Stock Market Price Prediction

Mohd Farzan, Md Sahil Ali, Aman Niyazi, Asif Khan
Department of Computer Science and Engineering, Integral University Lucknow-226026 India
Supervisor - Dr. Syed Haider Abbas
Integral University

Abstract-Stock exchange trade is an important and major activity when talking about financial markets. Given the inevitable uncertainty and volatility of stock prices, investors are always looking for ways to avoid losses and predict future trends to achieve the greatest possible profit. However, so far, there cannot be denied that there is no technology that can completely accurately predict future trends in the market, but several methods have been considered to improve the model as much as possible. Due to recent progress in machine learning (ML) and deep learning (DL), many algorithms have been used to predict stock prices. This article examines five algorithms. That is, we examine K-nearest Neighbor, linear regression, vector regression support, decision tree regression, and long-term short-term memory for predicting stock prices from 12 major companies in the Indian stock market. After a comprehensive study of various aspects of the application of ML in the stock market, a comprehensive implementation was implemented as part of this research work. The study has collected and used stock price data records for 12 companies over the past seven years. This paper also illustrates other efficient and robust techniques used to predict stock exchange trends. In detail, the methodology for achieving results was gradually discussed. Additionally, we performed a detailed comparative analysis of the above services for predicting stock prices, and results displayed in easy-to-read and graphical formats to better analyze them. The conclusions of this new data-comprehensive study were presented and concluded that the DL algorithm predicts stock prices or time series beyond all other algorithms, providing results that provide a wide range of accuracy.

Keywords: Stock market, Machine learning, K-Nearest Neighbour (K-NN), Linear Regression (LR), Support Vector Regression (SVR), Decision Tree Regression (DTR), and Long Short-Term Memory (LSTM)

1. INTRODUCTION

With very unpredictable trends and high market volatility, almost every stock exchange enthusiast wants to get something. The dramatic vibrations of the stock exchange, which could be a reputation for politics, brands, could be a good hike, for example, by the pandemic phase. The above factors can strongly influence the opinions and beliefs of potential investors that lead to market trends during pregnancy. It is important to understand these possible factors that cause change, but due to eternal global change and uncertainty, it is not sufficient to develop accurate methods of forecasting trends. However, there is constant effort to develop models or algorithms that will help investors predict change more accurately than before. One of the most well-known and employed possibilities for the formation of predictive models is the use of algorithms for machine learning (ML). ML is a concept in which a computer learns or predicts things without an external program with the help of previous knowledge and training.

There are several ML algorithms that can be used to implement predictions in interdisciplinary domains, such as the stock market, power requirements, and health areas. If the stock market is considered, it is a very time-dependent area, with prices fluctuating at specified minutes. Therefore, time series analysis is a comprehensive approach. One of the broadest techniques for performing time series modeling is the ARIMA model (automatic regressive integrated mean). Because the Arima model is a linear type model, it is suitable for stock market predictions, and therefore cannot be paid attention to fluctuations in the dataset due to the high market volatility. Nevertheless, ML and data science have been extremely sophisticated in recent years, leading to the development of specific algorithms that are highly efficient for predictive analytics, no matter what the field. In the past, we have examined and discussed several methods and algorithms related to ML. In the past, many research work has been given, and some common ML algorithms have been worked to be performed for prediction. However, this paper aims to develop an ML model with five different algorithms and continue to apply it to the stock market area to predict stock market trends. Five implementation models are then compared, highlighting the optimal model based on various power metrics such as symmetric average mean absolute percentage error (SMAPE), R2 value (R square), and root mean square error (RMSE). The ML algorithm is used to form the model and perform the remaining objectives of prediction and data analysis. The primary or more primarily trains the model on a good dataset, where the model learns the specified entries and uses this past experience and knowledge to classify and predict. This special part of the dataset intended for training models is known as the training dataset. This knowledge that these models meet the past then helps to predict more accurately. Therefore, ML is gradually gaining fame for use for the sectors and people on the stock exchange market, but is based on these ML models for investing in the stock market. For identification, this paper includes 5 ML algorithms, linear regression algorithms (linear regression), Lasso and Ridge regression algorithms (SVM)-Algorithms (SVM).

2. ABOUT STOCK MARKET

2.1. Stock Exchange

A stock exchange serves as a platform where shares of publicly listed companies are bought and sold. It functions as a type of secondary market. For a company to go public and offer its shares to investors, it must secure a listing on one of the recognized stock exchanges. After listing, a promoter typically offers a significant portion of shares to retail investors. Once this initial step is completed, those shares can then be freely traded in the secondary market or on the stock exchange.

In India, the two primary stock exchanges are the Bombay Stock Exchange (BSE), which lists around 5000 companies, and the National Stock Exchange (NSE), with about 1600 companies listed. Both the BSE and NSE operate in similar ways, following comparable trading procedures. Investors usually trade through Demat and trading accounts, which streamline the buying and selling process.

Stock exchanges play a crucial role in mobilizing public savings and directing funds into business ventures, benefiting both companies and investors. With rising inflation, low interest rates from banks, and the pursuit of better financial returns, many middle-class individuals are now turning to the equity markets. This shift highlights the growing relevance and necessity of stock exchanges in today’s investment landscape.

2.2. Open-High-Low-Close Charts

Open-high-low-close (OHLC) charts are a type of bar chart used to show the open, high, low, and close prices of stocks over consistent intervals. Each OHLC bar consists of a vertical line along with two short horizontal ticks. The vertical line represents the price range for that time span, with its top marking the highest price and the bottom marking the lowest. The horizontal ticks represent the opening and closing prices—where the tick on the left side indicates the opening price and the tick on the right side indicates the closing price for that specific period. This complete visual element is referred to as a price bar.

If the closing price (right tick) is higher than the opening price (left tick), it indicates a price rise during that period; conversely, if the closing tick is lower than the opening, it reflects a decline in price. OHLC charts are flexible and can be applied to any time frame, such as minutes, hours, days, or longer durations based on the user's needs. Though more detailed than simple line charts, OHLC charts convey the same level of information as candlestick charts—the key difference lies in how the data is visually presented. While OHLC uses horizontal ticks on either side of the vertical line to mark open and close prices, candlestick charts represent these values using the filled or hollow body of the candle.

2.3. Interpreting OHLC charts

There are several ways and methods of interpreting or analyzing OHLC charts. Fig.1 and Fig.2 shown below display the markings and denotations to understand OHLC values.

