Comparative Analysis of Machine Learning Models for S&P 500 Prediction

MSc Data Analytics

Soumiya Kanwar
Student ID: 21218323

School of Computing
National College of Ireland

Supervisor: Abid Yaqoob

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: ……………SOUMIYA KANWAR…………………………………………………………

Student ID: …………… X21218323…………………………………………………………….………

Programme: ……… MSc Data Analytics ………… **Year:** …2023-24……

Module: ……………Research Project…………………………………………………………….………

Supervisor: ……… ABID YAQOOB…………………………….…………………………………….

Submission Due Date: ………………14 Dec 2023……………………………………………………….………

Project Title: Comparative Analysis of Machine Learning Models for S&P 500 Prediction…………………………….……………………………………………

Word Count: ……………7667……………… **Page Count**………………21…………………….

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: …Soumiya Kanwar………………………………………………………………………

Date: ……14 Dec 2023…………………………………………………………………………

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Comparative Analysis of Machine Learning Models for S&P 500 Prediction

Soumiya Kanwar

X21218323

Abstract

Three different machine learning algorithms namely Random Forest Classification, LSTM network, and Logistics regression are used to predict price changes of S&P 500 stocks. The study employs two distinct timelines for analysis: one covering from year 2000 till now and involving different markets situations, the other concentrating just in recession periods of 2007-2009.

The study starts by collecting past S&P 500 stock price and the necessary characteristics to produce an inclusive dataset. Subsequently, Random Forest Classification, LSTM, and Logistic Regression are used as predictive models, where each one possesses its distinct advantages.

The three models are subjected to comparisons by evaluating how they perform on the task of forecasting stock prices in these two periods. The aim of this study is to discover what works and does not work in each trading algorithm during both ordinary fluctuations in the market and abnormal recession period.

This project reveals a lot about machine learning's use in financial prediction, which is of paramount importance to both investors, analysts, and researchers. The aim is to enlarge the applicability of predictive modelling regarding stock price prediction for the case study S&P 500 index through assessment of the models' accuracy, robustness, and adaptability during both steady and turbulent conditions.

1 Introduction

Investors, analysts, and researchers have always focused on the stock market, an ever-changing complex system. Forecasting stock prices precisely is still an overwhelming task, and in recent years, using machine learning algorithms seems to be a promising approach towards understanding financial market mysteries *(Kamal, n.d.) (Fu et al., 2021) (Martínez-Sánchez et al., 2023)*. The performance of the predictive models in complex market scenario helps the investment managers to take smart decisions and make their portfolio robust.

This study is motivated by the need to manoeuvre through the murky waters of financial markets, especially as it relates to the S&P 500 Index. A comparison of the efficiency of three different machine learning algorithms namely random forest classification and LSTM network as well as logistic regression for predicting change in prices of S&P 500 stocks. These algorithms are selected based on their distinct advantages to arrive at an overall assessment of their performance.

This research has its temporal focus on two important periods. This includes data ranging from the year 2000 up to the current day and gives an idea about the algorithm's efficiency in different market conditions. The second zooms on the recessionary period of 2007-2009 to critically examine how the models fared during turbulent economic times. This is what makes this dual-temporal approach unique; allowing the study to capture those characteristics that form part of predictive modelling during typical periods as well as in times of crisis.

Research question of this study: How do using deep learning and machine learning models affect how accurately and flexibly we can predict S&P 500 stock prices in different market situations, including normal difficulties and the recession from 2007 to 2009? Objective is to seeks to evaluate the predictive ability, consistency, and suitability of three specific machine learning models including Random Forest classification, LSTM networks, and Logistic regression in forecasting stock prices for the S&P 500. The evaluation covers a variety of markets situations, including routine movements as well as a recession period of 2007–2009. The aim is to assess a specificity of these algorithms in common fluctuations on markets and a robustness and flexibility, under extreme depressions like a recession. Furthermore, it will be compared to determine the strong/weak sides of every model that should be useful for investors/analysts/researchers.

The report delves into the topic of the thesis with an in-depth review and analysis of existing academic research, studies, and relevant literature in the related work section. A research methodology section follows, focusing on the datasets with Exploratory Data Analysis (EDA) carried out on both datasets to provide a better understanding of the data. The Design Specification section explains the architecture and flow of the models while the implementation involves providing different values to different models after data preprocessing. Finally, the evaluation compares all the resources obtained from the models. This comprehensive approach ensures that the report covers all aspects of the thesis, making it an effective and persuasive piece of work.

2 Related Work

2.1 Time Series Analysis and Modelling of Financial Data

In *(Mateusz and Ślepaczuk. 2020)*, algorithmic investment strategies are explored using classical methods and LSTM recurrent neural networks on two decades of S&P 500 data. It highlights the effectiveness of diverse strategy signals, surpassing the Buy & Hold benchmark. Concerns emerge about LSTM's robustness compared to classical algorithms, prompting exploration into optimizing signal combinations and investment techniques.

(González-Rivera and Arroyo, 2012) delves into the application of machine learning algorithms in predicting stock prices, emphasizing the need for robust models. This study contributes valuable insights into the challenges and potential advancements in the domain of predictive modelling for financial markets.

Shifting focus to *(Fu et al., 2021)*, the study investigates firm-specific investor sentiment and stock price crash risk in Chinese firms from 2005 to 2016. Introducing a sentiment index, it uncovers a significant positive correlation with crash risk, particularly in companies with lower liquidity. These findings provide valuable insights for market participants and regulators, revealing the nuanced impact of investor sentiment on crash risk.

2.2 Symbolic Data Analysis and Time Series Forecasting

(Papaioannou et al., 2017) introduces a trend-following trading strategy that outperforms the traditional "Buy and Hold" benchmark, emphasizing the importance of heuristic asset selection and proposing a data-driven approach using the Internet of Things. Similarly**,** *(Lohrmann and Luukka, 2019)* challenges binary classification norms by classifying S&P 500 returns into four classes using a random forest classifier, highlighting superior accuracy, and encouraging a more nuanced trading strategy. *(Vogl, 2024)* delves into chaos measures as predictors in a dynamic factor model, employing a deep learning neural network and suggesting further research avenues. Simultaneously**,** *(Parnes, 2020)* explores economic anomalies using LOESS regression, highlighting cyclicality patterns, and recommending future applications in other indexes.

