ICBiv

e ISBN 978-967-2776-41-3

International Conference in Business Management & Innovation (ICBiv) 2023

“Leveraging Innovativeness Towards Sustainability”

18 - 19 SEPTEMBER 2023

Proceeding Book

Editor

Ts Dr Umol Syamsyul Rakiman
Profesor Madya Dr Hazliza Haron
Madam Halimatussaadiah Iksan
Madam Nurul Fatihah Wan Ismail
Dr Wan Maisara Wan Mohamad
Dr Nawal Esa Yazid Esa

Department of Business and Management, UiTM Perak Branch

INTERNATIONAL CONFERENCE ON BUSINESS MANAGEMENT & INNOVATION

(ICBiv) 2023

Leveraging Innovativeness towards Sustainability

Proceeding Book
(Published November 2024)

Editor

Ts Dr Umol Syamsyul Rakiman
Profesor Madya Dr Hazliza Haron
Madam Halimatussaadiah Iksan
Madam Nurul Fatihah Wan Ismail
Dr Wan Maisara Wan Mohamad
Dr Nawal Esa Yazid Esa

Perpustakaan Negara Malaysia Cataloguing in Publication Data
No e ISBN: 978-967-2776-41-3

Cover Design: Ts Dr Umol Syamsyul Rakiman
Typesetting : Ts Dr Umol Syamsyul Rakiman
Dr Wan Maisara Wan Mohamad

Cawangan Perak
INSPIRED
INTERNATIONAL CONFERENCE ON BUSINESS MANAGEMENT AND INNOVATION 2023
https://icbiv2023.wixsite.com/icbiv2023
ICBiv 2023 INTERNATIONAL CONFERENCE ON BUSINESS AND INNOVATION

STOCK PRICE PREDICTION USING MACHINE LEARNING: EVIDENCE FROM PAKISTAN STOCK EXCHANGE

[1] Hassan Raza
[1] Associate Professor Department of Management Sciences, SZABIST, Islamabad, Pakistan
[1] Email: dr.hassan@szabist-isb.edu.pk

ABSTRACT

This study explores the application of machine learning techniques to predict the movement of stock prices in the market. Using a dataset of historical stock prices (KSE-100 Index), a Random Forest classifier is employed to make predictions about whether a stock will rise or fall in the future. The model is trained using a sliding window approach and is evaluated using precision, recall, and F1-score metrics. The study also includes back testing and hyper-parameter tuning to improve the model's performance. The results show that the model achieves a precision score of 58%, an improvement from the previous score of 48%. However, the overall accuracy of the model is only 58%, indicating that further improvements are necessary. The study also suggests future directions for research, including the use of alternative data sources, sentiment analysis, and more sophisticated algorithms. The study's findings have implications for investors and financial organizations, demonstrating the potential of machine learning to make more educated investment decisions and enhance financial forecasting and analysis.

Keywords: *Machine learning; Stock prices; Random Forest; Precision; Recall, Financial Forecasting.*

1. INTRODUCTION

The stock market is a dynamic and unpredictable environment that is influenced by a wide variety of factors, including economic data, company performance, political events, and investor emotions. It is crucial for investors to be able to make accurate forecasts regarding the swings of stock prices if they want to maximize their gains and minimize their losses. Traditional methods of predicting stock prices, such as fundamental and technical analysis, have been utilized for decades, but they have limitations in terms of accuracy and dependability. Newer ways of predicting stock prices have been developed.

It has become increasingly popular to apply machine learning algorithms to financial markets as these algorithms have emerged as a potent tool for predicting stock prices. In the course of this investigation, we intend to build a machine learning algorithm with the purpose of determining whether the price of a stock will go up or down the next day. Our primary objective is to maximize the number of true positives, which refers to the instances in which the algorithm forecasts that the price will rise and the increase actually occurs. In order to reduce the amount of money wasted as a result of false positives, the error metric that we will be employing for the algorithm is precision.

The need for precise and dependable stock price projections, which are essential for investors to have in order to make well-informed decisions, is the driving force behind this study. The research team behind this study is of the opinion that machine learning algorithms have the potential to produce more accurate predictions than more conventional methods, and they intend to show that this is the case through their investigation.

In the next section, the study conducted a literature review on the subject of predicting stock prices using machine learning algorithms. A comprehensive methodology for this research is presented, including the dataset that was utilized, the selection of features, and the algorithm that was applied. In the end, the study presented the findings and discussed how those findings should be interpreted by investors and researchers in the future.

2. LITERATURE REVIEW

2.1 Review Stage

Predicting the price of a stock is a difficult endeavor that has been the subject of extensive research within the finance field. Fundamental analysis, technical analysis, and statistical modelling are the three most commonly used methods for forecasting stock prices. Predicting the stock market is a significant challenge for the financial sector. The unpredictability and risk of the market necessitate the use of modern technologies to produce viable outcomes. Stock prices are influenced by politics, investor psychology, supply and demand, natural disasters, and other factors (Qiu & Song, 2016). With the proliferation of machine learning algorithms, however, academics have begun investigating their use to make more accurate and time-efficient predictions. Artificial intelligence is the most accurate method for predicting stock prices. Machine learning can be used to predict stock market prices using Artificial Neural Network (ANN), Support Vector Machines (SVM), and Long Short-Term Memory (LSTM). The most significant developments in ANN have occurred in financial market and business applications. Numerous investors utilize ANN for accurate stock index forecasting. Deep learning can extract theoretical features from data and discover latent nonlinear relationships without econometric assumptions or human talent (Chong et al., 2017).

