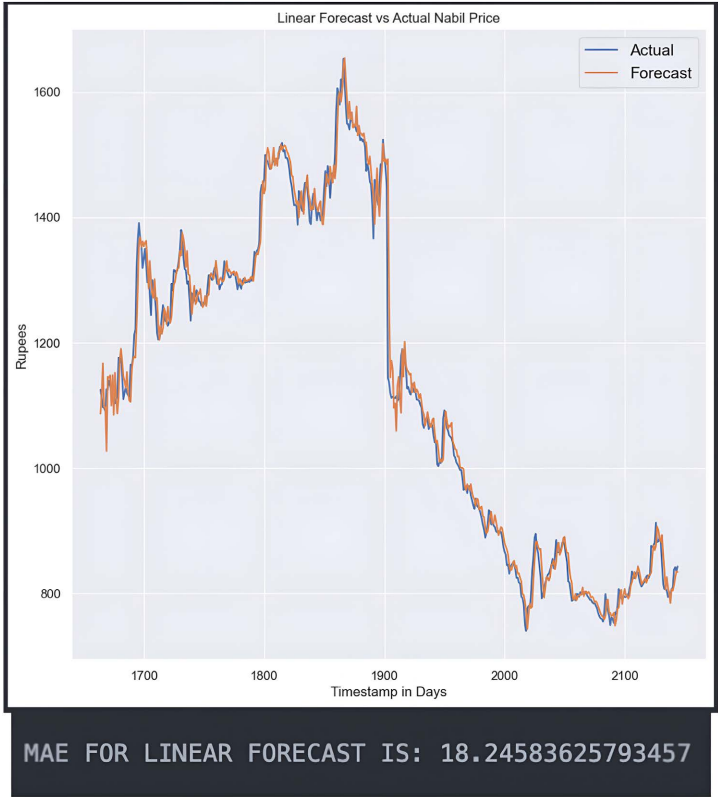


Figure 13. Linear forecast with MAE.

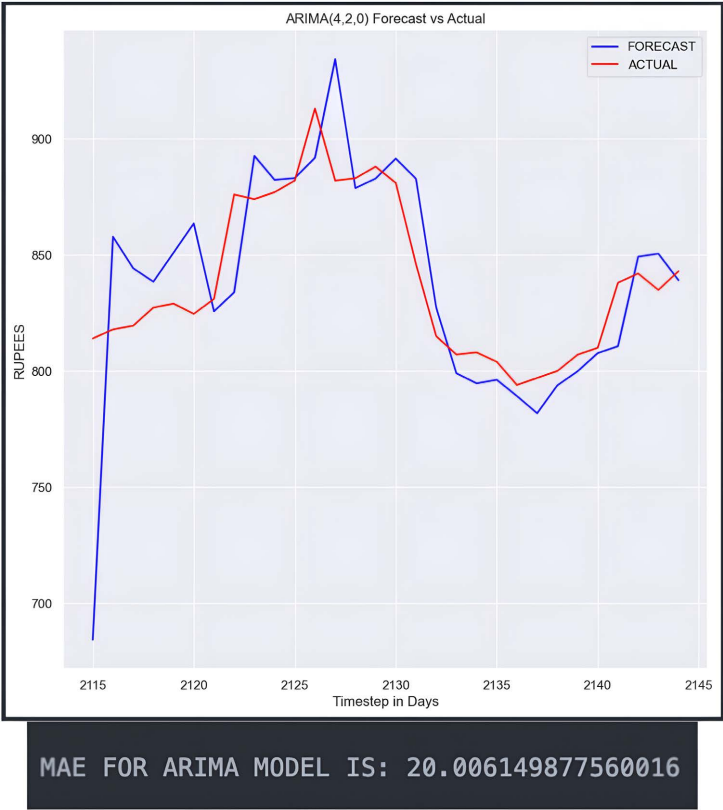


Figure 14. ARIMA forecast with MAE.

potent tool for time-series data analysis that considers both short- and long-term patterns. Its predictive abilities, however, may have been hampered by the complexity of financial market data and the existence of non-stationary trends, resulting in a marginally higher MAE than the Naive model. Despite this, ARIMA is still a useful method for modeling and comprehending time-series data because it provides information about the underlying trends and patterns that affect stock price changes. It is a good option for time-series prediction tasks because it can handle time-dependent relationships, but further optimization and parameter tuning may be necessary to increase its accuracy in capturing the nuances of financial markets [31].