2.3 Behavioural Finance and Investor Sentiment

(Papaioannou et al., 2017) introduces a trend-following trading strategy for the S&P 500 index using convolution computations, highlighting directional predictability, and suggesting heuristic asset selection. *(Lohrmann and Luukka, 2019)* focuses on classifying S&P 500 open-to-close returns, emphasizing the importance of momentum changes and challenging binary classification approaches**.** *(Vogl, 2024)* explores time-variation of chaos measures in S&P 500 returns, offering a dynamical system-based perspective and highlighting the potential of chaos measures as predictors. *(Parnes, 2020)* investigates economic anomalies in the S&P 500 index using LOESS regression, identifying cyclicality patterns, and recommending future anomaly detection studies. In contrast, *(Jadhav et al., 2021)* introduces a GAN-based approach to predict stock prices, combining sentiment analysis and LSTM for forecasting, with a focus on enhancing efficiency. *(Kroencke, 2022)* employs an event study approach to analyse stock prices and dividends during recessions, emphasizing the role of changes in the price of risk**.** *(P H and Rishad, 2020)* examines investor sentiment in the Indian stock market, revealing its impact on excess volatility and challenging the efficient market hypothesis. Lastly, *(Parveen et al., 2020)* explores cognitive biases in the Pakistan Stock Exchange, highlighting the influence of representative heuristic and overconfidence on investor decisions.

The collective merit lies in the variety of methodologies employed, including convolution computations, chaos measures, GANs, LOESS regression, and event studies. Each paper provides unique perspectives on forecasting, anomaly detection, sentiment analysis, and the impact of cognitive biases. Merits include empirical evidence, algorithmic advancements, and novel insights into market behaviour. Limitations across studies include the need for further testing, potential biases in certain methodologies, and challenges to existing financial theories. This compilation enriches the literature by combining traditional and advanced methodologies, offering a holistic view of financial markets, and encouraging future research in these directions.

2.3 Machine Learning and Deep Learning Models for Stock Price

(Xiao and Su, 2022) Exploring the intricate connections between machine learning techniques and financial forecasting, a notable research paper entitled "Predicting Stock Prices Over Time Using Advanced Modeling and Historical Patterns" stands out for its examination of applying ARIMA and LSTM algorithms to anticipate stock market fluctuations. Covering the decade from 2010 through 2019 on the New York Stock Exchange, the investigation unfolds with meaningful insights. The ARIMA and LSTM approaches highlight their abilities to manage both linear and nonlinear challenges in predicting changes to stock values over sequential time periods, skillfully analyzing individual numeric tendencies. Metrics like mean squared error, mean absolute error, and root mean squared error evaluate performance, revealing a balanced interplay achieving highly accurate results for projections made by both statistical techniques regarding stock indexes. Deep learning algorithms show promise, with LSTM performance exceeding traditional ARIMA methods. A hybrid ARIMA-LSTM model emerges as a strong candidate, demonstrating ability to forecast correlation coefficients useful for portfolio selection. The study recognizes the need to evaluate the model using earlier market data and urges researchers to address challenges. It invites those involved in China's stock market to consider these findings. While this combined approach shows predictive skill, the research humbly notes limitations and calls for further refinement by future work to better anticipate stock price movements.

3 Research Methodology

The dataset is divided into two distinct timelines: the first covering the time between 2000 and the current times, and the second covering the recessionary period (the years between 2007 and 2009). A unique feature of this case is that it incorporates a dual-temporary lens into the discussion on how stocks behave in varying markets.

Firstly, we collect a dataset with important financial information such as „Close, "„Volume, "Open. "„High ", ‶ and Low" from Yahoo Finance towards S & P 500 index. Aim is made possible by removing dividends and stock split columns to ease further analyses of the data set. The research moves to Exploratory Data Analysis (EDA) according to specific timelines. At the recessing time, the dataset should be narrow and descriptive statistics together (Table 1) with visualization will shed an overview of the situation.

	Open	High	Low	Close	Volume
count	756	756	756	756	7.56E+02
mean	1214.998584	1225.432288	1203.39	1214.750793	4.61E+09
std	253.106839	251.246626	254.737	252.949807	1.58E+09
min	679.280029	695.27002	666.79	676.530029	1.22E+09
25%	953.327515	979.697479	943.438	954.45752	3.39E+09
50%	1288.234985	1300.164978	1274.86	1288.664978	4.38E+09
75%	1443.935028	1450.224945	1433.21	1444.347504	5.63E+09
max	1564.97998	1576.089966	1555.46	1565.150024	1.15E+10

Table 1. descriptive statistics (2007-2009)

In Figure 1 the graph shows that the stock price has been going down since 2007, but it has started to go back up in recent years. The price is now lower than it was in 2007, but it is higher than it was in 2009.

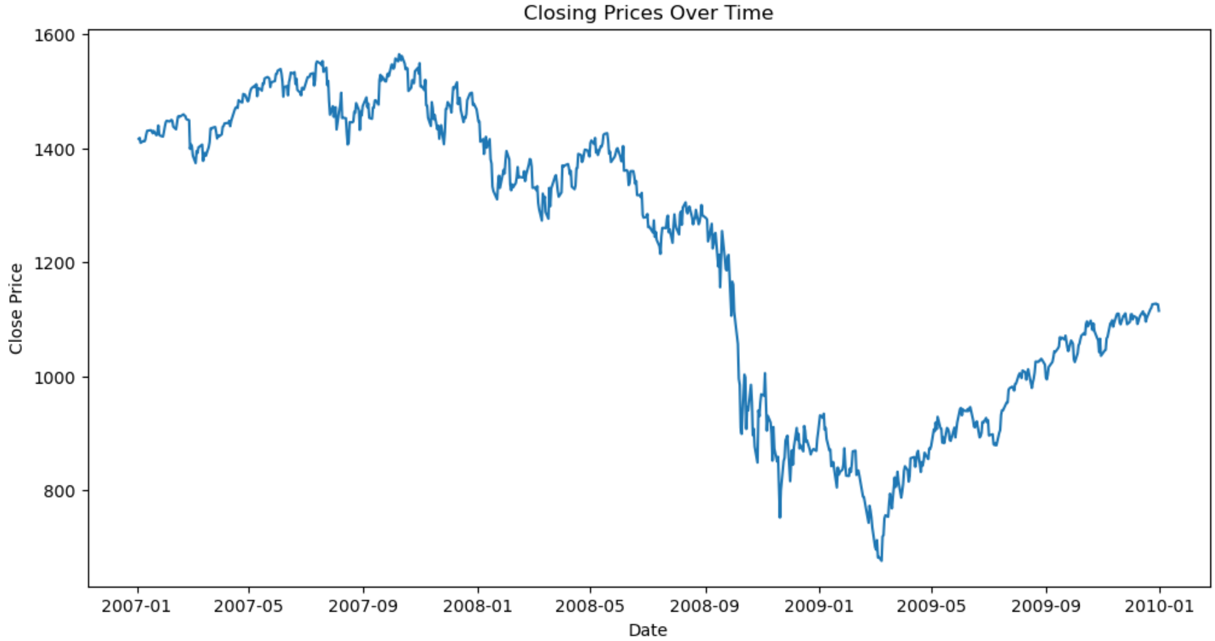

Figure 1. Closing Price Over time (2007-2009)

For the 2000 to now timeline, filtering of data is done, and relevant statistical measures (Table 2) and visualizations are used to describe important findings. Figure 2 shows that the price of cleaning supplies has increased steadily over the past 20 years.