Using machine learning algorithms to predict stock prices is the subject of a substantial corpus of literature. Other studies have investigated more recent models such as deep learning and reinforcement learning. It enables investors and businesses to evaluate stocks quantitatively in order to make more profitable decisions. Back propagation algorithm-based machine learning models outperform other neural network topologies for stock market index prediction (Sureshkumar & Elango, 2012).

In an investigation, Yu & Yan (2020) estimate stock prices using historical data and Deep Neural Networks (DNNs). The authors developed a deep learning framework for predicting stock prices using convolutional neural network (CNN) and long short-term memory (LSTM) artificial neural networks (ANN). The model consisted of four layers: input, CNN, LSTM, and output. CNN extracted local features from input data, whereas LSTM captured temporal dependencies. The proposed model outperformed conventional machine learning and deep learning algorithms on four stock datasets. Selvamuthu et al. (2019) It is possible to develop an algorithm for stock forecasting using a large amount of data and cutting-edge technology. This facilitates the purchase, sale, and selection of securities. Machine Learning and Data Mining are similar, but Machine Learning observes and analyses new algorithms and models while Data Mining only analyses them (Sheth & Shah, 2023).

Mokhtari et al. (2021) evaluate machine learning techniques for forecasting the stock market. The authors examine the pros and cons of statistical models, regression models, and deep learning techniques. For accurate predictions, feature engineering, data pre-processing, and model selection are required. In addition, they investigate the disadvantages of using artificial intelligence for stock market forecasting, such as model interpretability and historical data overfitting. Artificial Intelligence (AI) and machine learning have the potential to enhance the accuracy of stock market predictions, but additional research is required to address their limitations.

In their research, Sheth and Shah (2023) discuss the use of machine learning algorithms for stock market forecasting. The authors discuss the importance of stock market forecasting and how it can help investors make informed decisions. The paper examines several studies that have predicted the stock market using machine learning algorithms such as decision trees, random forests, and support vector machines (SVMs). The authors compare the accuracy of these stock market prediction models and conclude that SVMs are the most accurate. In addition, they examine studies that foretell the stock market using technical indicators and social media data. The authors then propose a novel algorithm that forecasts stock prices using a combination of technical indicators and social media sentiment analysis. Using stock market data from the real world, the proposed algorithm outperforms conventional machine learning models. The authors conclude that machine learning algorithms can be useful for predicting stock market trends and that their algorithm can provide accurate forecasts to investors. (Henrique et al., 2019) Machine learning can identify patterns, regularities, and differences in statistical data by comprehending historical occurrences, past pricing, etc. The ANN, SVM, and LSTM use machine learning algorithms to predict stock prices. As stated previously, the stock market is volatile and profitable. A method that aids investors in prediction and yields the most accurate results is essential for minimizing annual investment losses.

In addition, Ravikumar and Saraf (2020) employ regression and classification algorithms to forecast the stock prices of companies. From

2015 to 2020, they gathered data on stock prices and economic indicators. Using regression techniques such as linear, Lasso, and Ridge regression and classification algorithms such as Random Forest, K-Nearest Neighbors (KNN), and SVM, the authors predicted the stock values of companies. Random Forest and SVM algorithms outperformed other regression and classification methods in predicting the stock values of companies. The authors reported that economic parameters including crude oil prices, interest rates, and inflation rates improved predictions. By combining financial and economic data, machine learning algorithms can predict stock prices, thereby assisting investors and speculators in making informed stock market decisions.

Moein Aldin et al. (2012) evaluated forecasts for the TEPIX stock price index. This study calculates the effectiveness of practical metrics such as Moving Average, RSI, CCI, MACD, etc. using stock price, volume, and interest rate data. The index of stock prices consists of the closing, high, and low prices. An ANN model with a three-layered input structure predicts stock prices. There are versions for input, concealed, and output. The output value version predicts indices of the stock market. Utilize Mean Absolute Percentage Error (MAPE), Mean Absolute Error (MAE), and Root-Mean Square Error (RMSE) to validate results. The criteria combination (160, 30, 0.1, 0.039) maximized the commercial and analytic performance of the ANN model. It is precise 94% of the time.

Several studies have focused on the application of machine learning algorithms to stock price forecasting over the past several years. This category of algorithms includes neural networks, support vector machines, decision trees, random forests, and regression models. The ability of neural networks, which are included in this class of algorithms, to identify nonlinear relationships within the data has made them especially useful for predicting stock prices.