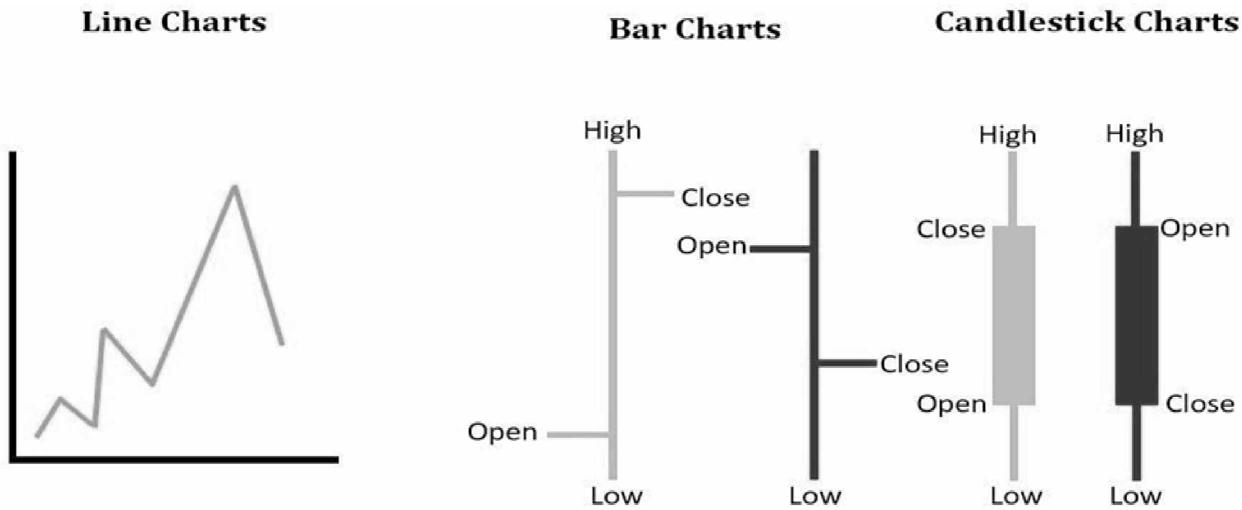


Fig.1: Types of charts showing OHLC values [11]

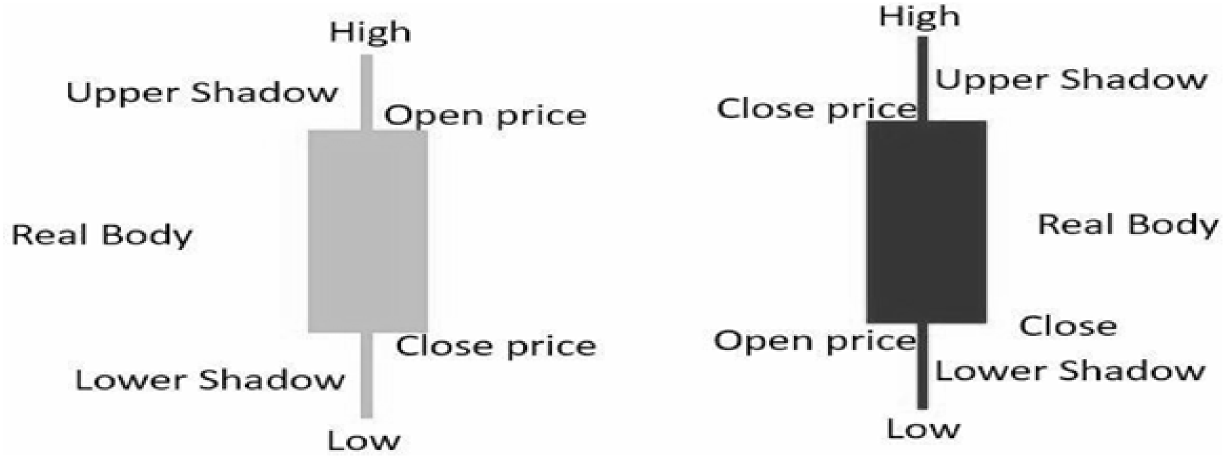


Fig. 2: Candlestick interpretation

1. Vertical height: The vertical height indicates the volatility of the stock market during a given period. The more the height of the vertical line, the more the volatility in the market.
2. Horizontal line: The leftmost and the rightmost points of the line imply the highest and lowest opening and closing values respectively. The similarity in the opening and closing values imply a condition of indecisiveness in the market.
3. Bar color: Black-colored bars imply an upward trend whereas red-colored bars imply a downtrend. Such information is handy when trying to analyze the trend strength and the direction.
4. Patterns: The major patterns are the inside bar, the outside bar, and a key reversal. Although key reversals do not appear very often, they are significant when they occur, giving reliable information to the traders regarding trend reversals of the signal whether upward accelerating or downward accelerating.

3. STOCK PREDICTION TECHNIQUES TAXONOMY

These techniques have gained popularity and have shown promising results in the field of stock analysis in the recent past.

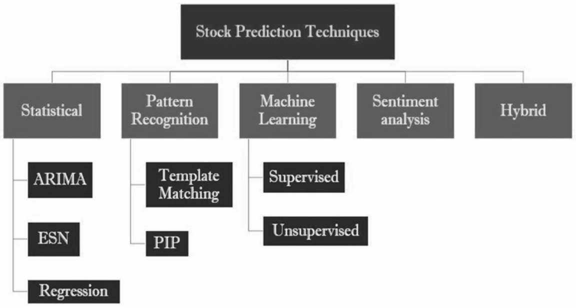


Fig. 3: Stock Prediction Techniques

When talking about recent advancements in stock market prediction, there are mainly four prominent categories: Statistical methods, Machine Learning

(ML), Pattern Recognition, and Sentiment Analysis. Besides these individual methods, there also exist techniques that combine several approaches, forming what are called hybrid models. The diagram illustrated above outlines the taxonomy of popular stock prediction approaches.

Before the widespread adoption of efficient ML models, statistical approaches were among the most recognized ways to examine and anticipate stock behavior. These methods assumed properties such as linearity, stationarity, and normality in data, which helped analysts build stock forecasting models. A commonly used term in such statistical frameworks is 'Time Series', which refers to the continuous and organized collection of data points over time—such as daily price levels, number of investors, or stock volumes. These statistical models often fall under the category of univariate analysis as they work with a single data variable at a time. Examples of such models include ARIMA (AutoRegressive Integrated Moving Average), ARMA (AutoRegressive Moving Average), STAR (Smooth Transition Autoregressive), and GARCH (Generalized Autoregressive Conditional Heteroskedasticity). ARMA, for instance, is a mix of two components: the autoregressive part, which reflects the momentum or trend observable in stock prices, and the moving average component, which explains sudden fluctuations or shocks found in time-based datasets.