	Open	High	Low	Close	Volume
count	6025	6025	6025	6025	6.03E+03
mean	1967.78925	1979.42306	1955.2864	1968.06145	3.33E+09
std	1054.89526	1060.22574	1049.33285	1055.13805	1.51E+09
min	679.280029	695.27002	666.789978	676.530029	3.56E+08
25%	1191.17004	1198.47998	1184.16003	1191.32996	2.08E+09
50%	1456.63001	1464.93994	1446.06006	1456.63001	3.44E+09
75%	2577.75	2586.5	2565.93994	2579.8501	4.16E+09
max	4804.50977	4818.62012	4780.04004	4796.56006	1.15E+10

Table 2. descriptive statistics (2000-present)

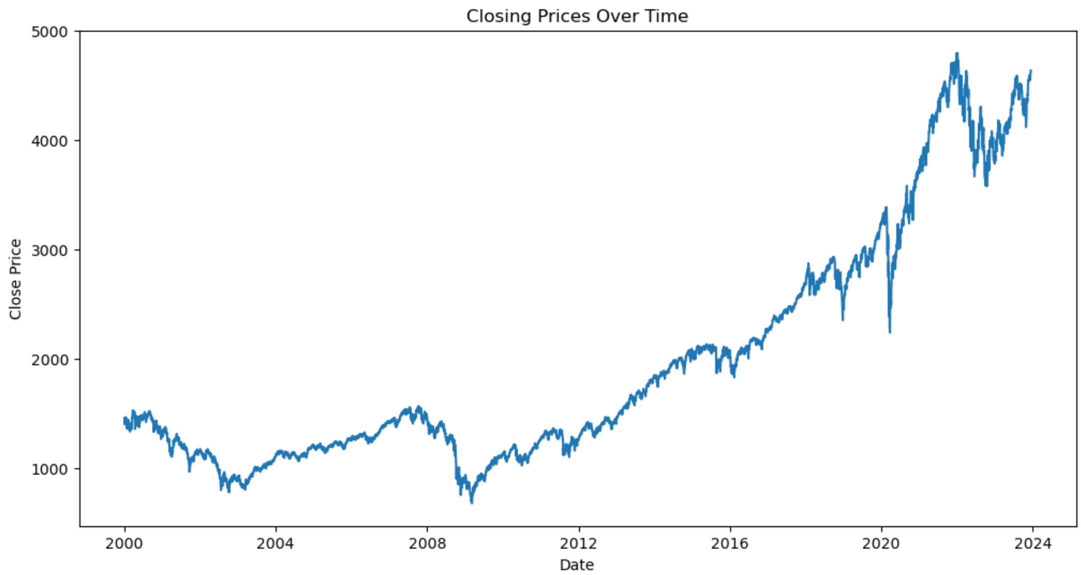

Figure 2. Closing Price Over time (2000- present)

Time series analysis is employed in both cases and timestamps are assigned to the index while close monthly and annual averages are depicted. An important milestone for each implementation is formulating the target variable. The binary dependent variable called “Target” for this purpose is derived from the relationship between today’s closing price with tomorrow’s closing price. This provides the foundation for other machine learning activities that follow.

The time series is transformed into binary format for the target variable of both timelines, and the data is prepared for classification jobs using Machine Learning Preparation. Stock trend forecasting is achieved by integrating machine learning libraries like scikit-learn and TensorFlow/Keras. The first step in this study is the development of an LSTM model. This happens in a well-defined order as it considers dataset specifics while adjusting different parameters by considering the characteristics of the finance dataset. The process begins with Create-LSTM-Model which introduces a specific function that builds up an LSTM model. There is an LSTM layer with 50 units that uses ReLU activation and a Dense layer that uses Sigmoid activation for binary classification. The optimization is done using Adam optimizer and binary cross entropy loss that are consistent with aiming at predictive model for binary outcomes. Next, feature selection and scaling with MinMaxScaler are done on (`Open`, `High`, `Low`, `Close`, `Volume`). This generates sets of data comprising of 10 data points each in form of sequences for input. these alignments are based on specific labels acquired from the “Target” column. The relevant training and testing sets that will be used in future model’s training are created by performing data splitting of the entire dataset via `train_test_split`. Splitting guarantees that the performance of the model is accurately evaluated against unseen data. Another important aspect involves using the trained LSTM model on the training data. The model is trained for different epochs ranging from 10, 50, 75, and 100 with each batch containing 32 iterations. These training epochs use the test data as a basis for validation. Notably, the LSTM model includes sentiment analysis, drawing inspiration from *(Jadhav et al., 2021)* architectural concept. Furthermore, the methodology considers behavioural factors, as highlighted in *(Parveen et al., 2020)*, providing additional context to the analysis.

Secondly, there is the initiation of a well performing Random Forest Classifier (RFC) model. There are 200 decision trees (n_estimators = 200) in the RFC model, and it will use 10 as the minimum split threshold (min_samples_split =10).

In addition, the dataset is split into the training and test set in a systematic manner. For the recession period, the last 75 rows are allocated for testing and for the timeline from 2000 to now, the last 100 rows are assigned for testing. The model also uses the "Close," "Volume," "Open," "High," "Low" columns during training as the predictor variables while the "Target" column is used as a target variable.

Once trained, the model is used to predict the test set and precisions scores are calculated and presented for review. A plot is created that reveals actual target values together with calculated ones, for improved interpretability. The methodology has a more specific part – back testing with predefined functions iterated over multiple slides of data (prediction and evaluation) in each window of data set. Dynamic approach helps understand how the model fares in various periods by aggregating individual iteration prediction outcomes.

In the next step, we feature engineers even more the model. Calculations are performed using rolling averages or "close" prices over different time periods (2, 5, 7, 10, 14, and 2, 5, 60, 250, 1000). Moreover, "tend" is a new thing which measures the average of "target" for specified intervals and makes the information gathering process more profound. Predictions are generated using the prediction function at different probability levels (0.5,0.6 and 0.7). Precision scores for every threshold are obtained through back testing, and this printout is essential in understanding the model. The Logistic Regression model starts with a fixed random number of seeds ('random_state'=1) in the earlier stage. It provides reproducible results as it applies the same procedures in all runs. The baseline is then set up using preset parameters upon which subsequent analysis is pegged.

With respect to data preprocessing and EDA, transformation of features and cleaning take place for training. EDA helps in getting an understanding of the structure of the dataset just as EDA was applied for the random forest classifier. This is followed by the creation of the target variable which is a binary class because of the relationship between the closing prices of today and tomorrow. Lastly, Logistic Regression model is implemented on specified training set with variables of Close, Volume, Open, High, and Low.

A model is then developed to make predictions and generate precision scores towards determining the efficiency and precision of the model in the designated test set. To quantify the model in changing settings, back testing procedures are used, which create a dynamic assessment approach. The additional study delves into how different threshold values, such as 0.5 and 0.6, affect important measures like precision. It also provides valuable information on how performance measures are impacted by variations in the threshold and the adaptiveness of the model.