In a study conducted by Yang et al. (2020), convolutional neural networks (also known as CNNs) and long short-term memory networks (also known as LSTM networks) were utilized. The authors discovered that their model outperformed both conventional statistical models and alternative machine learning techniques in terms of the precision of its predictions. In this article, the authors propose a deep learning-based method for predicting fluctuations in stock prices. Combining two potent deep learning models, convolutional neural networks (CNN) and long short-term memory (LSTM) networks, is the proposed method. The CNN model is used to derive features from the input data, which are then fed into the LSTM network to predict future price fluctuations. The authors conducted experiments with actual stock market data and compared the proposed method to a number of well-established machine learning algorithms. The results demonstrate that the proposed strategy outperforms conventional methods in terms of accuracy of prediction. In addition, the authors performed sensitivity analysis to determine the effect of various parameters on the accuracy of their predictions.

Shajalal et al. (2023) plan to use deep learning to predict product backorders in a separate study. A few products in their dataset have backorders, but the majority do not. To manage skewed data, the authors propose a deep neural network with many hidden layers. Using convolutional and recurrent neural network layers, the model recognizes temporal patterns and extracts features. The authors evaluate their method using precision, recall, F1-score, and accuracy. The experimental results demonstrate that the proposed method predicts product backorders on unbalanced data better than existing cutting-edge methods. The authors conclude that their method can predict product backorders and can be applied to other unbalanced datasets. Deep Residual Attention Network (DRAN) was the moniker given to the novel deep learning model that Liu et al. (2023) proposed for predicting stock prices. The DRAN is a multi-input, multi-output network that combines a convolutional neural network (CNN) with a residual learning and attention mechanism. As inputs, the model takes into account both technical indicators and financial news, and it is capable of producing both short- and long-term stock price forecasts. The authors trained and validated the model using real-world stock price datasets before comparing its results to those of other well-known machine learning models, including Support Vector Regression (SVR), Random Forest Regression (RFR), and the Long Short-Term Memory (LSTM) network. The findings revealed that DRAN outperformed all other models in terms of accuracy and consistency of prediction, indicating that it has a great deal of potential for use in the real world for stock price forecasting.

Wanjawa (2016) used reference data from the Nairobi Securities Exchange (NSE) and the New York Stock Exchange (NYSE) to predict stock indices using a feedforward complex perceptron with error backpropagation and 5:21:21:1 with 80% training data in 130,000 circulations.

Nairobi Securities Exchange and New York Stock Exchange forecasted Mean Absolute Percentage Error (MAPE) between 0.71 to 2.77 percent. The initial test showed 0.7% prototype defect. The second test compared the new model to two others. 1.83 was the lowest Root Mean Square Error (RMSE) model. MAPE forecasted a 0.71 percent stock return last time. The best model was 5:21:21:1. The highest accurate training sets required 1,000 records (80% of the dataset) trained over 130,000 times.

In addition, a feature importance analysis was conducted in order to determine which factors were most influential in determining future stock values. On the basis of the data, it appears that certain technical indicators, such as the Relative Strength Index (RSI) and the Moving Average Convergence Divergence (MACD), are more significant than others, whereas the sentiment of news articles has a relatively minor effect on the forecast performance. Overall, the research presents a novel, deep learning-based method for predicting stock prices. This strategy has potential utility for financial analysts and investors.

Despite the fact that these machine learning approaches have been quite effective, there are still many obstacles to overcome before they can be successfully applied to stock price prediction. Due to their susceptibility to rapid shifts and high levels of volatility, the availability of financial data presents a significant obstacle. In addition, market trends, economic conditions, and geopolitical events must be taken into account because they have the ability to affect stock prices.

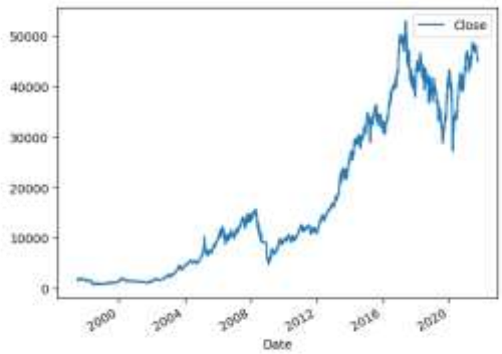
In general, the findings of these studies suggest that machine learning algorithms may be useful for predicting stock price fluctuations. To improve the accuracy of these models, it is useful to combine information from multiple sources, such as social media, technical indicators, and the tone of news articles. Optimizing the performance of a stock price prediction algorithm by maximizing precision as the error metric is one of the specific methodologies employed in this study, and it is a viable approach.

3. METHODOLOGY OF STUDY

The primary objective of this research is to design a machine learning model that can accurately forecast the movement of stock prices. The focus of this study is to develop a model that not only maximizes profits but also minimizes potential risks associated with stock trading. The intended outcome is to make profitable trades by buying stocks when the market opens and selling them when the market closes, with a high level of confidence that the price will increase or decrease. We are using the KSE-100 index as our desired portfolio in which we are interested to invest.