3.1.5. LSTM Forecast

The long short-term memory (LSTM) network is a well-known deep learning model that has demonstrated its expertise in handling sequential data and capturing long-term dependencies. As a result, it is particularly well-suited for stock price prediction. The LSTM model has an edge over the other models as it has improved memory cells. These cells are linked to each gateway. It also contains Forget gates. The connection to the memory stick is controlled by the forget gates. They are also responsible for remembering the error as per requirement and scaling the feedback by each step. Due to its architecture, LSTM was able to quickly analyze historical price movements and identify intricate patterns, both of which were feats that other algorithms might not have been able to accomplish [32]. It was able to respond quickly to shifting market conditions and identify even minute shifts in the general direction of trends, both of which significantly aided its superior ability to predict the future.

The LSTM model with a 30-day window outperformed every other algorithm and had the lowest possible value among the models that were evaluated as shown in **Figure 15**. Its mean absolute error (MAE) was 11.55. Because of its adaptability and ability to recognize long-term relationships, the LSTM model was able to make more precise and nuanced predictions than any other algorithm in this study, making it the most promising algorithm for predicting stock prices. It can also store information of any size. The input to the LSTM model is provided in the form of vectors. This vector comprises elements such as the previous closing price, open price, high and low prices, and the trade volumes of the current session. The vector also contains the results of the news analysis. The output layer of the model gives the final predicted price over specific periods [33].

3.1.6. Comparative Analysis

The disparities in MAE values that exist between the various algorithms as shown in **Table 10** highlight how critical it is to select appropriate models based on the prediction tasks at hand. When it comes to capturing the complexities of financial market data, more complex and sophisticated models, such as ARIMA and LSTM, demonstrate superior performance. While simpler models, such as Naive and SMA, might be sufficient for basic forecasting, more complex and



Figure 15. LSTM forecast with MAE.

Table 10. Models and their MAE.

Model	MAE
LSTM	11.557
Naive	13.175
Linear Regression	18.245
ARIMA	20.006
SMA 5	45.33
SMA 20	88.5

sophisticated models, such as ARIMA and LSTM, do not. In conclusion, the findings of the research present convincing evidence for the utilization of advanced machine learning models, such as LSTM, for accurate stock price prediction. The resounding success of LSTM demonstrates its suitability for handling dynamic financial data, highlighting its potential to inform trading decisions and optimize investment strategies. The thorough analysis of several different algorithms provides useful insights into the field of stock market prediction, thereby paving the way for further advancements in predictive accuracy and techniques for data-driven decision-making.

3.2. Research Question Two: Factors Influencing Performance of Models

In the pursuit of understanding the key factors influencing the performance of machine learning (ML) algorithms in stock price prediction, the project delved into an in-depth analysis that provides compelling insights. The quality and

quantity of historical stock price data surfaced as a pivotal factor that profoundly impacted algorithm performance. With access to a comprehensive dataset from NpseAlpha for the NABIL stock, spanning 10 years, the algorithms were equipped to harness an extensive historical context. This abundance of data allowed the ML models to discern long-term trends, enabling more accurate predictions. Additionally, the dataset's reliability ensured that the algorithms were trained on robust information, minimizing the risk of erroneous outcomes. The amount of data used for training ML algorithms significantly impacts their ability to discern patterns and trends in stock price movements. Having a substantial amount of historical data spanning multiple years enables the models to capture long-term dependencies and fluctuations in the stock market. Regarding the features required in the dataset, several key attributes were vital for effective stock price prediction. These features included the date, open price, close price, high, low, and volume. The date served as the temporal identifier, allowing the algorithms to understand the sequential nature of stock price data. The open price, close price, and high, and low values provided valuable information about the stock's price movements and trends. The volume feature, representing the number of shares traded, indicates the level of market activity and liquidity, which could significantly influence stock prices. Feature engineering and data preprocessing emerged as influential components that bolstered algorithm performance. Furthermore, meticulous data preprocessing techniques, including normalization and scaling, mitigated the risk of skewed results. Feature selection, an integral part of the project's methodology, ensured that only the most informative features were employed for training.