A comprehensive phase involves evaluating and calculating metrics for three models: LSTM, RFC, and logistic regression. To validate the trained LSTM model, tests run on a separate testing dataset yield accuracy, precision, MSE, RMSE, recall, and F1-score as performance measures. This set of metrics cumulatively gives an insightful account of the performance of the model, which includes factors like total accuracy, true positives rate, average absolute prediction error, detecting relevant cases, and a balanced measure that combines precision and recall. However, the selection will be based on what suits the objectives and needs of the given machine learning problem.

This analysis is consistent with the scientific method as it concentrates on precision score and relevant measurements for assessing model capacity to represent complexity of stock price movements in the case of economic recessions.

4 Design Specification

The provided set of Python libraries covers a wide range of functionalities essential for financial data analysis and machine learning tasks. Pandas serve as a powerful tool for data manipulation and analysis, particularly through their versatile DataFrame structures. NumPy enhances numerical computing capabilities with extensive support for array and matrix operations. Matplotlib and Seaborn offer robust data visualization tools, facilitating the creation of various plots and statistical graphics. yfinance is employed to retrieve financial data from Yahoo Finance, enabling access to historical market data and stock prices. The inclusion of Scikit-learn's RandomForestClassifier is notable for its application in machine learning, specifically for classification tasks by creating an ensemble of decision trees. Scikit-learn's metrics complement the process by providing evaluation measures such as precision, mean squared error, accuracy, recall, and F1 score. The MinMaxScaler from scikit-learn is beneficial for scaling numerical features, a common requirement in many machine learning algorithms. Lastly, TensorFlow is utilized for constructing and training neural network models, incorporating the Sequential model for a linear stack of layers and the Dense layer for fully connected neural network architecture. The inclusion of the LSTM (Long Short-Term Memory) layer further extends the capability to manage sequence prediction tasks. This amalgamation of libraries forms a comprehensive toolkit suitable for fetching financial data, conducting preprocessing tasks, constructing machine learning models, and evaluating their performance. A comprehensive approach is adopted during the predictive modelling implementation whereby widely used methods and procedures appropriate for financial data modelling are applied. The main framework comprises vital libraries like pandas for convenient data management, NumPy for numeric operations, and scikit-learn for machine learning related functions. Seaborn and matplotlib enhance visualization. Moreover, the `yfinance` library is used to access recent and credible financial information required for the analysis.

A complete process of preparing the data includes fetching historical financial data, conducting EDA, and selecting useful features. This is important in predicting because data must be verifiable and relevant to what is being predicted on. However, we will be employing two different timelines, one covers the period of 2000 to date while the other spans from the Great Economic Depression Era -2007- 2009. Validation of relevant time periods that will cover exploratory data analysis, model training, and back testing need be done.

LSTM neural network is used to estimate target variable based on financial data in the process. In the stages, we first define a function called `create_lstm_model`. This function adopts sequence Api in keras to build the lstm model connecting the lstm layer having 50 units and rectified linear unit as the activation function. Then, a single-dense layer with a sigmoid function follows. It complies with the use of the Adam optimizer and binary cross entropy loss.

After stating the model definition, data preprocessing starts. Key features like "Close," "Volume," "Open," "High," and "Low" undergo Min-Max scaling in terms of financial data. Subsequently, the LSTM model is trained on sequences of data which range from 10. The creation of these sequences together with the relevant target values is intended for training purposes. Using "train_test_split" function from scikit-learn, the dataset is split into training and testing sets. Finally, the LSTM model is trained with different number of epochs (10, 50, 75 and 100). Different measures of model performance are used such as accuracy, precision, MSE, RMSE, recall, and F1 score. Differences in epoch values make it possible to look at the way a model's performance transforms due to changes in the duration of training. The final aspect involves visualization. The actual vs. using matplotlib, predicted values of the target variable are plotted. The graph provides a direct comparison between the forecast and the fact for the given duration showing LSTM's forecasting ability on financial time series data.

A random forest classifier is an available machine learning model widely employed when looking at several different scenarios such as financial data analysis. Ensemble techniques like the random forest models use a mixture of several decision trees to ensure accurate projections on a variety of issues. For this realization, the model is set with fundamental attributes, which include the number of estimators, minimal number of splits, a random number used for repeatability, amongst others. 'Close,' "Volume," "Open," "High" and "Low" are basic metrics that form an integral part in the analysis of trend or behaviour of the market. Historical data is used in the training process making the model learn relations among the features which it later employs in prediction. The second testing stage considers the last one hundred observations and assesses how well the model performs on unseen observations or data.

An additional back testing feature is added in the model. This means that they generate forecasts at regular intervals, providing information about how well the model performs at different points of time. In this context, back testing is important in determining how effectively the model would have performed historically as well as its practical relevance for real world application. Feature engineering is introduced onto the model to further improve its predictive abilities. New predictor variables like rolling averages, ratios and trends get calculated and integrated. The model is flexible enough to manage more parameters, and the domain knowledge enhances the predictive power of the model. In this regard, prediction thresholds are adjusted to show model sensitivity toward diverse classification standards. As indicated by these measured parameters such as precision scores MSE and RMSE, one can readily tell that these models are quite accurate.

The Logistic Regression model is employed in this implementation for predicting monetary information. Logistic Regression is an extensively used type set of rules suitable for binary effects. In this context, the model is educated the usage of ancient information, with key functions which include "Close," "Volume," "Open," "High," and "Low" acting as predictors and the "Target" variable representing the binary class. After training, predictions are generated and evaluated for the usage of precision scores. A back testing mechanism is also conducted, presenting insights into the version's performance over different time intervals.

Feature engineering is brought to beautify the version's predictive abilities. Rolling averages, ratios, and traits are calculated and incorporated as new predictor variables, highlighting the model's adaptability to additional features. A threshold evaluation is achieved by adjusting prediction thresholds (0.5, 0.6), and various metrics, together with precision rankings, suggest squared blunders (MSE), and root mean squared errors (RMSE), are computed.

The models' performance is assessed using a wide range of test metrics comprising precision, recall, F1 score, mean squared error, and root mean squared error. In general, this facilitates a refined appreciation of their strong points and shortcomings. This aims at creating strong predictive models that will be able to forecast the S&P 500 index movement within this period under the careful planning and implementation of the predictive modelling process.

5 Implementation

Historical data on the index of S&P 500 are fetched by utilization of Python together with Pandas, NumPy, Matplotlib, Seaborn, yfinance, and scikit learn. For both timelines, dividends and stock split columns are taken out to simplify the dataset. The next focal point is EDA and caters according to the characteristics of each timeline. The dataset is narrowed down appropriately for the Recession period, while descriptive statistics and visualizations give an overview of the financial situation during a difficult period. For the 2000 to present timeline, data is filtered from 1/1/2000 until a similar series of statistical measures and visualizations are produced. Both timelines follow similar routes for time series analysis. Timestamping the index, the monthly and annual averages in terms of close price are depicted.