The historical data of portfolio index price (KSE-100) is collected from Yahoo Finance ranging from 1997 to 2023. The downloaded dataset included the Open, High, Low, Close, Volume, Dividends and Stock Splits data. The close prices data is displayed in the line graph below. The KSE 100 is an index tracking the performance of the top 100 businesses traded on Pakistan's Karachi Stock Exchange (KSE). In this line graph, the time period is shown along the horizontal axis, which extends from 1997 to 2023. The value of the KSE 100 index is plotted against time on the vertical axis. If the line is going up or down, that means the market is doing well or poorly. If the line is rising, market performance is improving; if it is falling, market performance is deteriorating. The line chart of the KSE 100 may show some spikes and dips that signify temporary market movements. They may result from national or international economic situations, political unrest, or other factors. The line's long-term trend, however, reveals the market's cumulative performance over time.

Figure 1. Line Graphs of Close Price KSE-100



The dataset is organized to make projections about future stock prices based on previous stock prices. The dataset has been divided into two sections: the training data section and the test data section. The machine learning model is trained with the help of the training data, while the performance of the model is assessed with the assistance of the test data. Therefore, we need to prepare the data before we can make any predictions regarding whether or not there will be an increase in the price of the stock tomorrow based on the data from today. Our goal is to determine whether the price of the stock goes up or down the following day, which is represented by the numbers 1 and 0 correspondingly. We will move the data from the days before "forward" by one day so that we can utilize historical data to create forecasts. This will help us avoid making the typical mistake of using the same day's data for prediction, which is quite common. In the end, we finished off the process of creating our training data set by combining the shifted data with the target variable.

4. RESULT AND INTERPRETATION

The process of training the model can then begin when the data has been partitioned into two sections. The act of teaching a machine learning model to recognize patterns in data and to make correct predictions or judgments based on those patterns is referred to as "model training." In order to set up the target, we start by making a new 'DataFrame' called 'data' and copying the 'Close' column into it. After that, we rename the column to 'actual_close' and save the changes. Because of this, we are able to monitor the real price at which the market closed each day. Next, we will build up the target by using the pandas rolling function to examine every two rows of the DataFrame. This will allow us to set up the target. This enables us to evaluate how the closing price of one day stacks up against the pricing of the following day. If the price at the end of the second trading day is higher than the price at the end of the first trading day, we will put a value of one in the Target column to indicate that the price increased. If the price at the end of the second day is lower than or equal to the price at the end of the first day, we put a value of 0 in the 'Target' column to indicate that the price went down. For the sake of this comparison, the 'Close' column will be utilized. The resulting 'Target' column has been updated to include the binary values that predicted by our machine learning model. The whole training process is discussed below;

Let X represent the historical data for the stock price, and let Y represent the target variable that indicates whether or not there is an increase or decrease in the stock price tomorrow (1 or 0, respectively). This can be expressed mathematically as follows:

$$X = \{x_1, x_2, x_3, \ldots, x_n\}$$

$$Y = \{y_1, y_2, y_3, \ldots, y_n\}$$

where n refers to the total number of data points in the collection.

In order to get the data ready for machine learning, we have to move the data from the days before "forward" by one day so that we are not making predictions based on the same-day data. This may be represented as:

$$X\_shifted = \{x_2, x_3, x_4, \ldots, x_n, x_{n+1}\}$$

$$Y\_shifted = \{y_2, y_3, y_4, \ldots, y_n, y_{n+1}\}$$

where X_shifted and Y_shifted represent the shifted data.

We may generate our own training dataset by combining the data that has been shifted with the variable that represents the target, and this dataset can be described as follows:

Training data:

$$Training\ data: \{(x_2, y_2), (x_3, y_3), (x_4, y_4), \ldots, (x_n, y_n)\}$$

To summarize, the following equation can be used to represent the process that was stated in the previous paragraph:

$$Training\ data: \{(X\_shifted[i], Y\_shifted[i])\}$$

for i = 1 to n-1

Training data: (X_shifted[i],Y_shifted[i) for i = 1 to n-1, where X_shifted[i] represents the ith element in the shifted X data and Y_shifted[i] represents the ith element in the shifted Y data. n is the total number of elements in the data set.

After training the model, the first 10 values of the data are showed in the table, it clearly shown if the values are increasing, the target value is given as 1 and when market decreased, it reported as value of zero.

Table 1. Data Training

Date	Actual_Close	Target
1997-07-02	1618.15002	NaN
1997-07-03	1648.84998	1
1997-07-07	1691.68005	1
1997-07-08	1726.17004	1
1997-07-09	1778.51001	1
1997-07-10	1745.34998	0
1997-07-14	1794.64002	1
1997-07-15	1810.83997	1
1997-07-16	1812.78003	1
1997-07-17	1845.30005	1

Now in the next step, we are going to apply the shift method to the DataFrame at this time, which will move all of the rows "ahead" by one business day. Because of this, the prices that were listed for 1997-07-02 are now associated with 1997-07-03, and the positions of the prices listed after that have been moved up one row. The purpose of this is to ensure that we are accurately projecting future pricing based on historical data.

If we didn't adjust the data in this way, we would be forecasting prices for 07-03 using data from 07-03, which would be an inaccurate assumption. Instead, we need to make our price predictions for 07-03 using information from 07-02. If we do not adhere to this concept, our model might operate well during testing, but it might not work in the real world, where we do not have knowledge of tomorrow's pricing to guide our forecasts. If we do not adhere to this approach, our model might fail to work in the real world. After this adjustment in the model, here we are presenting the first 10 values of our resultant data.