Properly splitting the dataset into training and testing subsets is another crucial aspect of dataset management. In our project, we employed a commonly used practice of an 80 - 20 split, where 80 percent of the data was designated for training, and the remaining 20 percent was reserved for testing the model's performance. This division allows the model to learn from a substantial portion of historical data while preserving a separate segment for assessing its generalization capabilities. The test dataset acts as a proxy for unseen future data, and evaluating the model on this segment helps gauge its ability to make accurate predictions for real-world scenarios. Another distribution grid 60 percent for training and 40 percent for testing was not well suited as the MAE error was almost quadrupled as shown in **Figure 16**. It shows how crucial is the train test split while modeling data for stock price prediction.

The selection and configuration of ML algorithms played a critical role in determining predictive accuracy. By meticulously choosing six distinct algorithms, including Naive, SMA 5, SMA 20, ARIMA, linear regression, and LSTM, our project aimed to uncover each model's strengths and weaknesses. Careful consideration of these algorithms' suitability for time-series prediction tasks ensured that the project's evaluation was comprehensive and relevant to the stock market domain. The project's focus on diverse algorithms demonstrated a thorough exploration of various methodologies, enhancing the credibility of our findings.

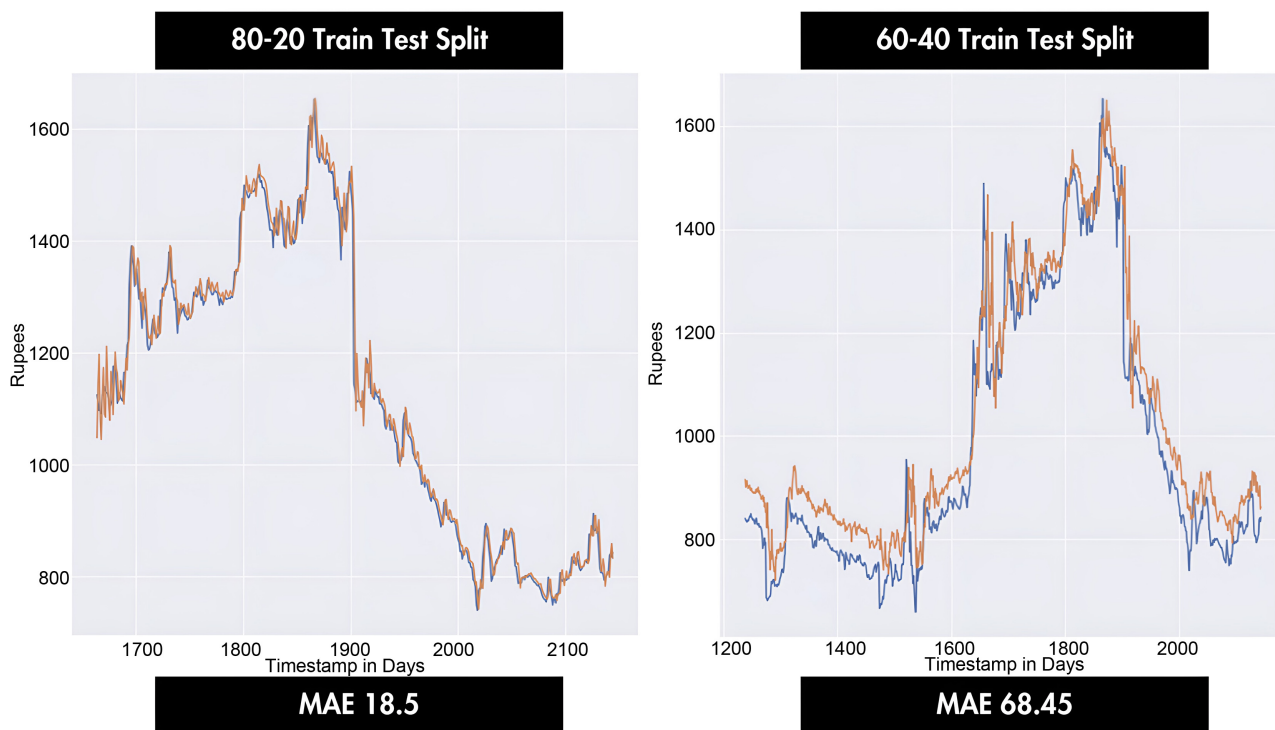


Figure 16. Train test distribution (80 - 20 vs 60 - 40).

3.3. Research Question Three: Ethical Considerations

Machine learning algorithms play a significant role in predicting stock prices, but they also raise ethical concerns. One of the primary concerns is the lack of transparency and interpretability of complex models, such as LSTM, which function as “black boxes.” This lack of transparency raises questions about the logic that underpins their predictions, which can cause mistrust and uncertainty in the financial market. It is essential to foster trust among investors and stakeholders to ensure transparency and accountability in the decision-making process of these algorithms. Another significant ethical issue is the degree to which machine learning algorithms are biased or fair. These algorithms are trained on historical data, which can perpetuate biases that can result in biased predictions and decision-making [34]. Data privacy and security are also important considerations, as machine learning algorithms may contain private information about companies, investors, and trading strategies. It is necessary to protect individuals’ rights to privacy and maintain faith in the financial system by preventing unauthorized access to this data and potential breaches. Market manipulation and high-frequency trading are also significant ethical concerns, as the speed with which trades can be executed using algorithmic trading can affect both stock prices and market stability. To stop market manipulation and protect the integrity of the market, regulators need to closely monitor practices like herding behavior, which can be influenced by the widespread use of machine learning algorithms. Another important ethical consideration is not to rely too heavily on computer programs, as placing too much weight on their forecasts without con-