The creation of the target variable is important in both the implementations. The relationship between todays and tomorrow's closing prices gives rise to a binary target variable called "Target." The binary classification approach serves as a foundation for subsequent machine learning operations. The target variable is set up in a binary format for both timelines while the data are transformed for the purpose of classification tasks by using Machine Learning Preparation. Integration of machine learning libraries, such as scikit-learn, which can be used for forecasting stock trends.

However, visualizations are important in both methods, giving a pictorial interpretation of the financial data for certain periods. Python, Pandas, NumPy, Matplotlib, Seaborn, scikit-learn or Tensorflow / Keras for deep learning are tools employed. Together, they facilitate better analyses of the data and its visualization. A detailed use of Long short-term memory (LSTM) models in forecasting stock markets' movement. (Figure 3) It is organized into different main stages specifically adapted to each data set and parameters of the models fitted accordingly to the features of the financial data.

The process commences with a section called LSTM Model Creation, which includes a function known as `create_lstm_model` that constructs an LSTM model. The model has an LSTM layer of 50 units, ReLu activation function and Dense layer with Sigmoid activation for Binary classification. The Adam optimiser used in conjunction with binary cross-entropy loss optimises this model.

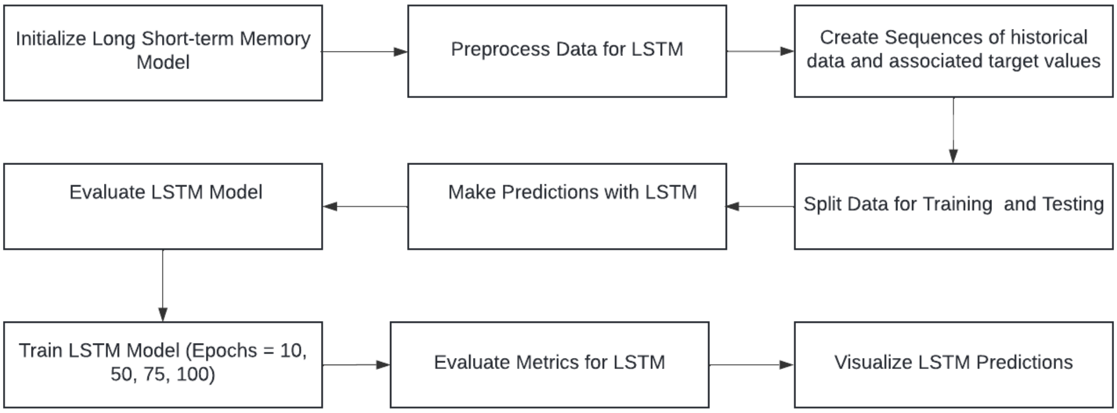

Figure 3: Flow chart of Long short-term Memory (LSTM)

In relation to Data Preprocessing for LSTM, selects features (`Close`, `Volume`, `Open`, `High`, `Low`) is scaled using MinMaxScaler. Afterwards, groups of data containing ten data points the LSTM model generates each. Acting as input these sequences is aligned to respective target labels obtained from "Target" column above. Data splitting of the dataset is done through train_test_split to create training and testing sets for subsequent model training. Training of the LSTM Model is where the created and trained LSTM will be applied to the training data. For this specific case, the model runs through 10, 50, 75 and 100 epochs consisting of batch size of 32 where the testing data is validated. It commences with building an efficient Random Forest classifier model. We configure our RFC model using 200 decision trees (n_estimators=200), setting the minimum-split threshold at 50 (min_samples_split=50) for reliability and stability on each run. The flow chart of the model can be seen in the figure 4.

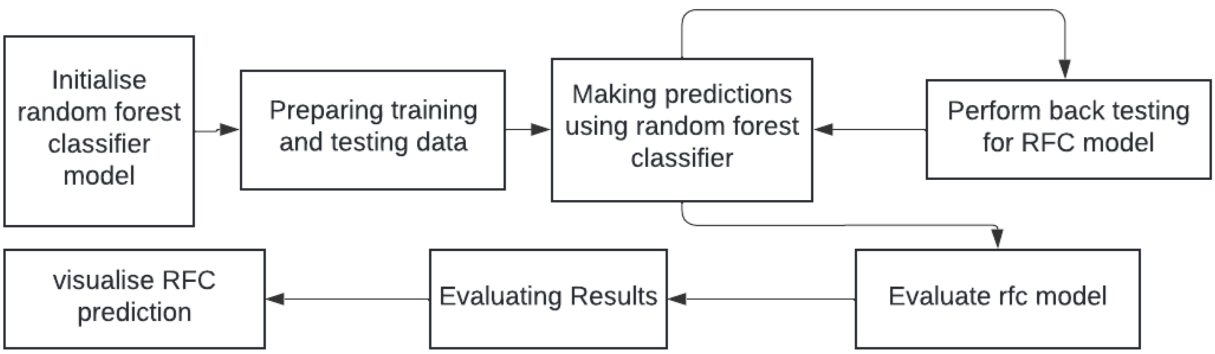

Figure 3: Flow chart of Random Forest Classifier (RFC)

The dataset is split into training and testing, respectively. Like for recession, the last 75 rows are for testing, and for 2000 to present, the last 100 rows. For this, the model is trained on training data and the “Close,” “Volume,” “Open,” “High,” and “Low” columns are used as predictors while the “Target” column becomes the target variable.

The model is used to forecast the test set, and the precision scores are computed and printed. A plot comprising the real target and estimated figures is designed for improved interpretability. Back testing is more detailed as it involves a pre-defined function that iterates through a series of predictions and assessments over each sliding window of the data set. This gives a dynamic look at how the model has performed across several periods using combined predictions from each iteration. The model is enhanced by using Feature Engineer. Computations are made using rolling averages or ratios of the “close” prices over various horizons (2, 5, 7, 10, 14 and 2, 5, 60, 250, 1000), Moreover, “tend” is an element that derives from averaging the “target” over specific intervals, thereby adding more depth into the collection of information. The `predict` function creates predictions according to multiple probability levels (0.5, 0.6, 0.7). Through back testing, a precision score for each threshold is calculated and printed, giving information on the plasticity of the model.

The algorithm uses a fixed random number of seeds (i.e., random_state = 1) to initialize the Logistic Regression model. The flow chart can be seen in figure 4. Doing so guarantees reproducibility, ensuring that results remain constant during execution of the model on other occasions. Firstly, this ensures the establishment of the Logistic Regression model with all its preconfigured settings. The following steps involve preprocessing of the data, Exploratory Data Analysis, and the Target Variable, which is closely aligned with that done in the implementation of Random Forest Classifier. The model is being trained on a training dataset, and predictions are being made for a test set. In addition, precision scores are computed, and a back-testing process is used to measure the model’s effectiveness on shifting subsets. Finally, the code attempts various threshold values including 0.5 and 0.6 to determine the impact on measures like precision, among others. The complete framework on Logistic Regression helps explain why it is capable when forecasting stock movements.