However, due to the inherently volatile nature of financial time series data, the ARMA model alone may not offer optimal accuracy, as it fails to accommodate the rapidly shifting variances within market conditions. In contrast, the ARIMA model goes a step further by incorporating differencing methods that convert non-stationary series into stationary ones, thus making them better suited for predictive modeling. By applying ARIMA to historical stock data, analysts can generate forecasts for upcoming values. Additionally, statistical prediction using multiple variables includes methods such as LDA (Linear Discriminant Analysis), QDA (Quadratic Discriminant Analysis), and other regression-based algorithms, each of which considers several input features to offer more refined predictions. Pattern Recognition, which is another significant predictive method, shares some conceptual similarities with ML but is implemented differently in the context of market analysis. The focus in pattern recognition lies in spotting consistent behavioral trends and recurrent sequences across datasets. Within financial markets, such patterns are often interpreted through visualizations like OHLC (Open-High-Low-Close) candlestick charts, widely used by seasoned traders and investors. These charts contain recurring shapes and structures—referred to as patterns—like wedges, flags, triangles, saucers, and head-and-shoulders. These patterns help market participants in predicting short-term or long-term movements in stock prices. This method of technical analysis is deeply reliant on the nature and depth of the stock dataset in use. An essential part of it is the visual study of chart patterns generated from time series, which represent the dynamic behavior of variables like price, trading volume, and indirectly computed metrics such as momentum.

Comparing chart patterns to historical movements helps analysts make informed projections about future price behavior. Charting is widely regarded as a powerful technique because it integrates historical volume data with price changes, offering context to current market valuations. Pattern recognition can be broadly classified into two major techniques. The first is PIP (Perceptually Important Points), which aims to reduce data complexity by selecting the most critical points that define the overall trend. The second is template matching, where current stock behavior is compared to predefined graphical templates in a way similar to object recognition in computer vision systems. With growing access to data and computational power, ML has also become one of the most popular and effective tools for stock prediction. In a broader sense, ML approaches can be separated into two categories: supervised and unsupervised learning. In supervised learning, the model is trained on labeled data, meaning the input data is tagged with corresponding outcomes. This helps the model learn the relationships between inputs and outputs. Once trained, the model can predict outputs for new data based on what it learned. Unsupervised learning, however, involves datasets that are not labeled. Here, the algorithm attempts to detect hidden patterns, structures, or groupings within the data on its own. Interestingly, unsupervised learning can also serve as a foundation for supervised learning by offering insights into the structure of data that may later be used to improve model performance. This layered approach enhances prediction effectiveness when integrated into more complex hybrid systems.

4. ML ALGORITHMS APPLIED

4.1. Linear Regression Algorithm

Linear regression is a supervised learning algorithm within the realm of Machine Learning. Instead of classifying data into categories, it focuses on forecasting continuous values within a given range. It functions by creating a straight-line correlation between the independent and dependent variables. However, its performance tends to decline when applied to non-linear datasets, especially those containing outliers. Researchers applied this method to stock market forecasting and discovered that, although it can be used to estimate daily stock prices, it presents considerable challenges that require attention. Therefore, predictions made using this model are not dependable enough for investors to make secure financial decisions.

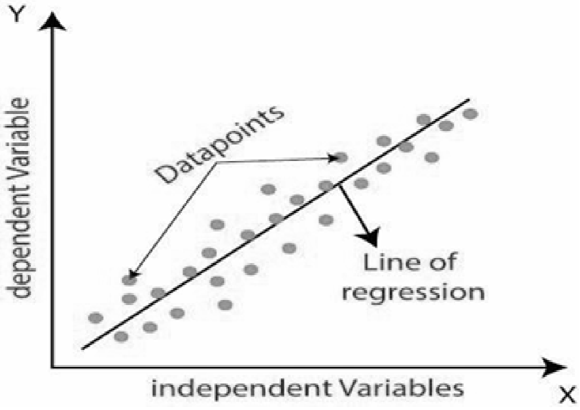


Fig. 4: Linear Regression [33]

4.2. K-Nearest Neighbours Algorithm

The K-nearest-neighbor (K-NN) algorithm is recognized as one of the most important and reliable techniques for separating data and is often preferred for use when the input data contains uncertainties. Introduced in 1951 by Evelyn Fix and Joseph Hodges, K-NN belongs to the supervised learning category. Though it can handle both classification and regression tasks, it is mainly employed for classification. Often called a lazy learner, this algorithm does not process the training data until it is needed for classifying or predicting new input. Initially, the dataset is simply stored. Moreover, K-NN is considered non-parametric, meaning that it does not rely on any predefined assumptions regarding the functional relationship between inputs and outputs. This characteristic allows it to flexibly adapt to different data structures during analysis.

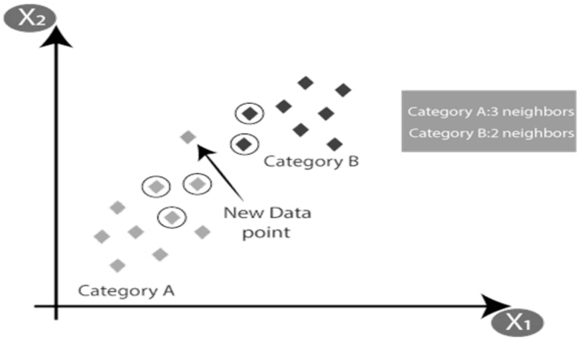


Fig.5: K-Nearest Neighbour Algorithm [36]

4.3 Support Vector Regression

The Support Vector Regression (SVR) algorithm is a widely used supervised learning technique designed to address both regression and classification problems. In SVR, a hyperplane is determined with the largest margin to ensure that the greatest number of data points lie within these margins. The hyperplane in this case represents the best-fit line that encompasses the highest number of points. The algorithm selects extreme data points, known as Support Vectors, which play a crucial role in forming the optimal hyperplane. The working principle of SVR is quite similar to that of the Support Vector Machine (SVM) algorithm. SVR is particularly effective when dealing with time series data and is often applied in forecasting tasks.

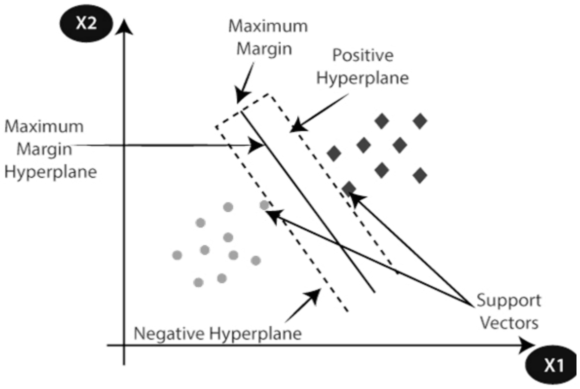


Fig.6: Support Vector Regression Algorithm [39]

4.4. Decision Tree Regression Algorithm

The Decision Tree Regression algorithm, a part of the supervised learning category, is commonly used for classification tasks but can also be applied in regression scenarios. The algorithm consists of internal nodes, representing branch structures, and leaf nodes, which represent the output or result of the algorithm. There are two key types of nodes: the decision node, which helps in making decisions and branches out into various paths, and the leaf node, which indicates the result of the decision and does not branch further. The root node serves as the starting point, expanding into a tree-like structure. Essentially, a decision tree divides data into sub-trees based on binary questions, such as a "Yes" or "No," making it easy to follow and interpret.