sidering broader market trends and qualitative factors can lead to less optimal investment decisions. Effective investment strategies must strike a balance between the insights provided by algorithms and human judgment. The application of machine learning algorithms in the financial sector brings with it a host of regulatory and legal challenges. It is crucial to ensure compliance with current financial regulations and protect against potential violations. Additionally, the rapid pace of technological advancement may force regulatory measures to be flexible and adaptable. To make effective use of machine learning algorithms, it is essential to encourage well-informed decision-making and financial literacy. Providing investors with educational opportunities can help them navigate the complexities of algorithmically driven financial markets and make informed decisions based on accurate information. In conclusion, machine learning algorithms hold significant promise for predicting stock prices, but their application must be guided by ethical principles. Factors to consider include openness and fairness, protection of personal information, accountability, and responsible decision-making.

4. Limitations and Future Work

4.1. Limitations

The project has several limitations, including reliance on a specific dataset from NepsAlpha, which focuses solely on the NABIL stock, and the data preprocessing phase. The narrow scope of the dataset may restrict generalizability to other stocks or the broader market. Additionally, the data can be complex and noisy, requiring careful cleaning and feature engineering. Market volatility and unforeseen events can introduce uncertainties into stock price prediction models, which can challenge the algorithms' ability to accurately forecast stock prices. Overfitting is an inherent concern when dealing with machine learning models. The absence of external factors in the dataset, such as economic indicators, company-specific fundamentals, or geopolitical events, may limit the model's ability to capture the full complexity of stock price movements. The deployment of ML models in real-world financial applications introduces practical challenges such as data updates, model retraining, and real-time prediction. Addressing these deployment challenges is essential for ensuring the continued effectiveness of the models in a live trading environment. The potential for unintended consequences or market manipulation demands a responsible and transparent approach to the deployment and use of predictive models. Interpreting the results of complex models like LSTM can be challenging, as their internal workings may be less transparent than traditional algorithms.