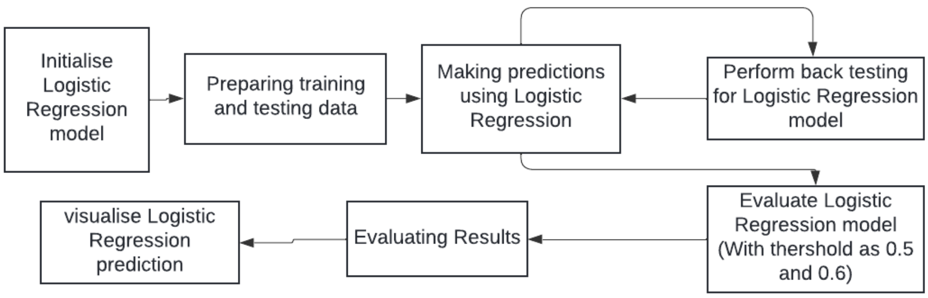

Figure 4: Flow chart of Logistic Regression

Finally, the model has a phase named “Evaluation and Calculation of Metrics” which is a crucial step in those three models that are: Long short-term memory (LSTM), Random Forest Classifier (RFC), and Logistic regression. To assess how well the trained LSTM model performed, tests were conducted on some testing data. These metrics are Accuracy, Precision, MSE, RMSE, Recall and F-Score as major measures.

6 Evaluation

6.1 The Great Recession (2007-2009)

6.1.1 Long short-term Memory

Table 3 presents the LSTM model results used on S&P500 stock price data spanning the Great Recession period, 2007-2009, providing useful information for both academic learning and practical purposes. Examining essential markers during various eras gives an insightful picture. While there is some variance in this loss, which does not show a clear downward trend, it might suggest certain difficulties in reaching convergence. Precision, which equals the ratio of true to predicted positives increases slightly from 57.33% at 10 epochs to 59.42% at 75 epochs. Nevertheless, it is also true that precision displays some degree of variation and peaks at epoch 75 which could be considered as ideal for model deployment.

Metrics	**epochs**			
	10	**50**	**75**	**100**
Loss	0.69087791	0.69131351	0.69142759	0.69352901
Precision	0.57333333	0.57432432	0.5942029	0.57377049
MSE	0.42666667	0.42666667	0.4	0.45333333
RMSE	0.65319726	0.65319726	0.63245553	0.67330033
Accuracy	0.57333332	0.57333332	0.60000002	0.54666667
Recall	1	0.98837209	0.95348837	0.81395349
F1 score	0.72881356	0.72649573	0.73214286	0.67307692

Table 3. LSTM results (2007-2009)

The MSE and RMSE are important for measuring the total model accuracy. They also vary with epochs, with the lowest ones being found at the 75th epoch. However, on recall, a measure of how well the model can find its positives falls from 100% at the tenth epoch to 81.4% at one hundred epochs. Precision-recall scores range from 72.9% to 73.2%, depending on an epoch.

6.1.2 Random Forest Classifier

By examining the RFC model's performance, we observe some relevant results and lessons regarding predicting S&P500 stock prices for the period between 2007-2009 considering the Great Recession. Variability is observed while analysing precision across different threshold values. Without threshold, it reaches 52.04%, but improves to 62.67% at threshold of 0.7. The MSE and RMSE are indicators of general model performance that were minimized at a 0.5 threshold which demonstrates higher accuracy at that point as we can see in table 4.

Table 5 shows, the model attains an accuracy of 62.67 percent, which is quite remarkable in terms of correct forecasts. It gets into recall, a measure of identifying positives, to the level of 83.33%. Sensitivity is very strong to the true cases (positives). A 'F1 score', an average of precision and recall, is quite balanced being at 0.71.

Threshold Value	precision	MSE	RMSE
No Value	0.520446	0.503953	0.709896
Threshold = 0.5	0.576923	0.452138	0.672412
Threshold = 0.6	0.596774	0.484725	0.696222
Threshold = 0.7	0.626667	0.539715	0.734653

Table 4. Random Forest Classifier (RFC) results (2007-2009)

Accuracy	Recall	F1 Score
0.626667	0.833333	0.714286

Table 5. Random Forest Classifier (RFC) model Performance

Regarding the practicality, the RCF model manifests adequate predictive performance in the recessionary economy. Because of its high accuracy and recall, it is very useful to capture positive instances, especially if we consider a threshold value equal to 0.7, which guarantees maximum precision. Threshold selection may be considered critical as it influences on precision, recall and eventually the model accuracy.

6.1.3 Logistic Regression

In the table 6, one can see the results of the logistic regression model which was applied to S&P500 stock price data during the Great recession. In this case, model accuracy refers to correct predictions as a proportion of all outcomes, which is equal to 56%. The recall of this model is 100% showing that the model detects all the positive cases. It shows reasonable balance between precision and recall having F1=0.72 score.

Accuracy	Recall	F1 Score
0.56	1	0.717949

Table 6. Logistic Regression model Performance

Without any threshold as shown in table 7, precision stands at 53.4%, while when a threshold of 0.5 is applied, precision rises to 56.1%. The general model performances are measured by using the mean squared error (MSE) and root mean square error (RMSE); the lowest recorded scores are for MSE = 0.6359842 and RMSE = 0.7944605 This indicates that the Logistic Regression model has some relevance for predicting stock price movements in the wake of the Great Recession which can be improved if a level threshold of 0.5 is employed. The findings on the dynamics within financial markets can be helpful for practitioners' decisions concerning financial market forecasts for the period of economic decline.

Threshold Value	Precision	MSE	RMSE
no value	0.533605	0.466395	0.682931
threshold = 0.5	0.56051	0.456212	0.675435
threshold = 0.6	0.513514	0.529532	0.727689

Table 7. Logistic Regression results (2007-2009)

6.2 2000 to Present.

6.2.1 Long short-term Memory

LSTM model outcomes from S&P500 stock price data, running from 2000 up to present. Looking at the loss by different epochs gives a gradual increase from 0.6917 at ten epochs to 0.6954 at one hundred epochs as shown in table 8. It indicates possible obstacles into arriving at convergence or errors reduction throughout a series of trainings.

Metrics	epochs			
	10	50	75	100
Loss	0.691685379	0.69372642	0.694603801	0.695401669
Precision	0.532003325	0.531192661	0.530941704	0.533841754
MSE	0.467996675	0.475477972	0.474646717	0.472984206
RMSE	0.684102825	0.689549108	0.688946091	0.687738472
Accuracy	0.532003343	0.524522028	0.525353312	0.527015805
Recall	1	0.9046875	0.925	0.875
F1 Score	0.694519805	0.669364162	0.674643875	0.663114269

Table 8. LSTM results (From 200 to present)

Importantly, a critical factor, precision, which represents proportion of true-positives among forecasted-positive items, varies slightly from epoch to epoch being between 53.2 percent and 53.4 percent. This means that on every occasion there remains a steady degree of correct detection of positive cases. MSE and RMS are among those highly valuable parameters for general model assessment which have relatively small changes depending on the epoch. The difference is not huge, but the trends indicate almost stable showing cross epochs.