Table 2. Shifting Data "Forward"

Date	Open	High	Low	Close
02/07/1997	NaN	NaN	NaN	NaN
03/07/1997	1593	1618	1590	1618
07/07/1997	1617	1649	1612	1649
08/07/1997	1674	1699	1674	1692
09/07/1997	1695	1726	1689	1726
10/07/1997	1727	1786	1711	1779
14/07/1997	1779	1784	1664	1745
15/07/1997	1757	1795	1747	1795
16/07/1997	1797	1830	1795	1811
17/07/1997	1808	1818	1803	1813

The above table clearly indicates that data has been shifted forward by one day, as mentioned earlier. The first row now contains NaN values, as there is no previous day's data to shift to this row. The other rows contain the prices and other information for each day.

The next thing we need to do is combine the target variable with the columns that will be used to predict the outcome. In order to accomplish this goal, we will use the join method on DataFrames. As soon as we joined all of our data together, we realized that in order to predict the objective variable, we are utilizing data from the day before. The columns "Close," "Volume," "Open," "High," and "Low" are the ones that are retained to make a prediction about our objective. Table no 3 represents first 10 values of our trained data, after incorporating all the previous steps, and now data is ready for creating our machine learning model.

Table 3. Final Training Data for Machine Learning Model

Dates	Close	Target	Close	Open	High	Low
3 July	1649	1	1618	1593	1618	1590
7 July	1692	1	1649	1617	1649	1612
8 July	1726	1	1692	1674	1699	1674
9 July	1779	1	1726	1695	1726	1689
10 July	1745	0	1779	1727	1786	1711
14 July	1795	1	1745	1779	1784	1664
15 July	1811	1	1795	1757	1795	1747
16 July	1813	1	1811	1797	1830	1795
17 July	1845	1	1813	1808	1818	1803
21 July	1912	1	1845	1812	1845	1809

4.1 Setting Up the Machine Learning Model, Random Forest Classifier

The following step is to apply machine learning techniques in order to develop a trustworthy model for predicting stock prices. We have also made our way here in order to use the random forest classifier for the same purpose. Because it can detect nonlinear correlations in the data and is reasonably resistant to overfitting provided that the appropriate parameters are used, a random forest classifier is an excellent choice as a default model for many applications. This is due to the fact that it may be used to classify the data.

The Random Forest Classifier is a method of machine learning that generates forecasts through the utilization of a forest of decision trees. The following econometric equation can be used to represent the Random Forest Classifier algorithm:

$$y = f(X, \theta) + \varepsilon$$

Where: y is the binary target variable that represents the increase or decrease in market prices of the portfolio (1 if the price went up, 0 if it went down), X is a matrix of predictor variables that includes past returns portfolio data, is a vector of unknown parameters that the algorithm seeks to estimate through training, and is the error term that represents the difference between the predicted values and the actual values. y is a binary target variable that represents the increase or decrease in market prices of the portfolio.

The random forest model is trained on the data, and the function f(X, θ) is the model's output for making predictions about the target variable. It is a collection of decision trees, with each tree being constructed from a different random subset of the predictor variables and a different random sample of the data. The results of each tree are added together before being used in the final prediction.

The following are the specific parameters of the model that are being used in this equation: n_estimators equals 100, which is the number of decision trees that make up the ensemble; min_samples_split equals 200, which is the bare minimum amount of samples that must be collected in order to split a node in a decision tree; random_state equals 1, which is the random seed that ensures repeatability.

4.2 Training the Model

When we've finished configuring the model, we'll be able to train it using the most recent one hundred rows of the dataset. For the purpose of predicting the next price, we are use all of the data, with the exception of the most recent 100 rows. When working with data from a time series, it is imperative that you never utilize information from the future to try to understand the past.

The fit technique educates the model by training it to make accurate predictions using our predictors. The following is a representation of this process that may be made using econometric equations:

Let Y, the future price of the KSE-100 index, be the goal variable representing an increase or reduction in the future (1/0), and let X, the behavior of past prices, serve as the predictor variables. Let Y be the target variable. We have historical data for the past T time periods, which are designated as (Y_t, X_t) for t = 1, 2, ..., T. These data cover the entirety of the time period in question. We wish to train a random forest classifier model to predict the direction of Y for the last 100 periods, and we will use the 100 data that we have collected.

The model can be represented as:

$$Y_t = f(X_t) + e_t$$

where f(.) is the function that maps X to Y, and e_t is the error term at time t. The random forest algorithm estimates this function using an ensemble of decision trees.

To train the model, we use the fit method with the following input arguments:

$$model.fit(X[:T-100], Y[:T-100])$$

This trains the model on the previous observations of X and Y.