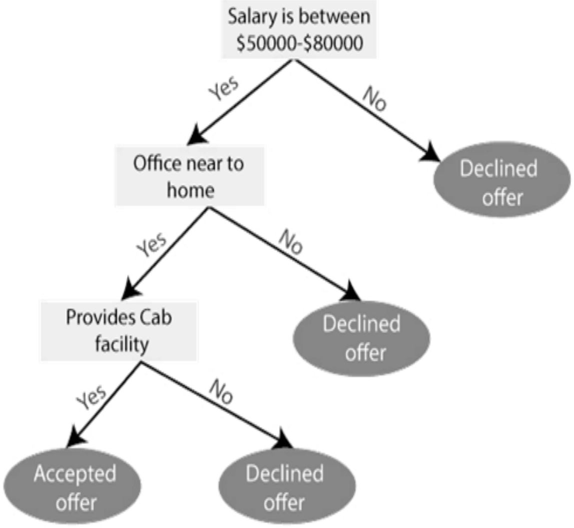


Fig. 7: Decision Tree Regression Algorithm [41]

4.5. Long Short-Term Memory Algorithm

Backpropagation with real-time recurrent learning or time can cause error signals, which travel backward in time, to either vanish or amplify significantly. The extent of temporal shifts in these error-incorporated signals largely depends on the size of the weights. In the case of signal blow-up, the weights tend to oscillate, whereas in the case of disappearance, the time required to bridge longer time lags either exceeds practical limits or, in extreme cases, the method fails to work. To address these issues, the Long Short-Term Memory (LSTM) algorithm was introduced in 1991 by Sepp Hochreiter and Jurgen Schmidhuber as a new type of recurrent neural network designed to overcome the challenges of error backpropagation. The initial version of the LSTM algorithm included only cells, input gates, and output gates. LSTM networks are effective at bridging long time breaks, even when input sequences are noisy or hard to compress, while ensuring that short-term dependencies are preserved, which previous systems struggled to achieve.

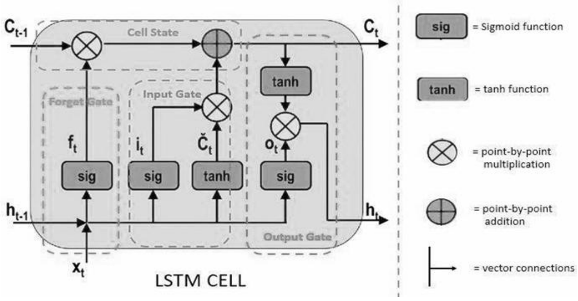


Fig.8: Long Short-Term Memory Algorithm

$$\begin{aligned} f_t &= \sigma_g(W_f x_t + U_f h_{t-1} + b_f) \\ i_t &= \sigma_g(W_i x_t + U_i h_{t-1} + b_i) \\ o_t &= \sigma_g(W_o x_t + U_o h_{t-1} + b_o) \\ c_t &= f_t \circ c_{t-1} + i_t \circ \sigma_c(W_c x_t + U_c h_{t-1} + b_c) \\ h_t &= o_t \circ \sigma_h(c_t) \end{aligned}$$

5. METHODOLOGY

The methodology constitutes the foundational pillar of any academic investigation, ensuring that the insights derived are both credible and precise. Accordingly, the methodology employed in this research has been intricately formulated and methodically structured to yield consistent and exhaustive inferences based on real-world implementation. The structured approach adopted for this study is presented through the following phases. The first phase involves the procurement of raw financial data from diverse publicly accessible databases. The second phase is centered on data preprocessing, encompassing processes such as normalization, standard scaling, anomaly detection, and various systematic techniques aimed at refining the dataset for optimal model performance. In the third phase, the dataset is strategically split into two segments—one designated for training and the other for testing purposes.

In the fourth stage, five individually developed models, each based on a distinct computational algorithm, are trained using the allocated training data. Following this, the trained models are validated on the testing set to evaluate their forecasting capabilities, capturing the extent of deviation between predicted and observed values. The final stage involves the comprehensive performance analysis of each of the five models—implemented across datasets of twelve different companies—using three critical statistical indicators: Symmetric Mean Absolute Percentage Error (SMAPE), R-squared Value (R^2), and Root Mean Square Error (RMSE). These evaluation metrics are widely recognized in predictive analytics and are employed here to facilitate a comparative assessment of algorithmic effectiveness. The performance results enable a nuanced analysis of the strengths and limitations of the selected models, namely K-Nearest Neighbors, Linear Regression, Support Vector Regression, Decision Tree Regression, and Long Short-Term Memory networks. This structured methodology ensures a reliable framework for deriving data-driven conclusions. The overall research process has been summarized in the flowchart presented below, while subsequent sections elaborate further on each procedural component in greater detail.

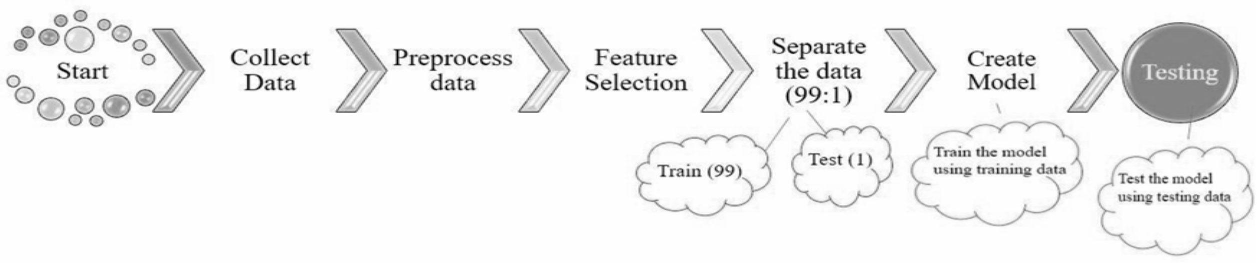
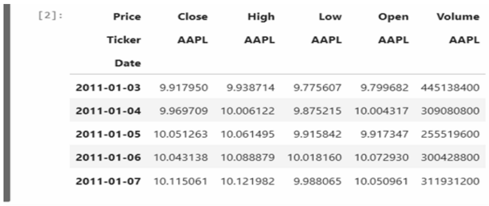


Fig.9: Methodology followed

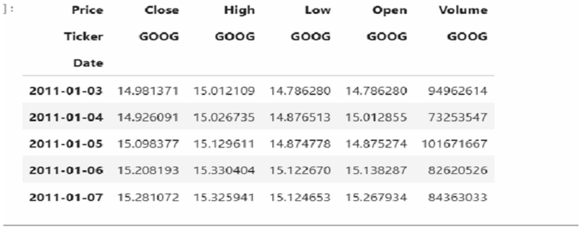
5.4. Data Description

The initial and imperative phase in most machine learning-based predictive modeling tasks involves the identification or acquisition of a suitably comprehensive dataset. In the present research, historical stock price data was procured from reputable open-access sources, specifically the Quandl database and the Bombay Stock Exchange, covering the span from January 2015 through April 2021. Each dataset encompasses critical features such as the prior day's closing value, the opening price, the intraday peak and minimum values, the most recent trade value, and the official closing price. The selected companies include industry giants such as Apple, Google, Axis Bank, Housing Development Finance Kotak Mahindra Bank. The cumulative dataset compiled from these corporations serves as the empirical bedrock for training and testing the implemented machine learning and deep learning algorithms, enabling a systematic evaluation of their predictive proficiency.