The accuracy changes slightly from 52.7 percent, at ten epochs, to about 52.7 percent at 100 epochs. The small fluctuation could imply that the model still correctly classifies instances with minimal change throughout training. The model’s ability to find positive examples is recorded as recall which declines from 100% at 10 epochs to 87.5% at 100 epochs. The reducing phenomenon also indicates problems that may arise when trying to collect each one of these good events as epochs come on.

Like the above trend, the F1 score (involving in a balance between precision and recall) declines, starting with a 69.5% at about 10 epochs till 66.3% near 100 epochs. This highlights the balance between accuracy and recall and the requirement of choosing the appropriate epoch value depending on intended targets.

6.2.2 Random Forest Classifier

The evaluation of the Random Forest Classifier (RFC) model, applied to S&P500 stock price data from 2000 to the present. Precision analysis reveals variations across different threshold values, with the highest precision observed at 53.33% when using a threshold of 0.6. The mean squared error (MSE) and root mean squared error (RMSE) metrics indicate the model's overall performance, with the lowest values observed at the 0.5 threshold, suggesting improved accuracy at this point (Table 9).

Threshold Value	Precision	MSE	RMSE
No Value	0.522542	0.519433	0.720717
Threshold = 0.5	0.533252	0.492472	0.701764
Threshold = 0.6	0.536866	0.522979	0.723173
Threshold = 0.7	0.4	0.536054	0.732157

Table 9. Random Forest Classifier (RFC) results (From 200 to present)

In terms of accuracy, the RFC model achieves a rate of 48%, indicating the proportion of correctly predicted instances. The recall, measuring the model's ability to identify positive instances, stands at 69.23%, signifying a substantial sensitivity to actual positive occurrences. The F1 score, which balances precision and recall, is 0.58, suggesting a reasonable trade-off between these two crucial metrics. (Table 10)

Accuracy	Recall	F1 Score
0.48	0.692308	0.580645

Table 10. Random Forest Classifier (RFC) model Performance

The RFC model demonstrates satisfactory predictive performance during the challenging Great Recession period, showcasing a notable balance between precision and recall. Practitioners may find value in the model's ability to effectively capture positive instances, particularly when employing a threshold of 0.6, which maximizes precision. However, the choice of the threshold significantly influences precision, recall, and overall model accuracy, underscoring the importance of careful threshold selection in real-world applications.

6.2.3 Logistic Regression

Different threshold values result in varying precisions, giving the highest precision of 68.69% with a threshold value of 0.6. This implies that MSE as well as RMSE measures of overall model performance are lowest at 0.5, which shows better accuracy at this value (Table 11).

Threshold Value	Precision	MSE	RMSE
No Value	0.535658	0.464342	0.681427
Threshold = 0.5	0.539669	0.459588	0.677929
Threshold = 0.6	0.626866	0.528922	0.72727
Threshold = 0.7	0.4	0.536054	0.732157

Table 11. Logistic Regression results (From 200 to present)

As regards to the precision, recall and F-score, the model attains an accuracy of 52%, which suggests the proportion of rightly predicted cases. A recall of 100% means that it is very sensitive towards the actual positive cases. Balancing the precision and the recall, the F1 score stands at 0.68, signifying an acceptable compromise of these significant factors (Table 12).

Accuracy	Recall	F1 Score
0.52	1	0.684211

Table 12. Logistic Regression model Performance

The model's ability to capture positive instances may be useful to practitioners particularly for a 0.6 threshold that boosts precision. This choice of threshold greatly affects precision, recall, and model accuracy altogether and shows its relevance for actual applications.

On conclusion, logistic regression comes out as essential in predicting stock behavior in recession with precision, accuracy, and recall ratio. This has implications for stakeholders making financial market prediction and provides basis for continuous improvements as well as refinements within application and decision-making process.

7 Discussion

7.1 The Great Recession (2007-2009)

In the comparative analysis of these models, it becomes apparent that each model brings unique strengths and presents distinct trade-offs. The Long Short-Term Memory (LSTM) model stands out for its ability to capture temporal dependencies, a crucial aspect in predicting stock price movements. However, the observed challenges in achieving convergence suggest a need for careful consideration of model architecture and tuning parameters to unlock its full potential. Exploring different LSTM architectures and addressing convergence issues may lead to improved performance, making it a more robust tool for capturing nuanced patterns in financial data.

On the other hand, the Random Forest Classifier (RFC) model exhibits robust overall performance, particularly in effectively capturing positive instances. This underscores the importance of thoughtful threshold selection, where the model achieves notable precision values. The model's strength in this area is pivotal for stakeholders who prioritize minimizing false positives. While the RFC model showcases strong capabilities, further refinement through hyperparameter tuning could unlock additional predictive capabilities. Optimizing the model's parameters may enhance its adaptability to diverse market conditions, contributing to its effectiveness in real-world financial forecasting scenarios.

The Logistic Regression model, positioned between the LSTM and RFC models, strikes a commendable balance between precision and recall. The model's flexibility and interpretability make it a valuable tool, especially with the observed improvements in accuracy through threshold adjustments. However, to elevate its performance further, exploring feature engineering strategies and fine-tuning model parameters would be beneficial. Integrating additional relevant features and refining the existing ones could contribute to a more nuanced understanding of the complex dynamics within financial markets, potentially leading to enhanced predictive accuracy.

In the pursuit of continuous improvement, all models could benefit from rigorous testing, cross-validation, and robustness assessments. Incorporating advanced techniques such as ensemble methods or experimenting with novel architectures may open new avenues for enhancing predictive capabilities. Additionally, staying attuned to the evolving landscape of financial markets and considering external factors that may impact stock prices could further refine the models' adaptability and resilience.

7.2 2000 to Present.

The LSTM model's gradual increase in loss suggests challenges in convergence or error reduction during training. This could be attributed to the complexity of capturing long-term dependencies in stock price data. To enhance performance, exploration of alternative architectures, experimenting with different hyperparameters, and incorporating external factors such as economic indicators may offer avenues for improvement. Additionally, investigating the impact of sequence length on model outcomes could provide insights into optimizing input data.

Moving to the Random Forest Classifier (RFC) model, it demonstrates notable precision, particularly when employing a threshold of 0.6. This precision is crucial in financial decision-making, where false positives can have significant consequences. However, the sensitivity of precision to threshold variations underscores the importance of careful threshold selection based on the specific goals of the application. Further experimentation with different threshold values and considering the trade-offs between precision and recall may provide a more nuanced understanding of the model's behavior.