After training the model, we can use it to predict the direction of Y for the last 100 periods:

$$Y\_pred = model.predict(X[T-100:])$$

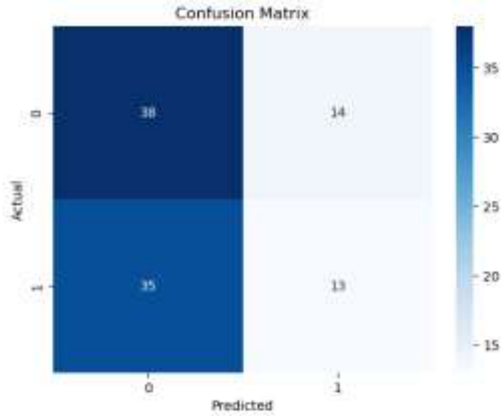
This predicts the direction of Y using the trained model on the last 100 observations of X.

4.3 Classification of The Accuracy Of The Model

We utilized the model's precision as the error metric so that we could determine how accurate it was. The ratio of the overall number of true positives to the total number of positive predictions makes up the accuracy score. A high precision score indicates that the model accurately predicted positive values; in other words, when the model projected that the stock prices would go up, they actually went up. This is the same thing as saying that the model correctly forecasted negative values.

However, the current model has a precision score of 0.48, which indicates that the model's predictions are not very accurate and should not be relied upon. This indicates that the model has to be further refined because the stock prices only went up 48% of the time when the algorithm predicted that they would go up.

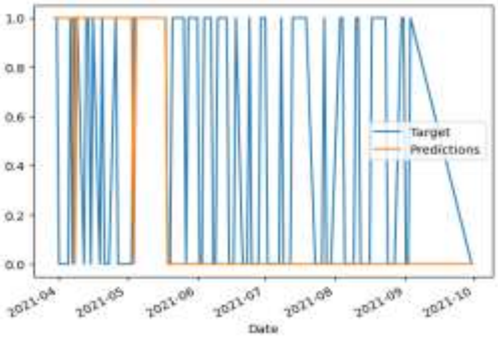
Figure 2. Confusion Matrix



The actual values are listed in the rows, while the expected values are listed in the columns. In light of the confusion matrix presented here, the model made the following predictions: 38 true negatives (TN), 13 true positives (TP), 14 false positives (FP), and 35 false negatives (FN). This indicates that 38% of the observations were correct, 14% of the observations were erroneous, 35% of the observations were false negatives, and 13% of the observations were correct.

It would appear that the model is not functioning up to expectations because it is only correct 48% of the time for the direction. We may look into this matter further by making a graph that compares the projected numbers to the actual values. The figure demonstrates that the model forecasted that the price would continue to increase each day. It does not make any predictions on the movement of the data. Therefore, despite the fact that this is not the best possible scenario, we need to make some adjustments to our model before moving on to the subsequent stage of backtesting.

Figure 3. Prediction Plot



On the other hand, this is merely the preliminary configuration of our model to check that everything is operating as it should. Backtesting our model using the full price history will allow us to derive an error metric that is representative of the true level of error. After we have finished putting the model together, the following step will be to do this.

4.4 Backtesting of the Model

It is vital to generate predictions on the whole dataset in order to improve our model rather than only focusing on the 100 most recent rows alone. By doing so, we are able to acquire an estimate of the model's inaccuracy that is more accurate; this is because the market conditions over the past 100 days may have been exceptional and may have an impact on future projections.

Backtesting, in which we only use data that was collected before the day we are attempting to predict, is something that needs to be done in order for this to be possible. It is not feasible to use data from the future to anticipate what happened in the real world in the past. Because of this, we must steer clear of using data collected after the day for which we are making a prediction.

Backtesting is accomplished by recursively going over all of the data in the dataset and training a model at intervals of 750 rows. Now that the process has been outlined, we will develop a function that will help us follow it, which will prevent us from having to rewrite the code each time we wish to run a backtest. We should train the model more frequently than every 750 rows, however for the sake of efficiency, we will increase this number. Idealistically, we should train the model more frequently than every 750 rows.

The result of performing backtesting and making adjustments to the hyper-parameters of the model yielded a new precession score of 58%. The previous score, which was just 48% accurate, has been significantly improved to 58%, representing a huge rise. Backtesting and hyper-parameter tweaking are two critical processes that must be completed in order to improve the overall performance of a machine learning model.

Following is the classification table after backtesting the model:

Table 4. Classification Table

Class	Precision	Recall	F1-score	Support
0	47%	76%	58%	2001
1	58%	28%	38%	2384
Total	53%	52%	48%	4385

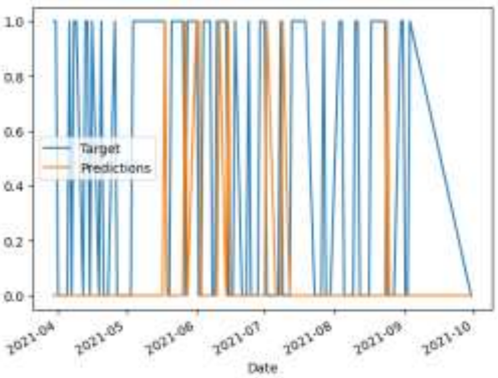
This table shows the performance of the classifier on the test set. The precision for class 0.0 is 47%, which means that when the classifier predicts a stock will go down, it is correct 47% of the time. The recall for class 0.0 is 76%, which means that of all the stocks that actually went down, the classifier identified 76% of them correctly. The F1-score for class 0.0 is 58%, which is the harmonic mean of precision and recall for that class.