[2]:

Price	Close	High	Low	Open	Volume
Ticker	AAPL	AAPL	AAPL	AAPL	AAPL
Date					
2011-01-03	9.917950	9.938714	9.775607	9.799682	445138400
2011-01-04	9.969709	10.006122	9.875215	10.004317	309080800
2011-01-05	10.051263	10.061495	9.915842	9.917347	255519600
2011-01-06	10.043138	10.088879	10.018160	10.072930	300428800
2011-01-07	10.115061	10.121982	9.988065	10.050961	311931200



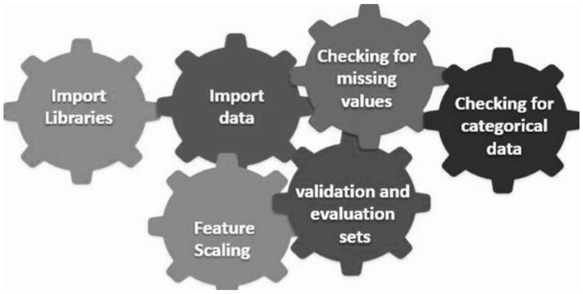
]:

Price	Close	High	Low	Open	Volume
Ticker	GOOG	GOOG	GOOG	GOOG	GOOG
Date					
2011-01-03	14.981371	15.012109	14.786280	14.786280	94962614
2011-01-04	14.926091	15.026735	14.876513	15.012855	73253547
2011-01-05	15.098377	15.129611	14.874778	14.875274	101671667
2011-01-06	15.208193	15.330404	15.122670	15.138287	82620526
2011-01-07	15.281072	15.325941	15.124653	15.267934	84363033

Fig.10 – Samples of stock datasets collected for twelve companies along with all the attributes

5.5. Data Pre-processing

Data preprocessing constitutes a vital phase in data-centric analytical endeavors, as it systematically converts raw, disordered information into a coherent and structured format, thereby enhancing its overall integrity and utility. This transformative procedure encompasses the elimination of extraneous or erroneous entries, the standardization of variables, and the meticulous preparation of data to facilitate the extraction of actionable insights. The success of a predictive model is contingent not upon the mere quantity of data available but rather on its veracity and quality. Core preprocessing operations include cleansing inconsistent values, segmenting datasets, applying scaling techniques, normalizing distributions, standardizing variables, and encoding non-numeric attributes. In this study, Min-Max scaling techniques were utilized to normalize and standardize the dataset, ensuring its suitability for subsequent algorithmic analysis. Null, missing, and ambiguous data points were detected and addressed methodically to avoid any analytical discrepancies. Prominent Python libraries such as NumPy and Pandas were deployed for data structuring and transformation, while Matplotlib was harnessed for visual representation. NumPy proved instrumental in executing computational tasks involving scientific operations, whereas Pandas facilitated robust data management and organization. Matplotlib, conversely, enabled the lucid depiction of statistical patterns through illustrative graphs and plots, enriching the interpretability of the dataset.


Import Libraries
Import data
Checking for missing values
Checking for categorical data
Feature Scaling
validation and evaluation sets

4.8 Splitting of data into train and test dataset

As previously discussed, advancing beyond the data preprocessing phase necessitates partitioning the dataset into distinct training and testing components. In this investigation, the dataset was segmented using a 99:1 ratio. This specific division was deliberately chosen due to the inherently dynamic and volatile nature of the dataset, which necessitates a larger training portion for dependable predictive modeling. Of the cumulative 2312 observations—representing trading days—only the most recent 8 entries were designated for testing purposes, while the remaining data was utilized for model training. Given the temporal structure of the data, which features interlinked sequences, a comprehensive portion of historical records is vital for effective time series modeling. A subset of the dataset was further earmarked for cross-validation to ensure robustness. To introduce randomness and mitigate potential sampling bias, randomly selected entries were allocated across both training and testing subsets, thereby improving model generalization during evaluation. The Scikit-learn library was employed to facilitate the partitioning process. The chosen split ratio holds strategic importance, as it bears a direct influence on model fidelity. Ensuring this balance allows for a more realistic appraisal of predictive capabilities across various algorithms and guarantees that the conclusions drawn are both statistically valid and empirically reliable.

4.9. Training of models

Model training represents a pivotal phase in machine learning (ML) workflows, as it empowers algorithms to discern latent structures and derive informative patterns, thereby enabling the generation of reliable predictions aligned with intended outcomes. In this study, five distinct algorithms—K-Nearest Neighbour (K-NN), Support Vector Regression (SVR), Linear Regression, Decision Tree Regression, and Long Short-Term Memory (LSTM)—were deployed and trained using the designated training data associated with twelve selected companies. This methodical training regimen ensured the mitigation of potential risks related to both overfitting and underfitting. Emphasis was placed on facilitating incremental improvements in the models' predictive performance through iterative learning. All algorithms utilized fall within the supervised learning paradigm, and the implementation strategy accordingly adhered to supervised learning principles. The training phase was inherently iterative and involved a process known as "model fitting," during which the algorithm adaptively learned from the data. Initially, model parameters were initialized with randomized values, allowing the learning algorithms the flexibility to optimally converge toward minimized error. This methodological flexibility facilitated the refinement of model weights across iterations, ensuring improved generalization to unseen data and robustness in the predictive outputs.

4.10. Testing the models

Model training stands as a critical phase in machine learning (ML) workflows, enabling algorithms to uncover hidden patterns and derive informative structures, thereby facilitating reliable predictions aligned with desired outcomes. This study implemented five distinct algorithms—K-Nearest Neighbour (K-NN), Support Vector Regression (SVR), Linear Regression, Decision Tree Regression, and Long Short-Term Memory (LSTM)—which were trained using specific training data from twelve prominent companies. This structured training process mitigated risks related to overfitting and underfitting, focusing on enhancing the models' predictive performance through iterative learning. All algorithms are based on the supervised learning framework, and the implementation adhered to supervised learning methodologies. The target variable, a key dependent attribute within the dataset, serves as the core element driving the learning process.

Key considerations, including the computational time required for training each algorithm and the time lag error, defined as the necessary temporal offset for shifting input data sequences backward, were thoroughly assessed. The training phase followed an iterative "model fitting" approach, where the algorithm adaptively learned from the data. Initially, model parameters were randomized, allowing the algorithm flexibility to minimize error and optimize performance. This adaptive learning process refined model weights with each iteration, ultimately improving generalization to unseen data and enhancing predictive accuracy.