The Logistic Regression model exhibits sensitivity towards positive cases, as indicated by a recall of 100%. The model's balanced compromise between precision and recall, as reflected in the F1 score, makes it a valuable tool for predicting stock behavior during economic downturns. However, the choice of the threshold significantly influences model outcomes. Practitioners must carefully consider the implications of different threshold values based on the desired balance between precision and recall.

The trade-offs between these metrics necessitate a nuanced understanding of the specific requirements of financial forecasting tasks. Achieving high precision may be crucial in risk-averse scenarios, while a balanced compromise between precision and recall could be more suitable in dynamic and uncertain financial markets.

8 Conclusion and Future Work

The study aimed to assess the predictive ability, consistency, and suitability of Random Forest Classification (RFC), Long Short-Term Memory (LSTM) networks, and Logistic Regression models in forecasting S&P 500 stock prices, encompassing market situations such as the Great Recession of 2007-2009. The findings provide valuable insights into the strengths and weaknesses of each model. The LSTM model excels in capturing temporal dependencies, the RFC model demonstrates high precision for positive instances, and the Logistic Regression model strikes a commendable balance between precision and recall.

While the LSTM model faces challenges in convergence, suggesting the need for exploration of alternative architectures and parameter tuning, the RFC model's robust performance highlights the importance of parameter selection. The Logistic Regression model's interpretability and balanced performance suggest avenues for improvement through feature engineering and parameter tuning.

The study identifies key findings and trade-offs, revealing insights into the nuances of predicting stock behavior from 2000. Despite LSTM convergence challenges, it provides valuable information on capturing long-term dependencies. Future work could explore alternative architectures, hyperparameter experimentation, and the inclusion of external economic indicators. Additionally, investigating the impact of sequence length on model results may offer optimization opportunities.

The RFC model's precision at a threshold of 0.6 is crucial in financial decision-making. Future research may involve experimenting with various threshold values to understand the model's behavior and the trade-offs between precision and recall. The Logistic Regression model, sensitive to positive cases, proves valuable for predicting stock behavior during economic downturns. The balance between precision and recall emphasizes the significance of practitioners carefully considering different threshold values.

Future research should prioritise refining and enhancing these models for improved real-world applicability. Rigorous testing, cross-validation, and robustness assessments are crucial for validating and fine-tuning the models. Exploring advanced techniques like ensemble methods or hybrid models could offer synergies to amplify predictive capabilities.

To extend the research, incorporating external factors and economic indicators into the models may provide a more holistic understanding of stock price movements. Adapting the models to different economic contexts and assessing their adaptability across diverse market conditions are crucial for generalizability.

Addressing limitations, such as LSTM convergence challenges and model sensitivity to threshold values, opens avenues for meaningful future research. Developing dynamic threshold selection strategies that adapt to changing market conditions could enhance the models' adaptability.

The practical implications of these models in financial decision-making necessitate ongoing refinement. Continuous monitoring of model performance, adaptation to evolving market dynamics, and integration with domain expertise are essential for sustained relevance. In terms of commercialization potential, once refined and validated, these models could be integrated into financial decision-support systems. Their ability to provide nuanced insights into stock behavior during economic downturns positions them as valuable tools for investors, analysts, and financial institutions.

While this study has advanced our understanding of LSTM, RFC, and Logistic Regression models in predicting stock behavior, there is room for further exploration and refinement. The proposed future work aims to enhance the models' adaptability, interpretability, and real-world applicability, paving the way for continued advancements in financial forecasting.

References

Fu, J., Wu, X., Liu, Y., Chen, R., 2021. Firm-specific investor sentiment and stock price crash risk. Finance Res. Lett. 38, 101442. https://doi.org/10.1016/j.frl.2020.101442

González-Rivera, G., Arroyo, J., 2012. Time series modeling of histogram-valued data: The daily histogram time series of S&P500 intraday returns. Int. J. Forecast. 28, 20–33. https://doi.org/10.1016/j.ijforecast.2011.02.007

Jadhav, R., Sinha, S., Wattamwar, S., Kosamkar, P., 2021. Leveraging Market Sentiment for Stock Price Prediction using GAN, in 2021 2nd Global Conference for Advancement in Technology (GCAT). Presented at the 2021 2nd Global Conference for Advancement in Technology (GCAT), IEEE, Bangalore, India, pp. 1–6. https://doi.org/10.1109/GCAT52182.2021.9587497

Kamal, S., n.d. An Analysis of Machine Learning Techniques for Economic Recession Prediction.

Kroencke, T.A., 2022. Recessions and the stock market. J. Monet. Econ. 131, 61–77. https://doi.org/10.1016/j.jmoneco.2022.07.004

Lohrmann, C., Luukka, P., 2019. Classification of intraday S&P500 returns with a Random Forest. Int. J. Forecast. 35, 390–407. https://doi.org/10.1016/j.ijforecast.2018.08.004

Martínez-Sánchez, J.C., Berrones-Santos, A., Martínez, J.A., 2023. Markowitz's Mean–Variance Interpretation under the efficient market hypothesis in the context of critical recession periods. J. Comput. Appl. Math. 434, 115227. https://doi.org/10.1016/j.cam.2023.115227

Mateusz, K. and Ślepaczuk, R. (2020) PREDICTING PRICES OF S&P500 INDEX USING CLASSICAL METHODS AND RECURRENT NEURAL NETWORKS, wWW.WNE.UW.EDU.PL. Available at: https://www.wne.uw.edu.pl/files/6215/9765/7140/WNE_WP333.pdf

P H, H., Rishad, A., 2020. An empirical examination of investor sentiment and stock market volatility: evidence from India. Financ. Innov. 6, 34. https://doi.org/10.1186/s40854-020-00198-x

Papaioannou, P., Dionysopoulos, T., Russo, L., Giannino, F., Janetzko, D., Siettos, C., 2017. S&P500 Forecasting and trading using convolution analysis of major asset classes. Procedia Comput. Sci. 113, 484–489. https://doi.org/10.1016/j.procs.2017.08.307

Parnes, D., 2020. Exploring economic anomalies in the S&P500 index. Q. Rev. Econ. Finance 76, 292–309. https://doi.org/10.1016/j.qref.2019.09.012
Parveen, S., Satti, Z.W., Subhan, Q.A., Jamil, S., 2020. Exploring market overreaction, investors' sentiments, and investment decisions in an emerging stock market. Borsa Istanb. Rev. 20, 224–235. https://doi.org/10.1016/j.bir.2020.02.002
Vogl, M., 2024. Chaos measure dynamics in a multifactor model for financial market predictions. Commun. Nonlinear Sci. Numer. Simul. 130, 107760. https://doi.org/10.1016/j.cnsns.2023.107760
Xiao, D., Su, J., 2022. Research on Stock Price Time Series Prediction Based on Deep Learning and Autoregressive Integrated Moving Average. Sci. Program. 2022, 1–12. https://doi.org/10.1155/2022/4758698