The precision for class 1.0 is 58%, which means that when the classifier predicts a stock will go up, it is correct 58% of the time. The recall for class 1.0 is 28%, which means that of all the stocks that actually went up, the classifier identified only 28% of them correctly. The F1-score for class 1.0 is 38%, which is the harmonic mean of precision and recall for that class.

Overall, the classifier has an accuracy of 50%, which is not very good. The macro average of precision, recall, and F1-score is around 52%, while the weighted average is around 48%. This suggests that the classifier is performing well on either class then the previous results, but although may need further improvements.

The precision score could be enhanced with the help of the number of factors, enhancement of the historical data, selection of the more stable market, inclusion of the more factors that determine the future price, selection of the more comprehensive technique and selection of more economic indicators.

Figure 4. Prediction Plot after Backtesting



For investors and financial organizations who rely on accurate and timely projections of stock values, the results of this study have major significance. Investors can have the ability to make more educated investment decisions and potentially earn higher returns by employing techniques from the field of machine learning to conduct analyses of past data and make predictions about future prices.

This study also sheds insight on the possible applications of machine learning in financial forecasting and analysis. As research into artificial intelligence continues to make strides forward, there is a growing possibility that financial analysis will benefit from the application of machine learning algorithms, which are expected to enhance both accuracy and efficiency.

On the other hand, it is essential to keep in mind that machine learning models are not infallible and do not guarantee correct predictions in all situation. When making decisions about their investments, investors should always exercise caution and consider the forecasts generated by machine learning as only one of numerous relevant aspects.

5. FUTURE DIRECTIONS

The current model employs a Random Forest classifier, which is a machine learning method that is utilized frequently. On the other hand, there are more sophisticated algorithms such as neural networks, deep learning, and gradient boosting, all of which have the potential to improve the accuracy of the model. These algorithms are capable of capturing relationships that are more intricate between the features and the target variable.

The use of sentiment analysis of news stories and social media can provide additional information about how the market feels about a specific asset if it is done correctly. The process of employing techniques from natural language processing to extract and quantify feelings, opinions, and attitudes included within text data is referred to as sentiment analysis. A gauge of the market's optimism or pessimism about a specific asset can be obtained through sentiment analysis, which can then be utilized to improve the model's ability to make accurate forecasts.

Alternative Data Sources Satellite imagery and web scraping are just two examples of the many alternative data sources that are available and have the potential to provide useful information regarding the industry. Web scraping can provide real-time data on consumer behavior and online transactions, whilst satellite imagery can provide insights into economic activity and commodity production. Satellite imagery can also provide insights into consumer behavior. The accuracy of the model could potentially be improved by including these alternative data sources.

The existing model makes use of technical indicators as features; however, it is possible that other features, such as price-volume connections and volatility measures, can be derived from the data that is currently available and incorporated into the model. The accuracy of the model can be improved by the use of feature engineering, which provides a more thorough depiction of the dynamics of the market.

Ensemble methods involve the combination of numerous models in order to increase the overall accuracy of the forecast. For instance, the model may be integrated with a number of other classifiers or regression models in order to generate an ensemble of models that produce predictions by cooperating with one another. Ensemble approaches have the potential to lessen the likelihood of overfitting and to improve the generalization performance of the model.

6. CONCLUSION

This study intends to investigate the application of machine learning strategies to the forecasting of stock prices. Using historical data from the Pakistan stock market, we trained a Random Forest classifier to predict whether the price of a stock will increase or decrease in the future. They employ technical indicators such as moving averages and the relative strength index (RSI) as model characteristics. The tuning procedure for hyperparameters, backtesting, and model evaluation are discussed.

After completing the backtesting procedure, the results indicate that the Random Forest classifier has a precision score of 58%, which is an improvement over the previous score of 48%. However, the model's accuracy is only 58%, which still requires refinement. With the addition of more factors, the improvement of historical data, the selection of a more stable market, the inclusion of more factors that determine the future price, the selection of a more comprehensive technique, and the selection of more economic indicators, the precision score can be enhanced.

Future directions for improving the model's accuracy are also provided, including the use of more advanced algorithms such as neural networks, deep learning, and gradient boosting; sentiment analysis of news articles and social media; the use of alternative data sources such as satellite imagery and web scraping; and feature engineering. At the conclusion of the article, the author reminds readers that machine learning algorithms are not foolproof and that investors must exercise prudence when making investment decisions.

7. REFERENCES

Chong, E., Han, C., & Park, F. (2017). Deep Learning Networks for Stock Market Analysis and Prediction. *Expert Systems with Applications*, *83*(April), 187–205. http://ac.els-cdn.com/S0957417417302750/1-s2.0-S0957417417302750-main.pdf?_tid=0d300a54-78da-11e7-ab02-00000aacb35f&acdnat=1501826538_c99481212aa82d83961ec6ff566751a4

Henrique, B. M., Sobreiro, V. A., & Kimura, H. (2019). Literature review: Machine learning techniques applied to financial market prediction. *Expert Systems with Applications*, *124*, 226–251. https://doi.org/10.1016/j.eswa.2019.01.012

Liu, M., Sheng, H., Zhang, N., Chen, Y., & Huang, L. (2023). A New Deep Network Model for Stock Price Prediction. In Y. Xu, H. Yan, H. Teng, J. Cai, & J. Li (Eds.), *Machine Learning for Cyber Security* (pp. 413–426). Springer Nature Switzerland.