6. RESULTS

This section presents the results of the successful implementation of the project through tables and graphs. In this study, several algorithms including K-Nearest Neighbour, Support Vector Regression, Linear Regression, Lasso and Ridge Regression, and Long Short-Term Memory were chosen to predict stock prices for twelve distinct companies. The dataset covered a substantial period, from 2015 to 2021. The models were assessed using 8 trading days for testing, with a 99:1 training-to-testing ratio. Out of the total 2312 data points, 2304 were allocated for training, and the remaining 8 days were reserved for testing the models. The performance of these models was assessed using three pivotal metrics: Symmetric Mean Absolute Percentage Error (SMAPE), R-squared (R2), and Root Mean Square Error (RMSE). These metrics are well-established and serve as fundamental criteria for evaluating and comparing the effectiveness of the models examined in this research.

Parameter	SMAPE (Symmetric Mean Absolute Percentage Error)				
Algorithms	K-Nearest Neighbors	Linear Regression	Support Vector Regression	Lasso Regression	Ridge Regression
Adani Ports	10.99	9.18	9.51	10.68	1.65
Asian Paints	11.63	9.35	8.42	9.78	1.67
Axis Bank	16.67	10.37	5.64	8.48	1.88
HDFC	20.46	11.00	6.22	12.56	2.19
Hindustan Unilever	10.95	9.62	4.88	8.82	1.38
ICICI Bank	14.45	8.92	7.02	7.37	2.31
Kotak Bank	12.44	10.51	5.81	10.26	1.43
Maruti	15.92	11.13	3.09	13.92	1.32
NTPC	13.59	12.39	3.08	9.60	1.13
Tata Steel	16.08	13.10	5.75	8.06	1.75
TCS	15.84	9.95	4.41	10.05	1.40
Titan	12.90	3.50	3.33	11.39	1.06
Average	14.32	9.91	5.59	10.08	1.59

Table I displays the tabulated results for the SMAPE acquired when a particular model was tested for that particular company's dataset. With the ideal value for SMAPE being close to zero, from this table, it is observed that out of all the five different algorithmic models, the DL algorithm i.e., Lasso algorithm has rendered the best predictive performance, as it has the least value of error (1.59), followed by Support Vector Regression, with a SMAPE of 5.59

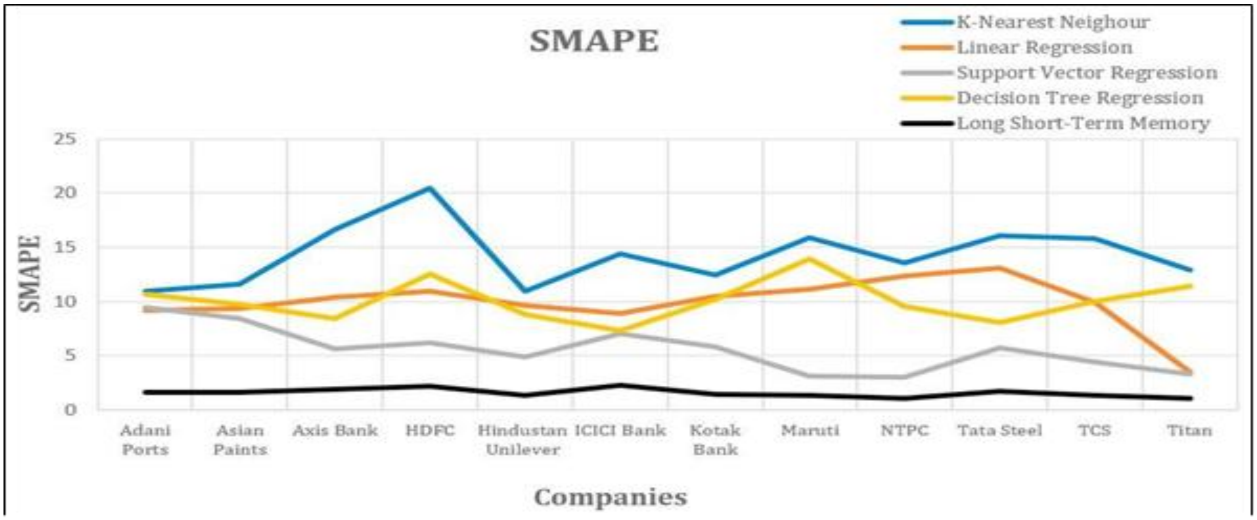


Fig. 12: The SMAPE for all algorithms plotted against the companies

The plot displayed in Fig. 12 illustrates the predictive performance of all five algorithms, with SMAPE values plotted for each of the twelve companies. From the plot, it is evident that the LSTM algorithm achieved the best performance, as its corresponding black plot (LSTM) lies significantly lower than the others. Additionally, it can be concluded that the SVR algorithm provided the second-best results. However, the remaining models do not appear to be suitable choices for predictive analytics, as they show a high level of error and are less reliable for accurate predictions.

Table II: Tabulated results showing R-squared value for all models and companies

Parameter	R^2 (R squared				
Algorithms	K-Nearest Neighbors	Linear Regression	Support Vector Regression	Lasso Regression	Ridge Regression
Adani Ports	-0.22	-6.67	-1.76	-3.25	-0.90
Asian Paints	-1.66	-5.57	-2.21	-2.75	-0.45
Axis Bank	-5.32	-3.43	-1.05	-4.18	0.59
HDFC	-1.56	-2.47	-2.77	-3.04	-0.62
Hindustan Unilever	-2.03	-1.07	-0.35	-2.39	0.27
ICICI Bank	-2.59	-1.59	-2.72	-2.23	0.45
Kotak Bank	-3.78	-2.31	-3.38	-3.90	-0.01
Maruti	-1.01	-0.65	-2.61	-0.73	-0.99
NTPC	-6.90	-2.31	-1.11	-0.84	-0.02
Tata Steel	-2.16	0.48	-1.13	-2.01	0.80
TCS	-0.83	-1.18	-0.51	-0.92	-0.84
Titan	-0.21	-0.57	-0.12	-1.72	0.31
Average	-2.42	-2.27	-1.69	-2.33	-0.11

Table II presents the results for the R-Squared value (R2) obtained when testing each model on the dataset of the respective company. Since the ideal R-Squared value is as close to a non-negative ‘1’ as possible, it can be observed from the table that, out of all five models, the Deep Learning (DL) model, i.e., the Long Short-Term Memory (LSTM) algorithm, yielded the best results, with an R-Squared value near 1 (specifically, -0.11). This is followed by the Support Vector Regression (SVR) algorithm, which has an R-Squared value of -1.69, and the remaining models show less favorable results.