4.2. Future Work

Future research should explore the integration of sentiment analysis of financial news and market sentiments into predictive models, gaining valuable insights into investor sentiment and its impact on stock prices. Expanding beyond tradi-

tional stock market indicators, such as economic indicators, company-specific fundamentals, and geopolitical events, can provide a more holistic understanding of stock price movements. Advanced deep learning models like GRU or Transformer-based models like BERT could improve predictive accuracy [35]. Ensemble methods, like stacking or boosting, can combine predictions of multiple machine learning models, potentially reducing model bias and variance. Integrating predictive models developed in this project into algorithmic trading systems can harness their predictive prowess to make real-time trading decisions. The ability to process vast amounts of data quickly and accurately can be leveraged to identify profitable trading opportunities and execute trades with precision and efficiency. This pursuit of future endeavors will lead to significant progress in stock price prediction, providing investors and traders with more accurate and reliable tools to navigate the complexities of the financial market.

5. Conclusion

In conclusion, the research embarked on an appropriate analysis of machine learning algorithms for stock price prediction, which resulted in a decent contribution to the field of financial analytics. The utilization of a combination of the OSEMN methodology and the Agile Scrum methodology made it possible to evaluate, select, and implement the best-performing LSTM algorithm into the Predict It Django web application. This included everything from the collection of data to the development of web applications. The findings of the research shed light on the accuracy and efficiency of various machine learning algorithms, with the LSTM model emerging as the best performer with MAE of 11.55, thereby showcasing its potential for real-world financial applications. Ethical considerations were thoughtfully addressed, and the importance of the responsible and open application of predictive modeling in financial decision-making was emphasized throughout the process. When we consider the future, the recommendations made by the project clear the way for further developments in the industry. Some examples of these developments include incorporating sentiment analysis, exploring additional features, and implementing predictive models in algorithmic trading. This project, despite its limitations, serves as a platform for further research, providing tools that enable investors, traders, and financial practitioners to make well-informed decisions in dynamic and ever-evolving markets.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

- [1] Vijh, M., Chandola, D., Tikkiwal, V.A. and Kumar, A. (2020) Stock Closing Price Prediction Using Machine Learning Techniques. *Procedia Computer Science*, **167**, 599-606. <https://doi.org/10.1016/j.procs.2020.03.326>