Moein Aldin, M., Dehghan Dehnavi, H., & Entezari, S. (2012). Evaluating the Employment of Technical Indicators in Predicting Stock Price Index Variations Using Artificial Neural Networks (Case Study: Tehran Stock Exchange). *International Journal of Business and Management*, *7*(15), 25–34. https://doi.org/10.5539/ijbm.v7n15p25

Mokhtari, S., Yen, K. K., & Liu, J. (2021). Effectiveness of Artificial Intelligence in Stock Market Prediction based on Machine Learning. *International Journal of Computer Applications*, *183*(7), 1–8. https://doi.org/10.5120/ijca2021921347

Qiu, M., & Song, Y. (2016). Predicting the Direction of Stock Market Index Movement Using an Optimized Artificial Neural Network Model. *PLOS ONE*, *11*(5), e0155133. https://doi.org/10.1371/journal.pone.0155133

Ravikumar, S., & Saraf, P. (2020). Prediction of stock prices using machine learning (regression, classification) Algorithms. *2020 International Conference for Emerging Technology, INCET 2020*, 1–5. https://doi.org/10.1109/INCET49848.2020.9154061

Selvamuthu, D., Kumar, V., & Mishra, A. (2019). Indian stock market prediction using artificial neural networks on tick data. *Financial Innovation*, *5*(1). https://doi.org/10.1186/s40854-019-0131-7

Shajalal, M., Hajek, P., & Abedin, M. Z. (2023). Product backorder prediction using deep neural network on imbalanced data. *International Journal of Production Research*, *61*(1), 302–319. https://doi.org/10.1080/00207543.2021.1901153

Sheth, D., & Shah, M. (2023). Predicting stock market using machine learning: best and accurate way to know future stock prices. *International Journal of System Assurance Engineering and Management*, *14*(1), 1–18. https://doi.org/10.1007/s13198-022-01811-1

Sureshkumar, K. K., & Elango, N. M. (2012). Performance Analysis of Stock Price Prediction. *Global Journal of Computer Science and Technology*, *12*(1), 19–26. https://computerresearch.org/index.php/computer/article/view/426/426

Wanjawa, B. W. (2016). *Evaluating the Performance of ANN Prediction System at Shanghai Stock Market in the Period 21-Sep-2016 to 11-Oct-2016*. *147*(March), 11–40. http://arxiv.org/abs/1612.02666

Yang, C., Zhai, J., Tao, G., & Haajek, P. (2020). Deep Learning for Price Movement Prediction Using Convolutional Neural Network and Long Short-Term Memory. *Mathematical Problems in Engineering*, *2020*. https://doi.org/10.1155/2020/2746845

Yu, P., & Yan, X. (2020). Stock price prediction based on deep neural networks. *Neural Computing and Applications*, *32*(6), 1609–1628. https://doi.org/10.1007/s00521-019-04212-x

Pejabat Perpustakaan
Librarian Office

Universiti Teknologi MARA
Cawangan Perak
Kampus Seri Iskandar
32610 Bandar Baru Seri Iskandar,
Perak Darul Ridzuan, MALAYSIA
Tel: (+605) 374 2093/2453 Faks: (+605) 374 2299

UNIVERSITI TEKNOLOGI MARA

Surat kami : 700-KPK (PRP.UP.1/20/1)
Tarikh : 20 Januari 2023

Prof. Madya Dr. Nur Hisham Ibrahim
Rektor
Universiti Teknologi MARA
Cawangan Perak

Tuan,

PERMOHONAN KELULUSAN MEMUAT NAIK PENERBITAN UiTM CAWANGAN PERAK MELALUI REPOSITORI INSTITUSI UiTM (IR)

Perkara di atas adalah dirujuk.

2. Adalah dimaklumkan bahawa pihak kami ingin memohon kelulusan tuan untuk mengimbas (*digitize*) dan memuat naik semua jenis penerbitan di bawah UiTM Cawangan Perak melalui Repositori Institusi UiTM, PTAR.

3. Tujuan permohonan ini adalah bagi membolehkan akses yang lebih meluas oleh pengguna perpustakaan terhadap semua maklumat yang terkandung di dalam penerbitan melalui laman Web PTAR UiTM Cawangan Perak.

Kelulusan daripada pihak tuan dalam perkara ini amat dihargai.

Sekian, terima kasih.

"BERKHIDMAT UNTUK NEGARA"

Saya yang menjalankan amanah,

SITI BASRIYAH SHAIK BAHARUDIN
Timbalan Ketua Pustakawan

nar

Setuju.

27.1.2023

PROF. MADYA DR. NUR HISHAM IBRAHIM
REKTOR
UNIVERSITI TEKNOLOGI MARA
CAWANGAN PERAK
KAMPUS SERI ISKANDAR