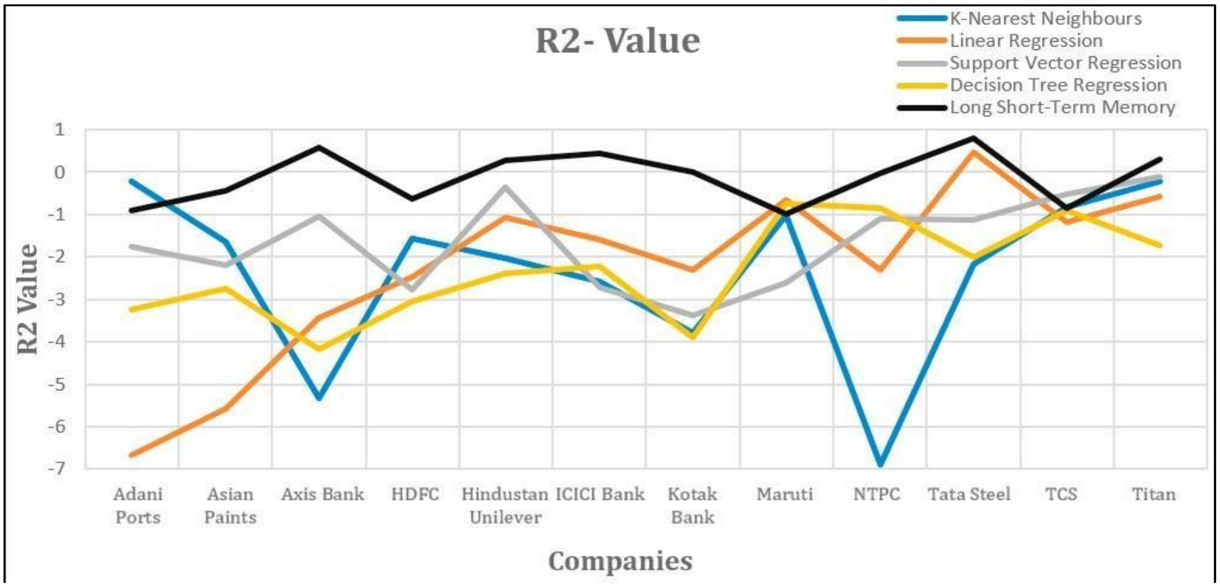


Fig. 13: The R-squared value for all algorithms plotted against the companies

Figure 13 illustrates the predictive performance of all five algorithms in terms of R-squared values plotted across the twelve companies. A careful examination of the graph reveals that the LSTM algorithm produces the best results, as the topmost black plot line (LSTM) is the closest to ‘1’ when compared to the other algorithms. Furthermore, it can be concluded that the SVR algorithm offers the second-best performance, while the other models do not demonstrate significant or reliable prediction accuracy.

Table III: Tabulated results showing RMSE value for all models and companies

Parameter	RMSE (Root Mean Square Error)				
Algorithms	K-Nearest Neighbors	Linear Regression	upport Vector Regression	Lasso Regression	Ridge Regression
Adani Ports	29.78	43.76	37.94	43.25	16.22
Asian Paints	51.78	37.68	80.44	40.12	14.31
Axis Bank	47.41	60.03	66.23	48.65	15.77
HDFC	66.16	63.37	54.02	58.99	35.71
Hindustan Unilever	40.01	45.11	36.89	62.11	40.05
ICICI Bank	50.20	49.34	38.90	49.70	16.09
Kotak Bank	49.91	50.01	41.55	52.91	34.82
Maruti	84.72	73.64	21.70	63.22	12.94
NTPC	63.65	25.19	15.28	41.61	10.60
Tata Steel	70.17	50.26	54.18	71.07	22.80
TCS	67.13	55.68	58.74	44.14	30.56
Titan	56.36	60.39	50.49	24.46	20.83
Average	56.44	51.20	46.36	50.01	22.55

Table III presents the results for the Root Mean Square Error (RMSE) obtained when testing each model with a specific company's dataset. Since the ideal RMSE value is zero, it is evident from this table that the Long Short-Term Memory (LSTM) algorithm, a deep learning model, achieved the best performance with an RMSE of 22.55. This was followed by Support Vector Regression, which produced an RMSE of 46.36, and so on for the remaining models.

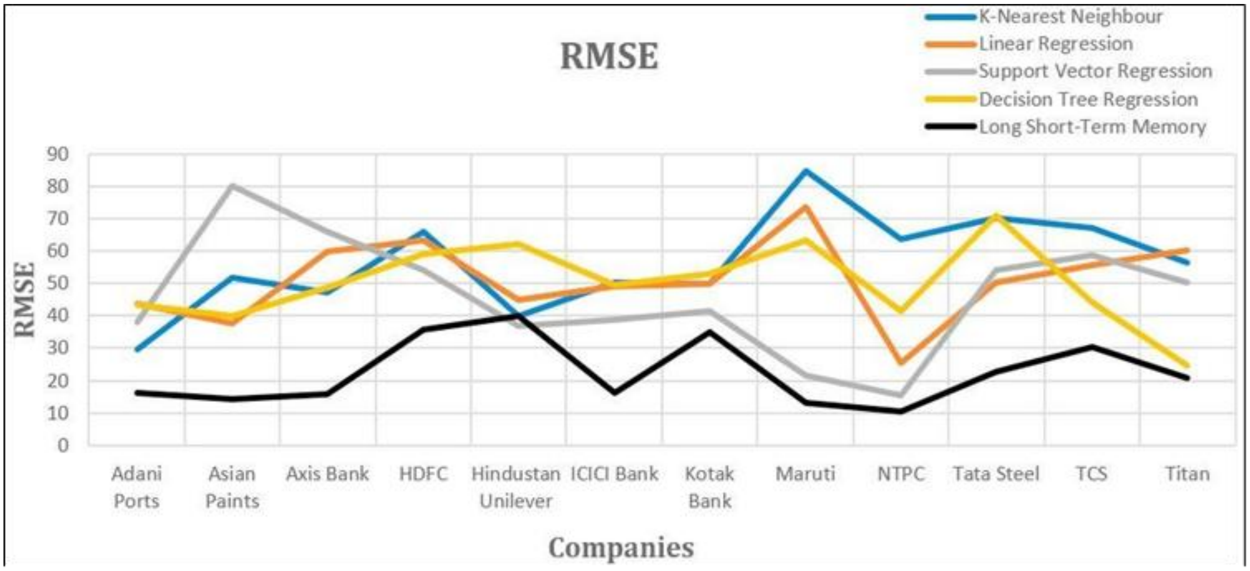


Fig.14: The RMSE for all algorithms plotted against the companies

The plot in Fig.14 illustrates the predictive performance of all five algorithms in terms of RMSE, plotted across twelve companies. Upon close examination of the chart, it is apparent that the LSTM algorithm has provided the best results, as its lowest black plot line (LSTM) is the closest to ‘0’, outperforming the other models. The SVR algorithm appears to have the second-best performance, with the remaining models trailing in terms of predictive accuracy and efficiency. After carefully reviewing all the plotted and tabulated results, it is clear that the Deep Learning (DL) model, LSTM, stands out as the top choice for predictive analytics among the selected algorithms, with the Support Vector Regression algorithm following closely behind, also yielding favorable results. Linear Regression and Decision Tree Regression produced almost identical outcomes, whereas K-NN exhibited the least favorable performance, primarily due to being a classification algorithm, not suited for prediction tasks.