- [2] Idrees, S.M., Alam, M.A. and Agarwal, P. (2019) A Prediction Approach for Stock Market Volatility Based on Time Series Data. *IEEE Access*, **7**, 17287-17298. <https://doi.org/10.1109/access.2019.2895252>
- [3] Parray, I.R., Khurana, S.S., Kumar, M. and Altalbe, A.A. (2020) Time Series Data Analysis of Stock Price Movement Using Machine Learning Techniques. *Soft Computing*, **24**, 16509-16517. <https://doi.org/10.1007/s00500-020-04957-x>
- [4] Raubitzek, S. and Neubauer, T. (2022) An Exploratory Study on the Complexity and Machine Learning Predictability of Stock Market Data. *Entropy*, **24**, Article 332. <https://doi.org/10.3390/e24030332>
- [5] Padakandla, S. (2021) A Survey of Reinforcement Learning Algorithms for Dynamically Varying Environments. *ACM Computing Surveys*, **54**, 1-25. <https://doi.org/10.1145/3459991>
- [6] Bouteska, A. and Regaieg, B. (2020) Loss Aversion, Overconfidence of Investors and Their Impact on Market Performance Evidence from the US Stock Markets. *Journal of Economics, Finance and Administrative Science*, **25**, 451-478. <https://doi.org/10.1108/jefas-07-2017-0081>
- [7] Ren, R., Wu, D.D. and Liu, T. (2019) Forecasting Stock Market Movement Direction Using Sentiment Analysis and Support Vector Machine. *IEEE Systems Journal*, **13**, 760-770. <https://doi.org/10.1109/jsyst.2018.2794462>
- [8] Patel, J., Shah, S., Thakkar, P. and Kotecha, K. (2015) Predicting Stock and Stock Price Index Movement Using Trend Deterministic Data Preparation and Machine Learning Techniques. *Expert Systems with Applications*, **42**, 259-268. <https://doi.org/10.1016/j.eswa.2014.07.040>
- [9] Kandel, L.R. (2018) Risk and Return Analysis of Commercial Banks of Nepal (with Reference to NABIL and NIBL). *Pravaha*, **24**, 109-119. <https://doi.org/10.3126/pravaha.v24i1.20231>
- [10] Xu, S.Y. (2014) Stock Price Forecasting Using Information from Yahoo Finance and Google Trend. Master's or doctoral thesis, UC Berkely.
- [11] Nayak, S.C., Misra, B.B. and Behera, H.S. (2014) Impact of Data Normalization on Stock Index Forecasting. *International Journal of Computer Information Systems and Industrial Management Applications*, **6**, 257-269.
- [12] Rouf, N., Malik, M.B. and Arif, T. (2021) Predicting the Stock Market Trend: An Ensemble Approach Using Impactful Exploratory Data Analysis. In: Bhattacharya, M., Kharb, L. and Chahal, D., Eds., *Communications in Computer and Information Science*, Springer, 223-234. https://doi.org/10.1007/978-3-030-88378-2_18
- [13] Shen, S., Jiang, H. and Zhang, T. (2012) Stock Market Forecasting Using Machine Learning Algorithms. Department of Electrical Engineering, Stanford University.
- [14] Pal, A. and Prakash, P.K.S. (2017) Practical Time Series Analysis: Master Time Series Data Processing, Visualization, and Modeling Using Python. Packt Publishing Ltd.
- [15] Kumar, D., Sarangi, P.K. and Verma, R. (2022) A Systematic Review of Stock Market Prediction Using Machine Learning and Statistical Techniques. *Materials Today: Proceedings*, **49**, 3187-3191. <https://doi.org/10.1016/j.matpr.2020.11.399>
- [16] Mahajan, S.D., Deshmukh, K.V., Thite, P.R., Samel, B.Y. and Chate, P.J. (2016) Stock Market Prediction and Analysis Using Naïve Bayes. *International Journal on Recent and Innovation Trends in Computing and Communication*, **4**, 121-124.
- [17] Chatzis, S.P., Siakoulis, V., Petropoulos, A., Stavroulakis, E. and Vlachogiannakis, N. (2018) Forecasting Stock Market Crisis Events Using Deep and Statistical Ma-

- chine Learning Techniques. *Expert Systems with Applications*, **112**, 353-371. <https://doi.org/10.1016/j.eswa.2018.06.032>
- [18] Gururaj, V., Shriya, V.R. and Ashwini, K. (2019) Stock Market Prediction using Linear Regression and Support Vector Machines. *International Journal of Applied Engineering Research*, **14**, 1931-1934.
- [19] Ristanoski, G., Liu, W. and Bailey, J. (2013) Time Series Forecasting Using Distribution Enhanced Linear Regression. *Lecture Notes in Computer Science*, **7818**, 484-495. https://doi.org/10.1007/978-3-642-37453-1_40
- [20] Ariyo, A.A., Adewumi, A.O. and Ayo, C.K. (2014) Stock Price Prediction Using the ARIMA Model. 2014 *UKSim-AMSS 16th International Conference on Computer Modelling and Simulation*, Cambridge, 26-28 March 2014, 106-112. <https://doi.org/10.1109/uksim.2014.67>
- [21] Zhu, Z. and He, K. (2022) Prediction of Amazon's Stock Price Based on ARIMA, Xgboost, and LSTM Models. *Proceedings of Business and Economic Studies*, **5**, 127-136. <https://doi.org/10.26689/pbes.v5i5.4432>
- [22] Khan, S. and Alghulaiakh, H. (2020) ARIMA Model for Accurate Time Series Stocks Forecasting. *International Journal of Advanced Computer Science and Applications*, **11**, 524-528. <https://doi.org/10.14569/ijacsa.2020.0110765>
- [23] Mondal, P., Shit, L. and Goswami, S. (2014) Study of Effectiveness of Time Series Modeling (ARIMA) in Forecasting Stock Prices. *International Journal of Computer Science, Engineering and Applications*, **4**, 13-29. <https://doi.org/10.5121/ijcsea.2014.4202>
- [24] Trafalis, T.B. and Ince, H. (2000) Support Vector Machine for Regression and Applications to Financial Forecasting. *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium*, Como, 27 July 2000, 348-353. <https://doi.org/10.1109/ijcnn.2000.859420>
- [25] Rather, A.M. (2021) LSTM-Based Deep Learning Model for Stock Prediction and Predictive Optimization Model. *EURO Journal on Decision Processes*, **9**, Article ID: 100001. <https://doi.org/10.1016/j.ejdp.2021.100001>
- [26] Moghar, A. and Hamiche, M. (2020) Stock Market Prediction Using LSTM Recurrent Neural Network. *Procedia Computer Science*, **170**, 1168-1173. <https://doi.org/10.1016/j.procs.2020.03.049>
- [27] Pramod, B.S. and Shastry, M.P.M. (2021) Stock Price Prediction Using LSTM. *Test Engineering and Management*, **83**, 5246-5251.
- [28] Gao, Y., Wang, R. and Zhou, E. (2021) Stock Prediction Based on Optimized LSTM and GRU Models. *Scientific Programming*, **2021**, Article ID: 4055281. <https://doi.org/10.1155/2021/4055281>
- [29] Tsai, C.F. and Wang, S.P. (2009) Stock Price Forecasting by Hybrid Machine Learning Techniques. *Proceedings of the International Multi Conference of Engineers and Computer Scientists*, **1**, 60.
- [30] Wen, M., Li, P., Zhang, L. and Chen, Y. (2019) Stock Market Trend Prediction Using High-Order Information of Time Series. *IEEE Access*, **7**, 28299-28308. <https://doi.org/10.1109/access.2019.2901842>
- [31] Zeng, Z. and Khushi, M. (2020) Wavelet Denoising and Attention-Based RNN-ARIMA Model to Predict Forex Price. 2020 *International Joint Conference on Neural Networks (IJCNN)*, Glasgow, 19-24 July 2020, 1-7. <https://doi.org/10.1109/ijcnn48605.2020.9206832>

- [32] Skehin, T., Crane, M. and Bezbradica, M. (2018) Day Ahead Forecasting of FAANG Stocks Using ARIMA, LSTM Networks and Wavelets. *Proceedings of the 26th AIAI Irish Conference on Artificial Intelligence and Cognitive Science (AICS 2018)*, Dublin, 6-7 December 2018, 334-340.
- [33] Mahadik, A., Vaghela, D. and Mhaisgawali, A. (2021) Stock Price Prediction Using LSTM and ARIMA. 2021 *Second International Conference on Electronics and Sustainable Communication Systems (ICESC)*, Coimbatore, 4-6 August 2021, 1594-1601. <https://doi.org/10.1109/icesc51422.2021.9532655>
- [34] Turner Lee, N. (2019) Algorithmic Bias Detection and Mitigation: Best Practices and Policies to Reduce Consumer Harms. Brookings.
- [35] Nikou, M., Mansourfar, G. and Bagherzadeh, J. (2019) Stock Price Prediction Using DEEP Learning Algorithm and Its Comparison with Machine Learning Algorithms. *Intelligent Systems in Accounting, Finance and Management*, **26**, 164-174. <https://doi.org/10.1002/isaf.1459>