7. FUTURE SCOPE

With the increasing demand for machine learning (ML) across industries, business models, and healthcare, the focus is on developing more sophisticated models capable of delivering accurate and reliable predictions from vast amounts of data. However, research and analysis indicate that ML often struggles to produce reliable results when applied to time series forecasting. A promising solution lies in deep learning (DL) and neural networks, which consistently outperform traditional ML methods in time series predictions. The findings of this study align with the theoretical performance of these algorithms. Looking ahead, further exploration of additional algorithms and datasets could enhance the comparison between ML and DL techniques. Unlike ML, which demands considerable human involvement, DL techniques mimic the brain's structure with multiple layers, enabling more autonomous feature identification through hierarchical arrangements. A key advantage of DL is its ability to improve as more data becomes available, a vital attribute for data-intensive tasks like time series forecasting. Future advancements in both ML and DL models, characterized by minimal human intervention, faster prediction times, and greater accuracy in handling large datasets, are essential. These improvements will reduce complexity, enhance cost-effectiveness, and pave the way for faster, more accurate predictions. Future research in DL and ML holds immense potential to create more powerful systems capable of providing real-time, precise predictions with minimal human involvement, fostering advancements across various fields.

8. CONCLUSIONS

This study focused on developing machine learning (ML) models designed to predict stock prices with enhanced accuracy, empowering traders and investors to make well-informed decisions and optimize profits by investing at the opportune moment. The project successfully implemented five distinct algorithms—K-Nearest Neighbours, Linear Regression, Support Vector Regression, Decision Tree Regression, and Long Short-Term Memory (LSTM)—to build predictive models for stock price forecasting of twelve major Indian companies: Adani Ports, Asian Paints, Axis Bank, Housing Development Finance Corporation Limited (HDFC) Bank, Industrial Credit and Investment Corporation of India (ICICI) Bank, Kotak Bank, Hindustan Unilever Limited, Maruti, National Thermal Power Plant Corporation (NTPC), Tata Steel, Tata Consultancy Services (TCS), and Titan. Subsequently, a thorough comparative analysis was performed to evaluate the performance of these algorithms in stock price prediction.

The research leveraged stock price data spanning from 2015 to 2021, with findings indicating that deep learning (DL) models outperform traditional machine learning (ML) algorithms in forecasting time series data. Among the five models tested, Long Short-Term Memory (LSTM), a deep learning approach, achieved the most precise predictions. The results section of this paper evaluates the models based on three key metrics: Symmetric Mean Absolute Percentage Error (SMAPE), R-Squared Value (R^2), and Root Mean Square Error (RMSE). After a comprehensive analysis, it was concluded that LSTM provided the most effective results for time series forecasting, registering minimal error values: SMAPE (1.59), R^2 (-0.11), and RMSE (22.55). The second-highest performing model was Support Vector Regression, with SMAPE (5.59), R^2 (-1.69), and RMSE (46.36). Both Linear Regression and Decision Tree Regression showed near-identical results, while K-Nearest Neighbours performed the least well, as it is primarily designed for classification rather than regression tasks. Thus, the outcomes corroborate theoretical expectations, highlighting the advantage of deep learning in stock price prediction.

REFERENCE

[1] Pei-Yuan Zhou, Keith C.C. Chan, Member, IEEE, and Carol XiaojuanOu, “Corporate Communication Network and Stock Price Movements: Insights From Data Mining”, IEEE 2021

[2] Atsalakis GS, Valavanis KP. Forecasting stock market short-term trends using a neuro-fuzzy based methodology. Expert Syst Appl. 2009;36(7):10696–707.

[3] Ayo CK. Stock price prediction using the ARIMA model. In: 2022UKSim-AMSS 16th international conference on computer modelling and simulation. 2014.

[4] Brownlee J. Deep learning for time series forecasting: predict the future with MLPs, CNNs and LSTMs in Python. Machine Learning Mastery. 2021. https://machinelearningmastery.com/time- series-prediction-lstm-recurrent-neural networkspython-keras.

[5] Shih D. A study of early warning system in volume burst risk assessment of stock with Big Data platform. In: 2019 IEEE 4th international conference on cloud computing and big data analysis (ICCCBDA). 2019. pp. 244–8.

[6] Sirignano J, Cont R. Universal features of price formation in fnancial markets: perspectives from deep learning. Ssrn. 2018. https://doi.org/10.2139/ssrn.3141294.

[7] Thakur M, Kumar D. A hybrid fnancial trading support system using multi-category classifers and random forest. Appl Soft Comput J. 2018;67:337–49. https://doi.org/10.1016/j.asoc.2018.03.006.

[8] Tsai CF, Hsiao YC. Combining multiple feature selection methods for stock prediction: union, intersection, and multiintersection approaches. Decis Support Syst. 2020;50(1):258–69. https://doi.org/10.1016/j.dss.2020.08.028.

[9] Tushare API. 2018. https://github.com/waditu/tushare. Accessed 1 July 2020.

[10] Wang X, Lin W. Stock market prediction using neural networks: does trading volume help in short-term prediction?. n.d.

[11] Weng B, Lu L, Wang X, Megahed FM, Martinez W. Predicting short-term stock prices using ensemble methods and online data sources. Expert Syst Appl. 2020;112:258–73. https://doi.org/10.1016/j.eswa.2018.06.016.

[12] Zhang S. Architectural complexity measures of recurrent neural networks, (NIPS). 2019. pp. 1–9.

[13] Ashish Sharma, Dinesh Bhuriya, Upendra Singh. "Survey of Stock Market Prediction Using Machine Learning Approach", ICECA 2022.

[14] Loke.K.S. “Impact Of Financial Ratios And Technical Analysis On Stock Price Prediction Using Random Forests”,

[15] Xi Zhang1, Siyu Qu1, Jieyun Huang1, Binxing Fang1, Philip Yu2, “Stock Market Prediction via Multi-Source Multiple Instance Learning.” IEEE 2021.

[16] VivekKanade, BhausahebDevikar, SayaliPhadatare, PranaliMunde, ShubhangiSonone. “Stock Market Prediction: Using Historical Data Analysis”, IJARCSSE 2022.

[17] SachinSampatPatil, Prof. Kailash Patidar, Asst. Prof. Megha Jain, “A Survey on Stock Market Prediction Using SVM”, IJCTET 2021.

[18] https://www.cs.princeton.edu/sites/default/files /uploads/Saahil_magde. Pdf

[19] Hakob GRIGORYAN, “A Stock Market Prediction Method Based on Support Vector Machines (SVM) and Independent Component Analysis (ICA)”, DSJ 2021.

[20] RautSushrut Deepak, ShindeIshaUday, Dr. D. Malathi, “Machine Learning Approach In Stock Market 9. Prediction”, IJPAM 2022.