CHAPTER VI

MODIFIED EXTREME LEARNING MACHINE ALGORITHM WITH DETERMINISTIC WEIGHT MODIFICATION FOR STOCK PRICE PREDICTION USING SENTIMENT ANALYSIS

6.1 Introduction

To create prediction models for financial markets, researchers have mined historical, social media, and news item data using various ML algorithms. In the days before social media and internet news sources gained the prominence they have now, stock market predictions had to be made by researchers using historical data. Users of social networking sites can communicate with one another directly. These platforms have grown in popularity as places where users can exchange the findings of financial analyses related to financial securities.

This leads to the massive volumes of social media data that become available, attracting academics to mine these data for profit. As a result, studies on investor views shared on social media have been done to forecast stock markets. When it comes to a particular stock, financial analysts and stock market traders tweet about their feelings. Such emotions can be found in tweets using machine learning algorithms and NLP approaches. Emotions on Twitter and financial markets have been found to correlate, and this link can be utilized to predict certain aspects of a stock market.

Investors can now obtain up-to-date and very valuable information, including public sentiments that impact investment decisions, thanks to social media platforms. An association rule mining technique was proposed by researchers to identify trends in the relationship between changes in stock prices and public mood. Online news items that are accessible to the general public offer an intriguing kind of data that may be mined and analyzed to obtain relevant information. The news can be divided into three categories: political, financial, and general news. It is also possible to forecast the stock market using this kind of data. Similar to social media, political events affect stock price gains. As a result, scholars have also focused on the examination of political news for the prediction of stock.

6.2 Sentiment analysis

By examining and evaluating the feelings, thoughts, and attitudes stated in news stories, social media posts, financial reports, and other textual data about stocks and the market, sentiment analysis contributes significantly to stock market prediction. As behavioral finance has matured, there has been an increased focus on the impact of irrational factors among investors on the stock market. Stock movements, for instance, will be impacted by personality traits in the investing process and the aggravation that results from making poor decisions.

There's growing indication that depositors are not irrational, and investor associates are becoming more regular and easier as social networks become more and more essential to people. As a result, the opinions shared on social media and by other investors may have an impact on an investor's disposition, alter their course of action, and have some additional influence on the stock market. Few works have been provided that use sentiment analysis to anticipate a particular stock price; nonetheless, there is various research that shows a correlation between the movement of stocks and the emotional nature of online discourse. Because sentiment analysis aids in understanding market sentiment, which affects stock movements, sentiment analysis is important for forecasting stock prices.

Supply and demand, business performance, economic indicators, and geopolitical developments are just a few of the variables that affect stock values, which are a reflection of historical market behavior. Investors might, however, be blind to the psychological and emotional forces influencing market fluctuations without sentiment analysis. Before changes in investor sentiment are reflected in stock prices, sentiment analysis can assist detect them. Investors may lose out on opportunities or run the danger of taking on more risk if they ignore important signs that point to shifts in market sentiment.

The complete range of information included in news stories, social media conversations, company announcements, and other sources that can affect investor mood may not be fully captured by price data alone. Important information from these sources is lost when sentiment analysis is neglected. There are various ways that sentiment analysis aids in stock market forecasting.

- **Sentiment analysis:** Sentiment analysis can expose universal market sentiment, whether it is optimistic (positive) or bearish (negative), by investigating the attitudes of market players. This can support investors in determining the current state of the market and adjusting their policy consequently.

- **Media analysis:** Press statements, news training, and media reporting about particular stocks at large can all be studied using sentiment analysis. The emotion expressed in these sources, whether favorable/unfavorable, can affect stock prices and investor insights.

- **Social Media analysis:** Real-time opinions about stocks may be found in profusion on social media sites such as Reddit, Twitter, and StockTwits. Sentiment analysis of stakes on social media can be used to spot developing themes, swings, and trends in opinion that could disturb stock prices.

- **Earnings call analysis:** Sentiment analysis is earnings calls and other business communications that can be used to regulate investor responses to a company's sentiment of management. The sentiment about a company's earnings call can deliver an understanding of possible future variations. Investigative reactions to specific events, like as unions and acquisitions, product presentations, governmental changes, and geopolitical developments, can be skilled with the help of sentiment analysis. To predict how these occurrences may impact stock prices, it can be accommodating to evaluate the atmosphere surrounding them.

- **Trading strategies:** Sentiment analysis is a tool for dealers and investors can use to create transaction strategies based on sentiment. Contrarian depositors would search for chances to buy during periods of extreme negativity in the hopes of seeing probable growth.

- **Risk management:** An investor sentiment and sentiment-driven marketplace unpredictability can both offer hazards that can be recognized. Better risk management and portfolio divergence can be facilitated by an understanding of emotion.

- **Algorithmic business:** To industrialize trading choices based on sentiment signals, sentiment analysis is being included in algorithmic trading models more and more. Trades can be carried out by these approaches according to preset sentiment thresholds or sentiment-driven patterns.

- **Predictive analytics:** ML can be taught to predict future stock movements based on sentiment trends and patterns by evaluating past sentiment data combined with stock price movements.

6.3 CNN

CNN is a kind of DL method and it functions similarly to biological neurons and is constructed from neurons. The architecture of CNN is shown in Figure 25. After receiving some input, each neuron does a dot product before producing an output. It is composed of multiple convolutional layers, followed by a multilayer neural network. A CNN's basic form is a 2D network, which is primarily utilized for images. The pooling, dense, dropout, and convolutional layers are the layers that makeup CNN. One may also utilize 1D CNN for data forecasting. Like Sigmoid, ReLU, Tanh, and other activation functions, it too has one. CNN does not know the precise feature mapping when it receives new input.

To determine the proper feature mapping, it so builds a convolutional layer and convolves it. CNN's pooling layer can reduce the huge inputs. ReLU is the CNN activation function that is most frequently utilized. Its operation is straightforward: whenever a negative integer appears, it is substituted with 0. CNN also contains hidden layers. These layers are used in the error minimization process.

An artificial neural network with specific capabilities, a CNN is mostly used to analyze structured grid data, such as photographs. CNNs have proven remarkably effective in a wide range of applications, particularly computer vision tasks. The technique of classifying user sentiment in stock market prediction using CNNs entails analyzing textual data, such as financial reports, social media posts, and news stories, to identify the sentiment represented in these texts. After that, this sentiment might be utilized as a feature to forecast changes in the stock market. [72].

The CNN is divided into three portions such as input, convolution, and classification layer [73]. An $r \times u$ is the text word vector, where r - number of distinctive phrases and u - result of data processing. The convolutional layer first runs over the convolution kernel w of length h to convolve the word vector matrix. This is precisely:

$$t_i = f(w * s_{i,i+h-1} + b) \tag{29}$$

$s_{i,i+h-1}$ is a continuous text section made up of phrases i^{th}through $i + 1$ phrase. $*$ is the convolution operation. f- nonlinear function. b - bias term Maximum value pooling is used to reduce dimensionality and keep the number of features constant.

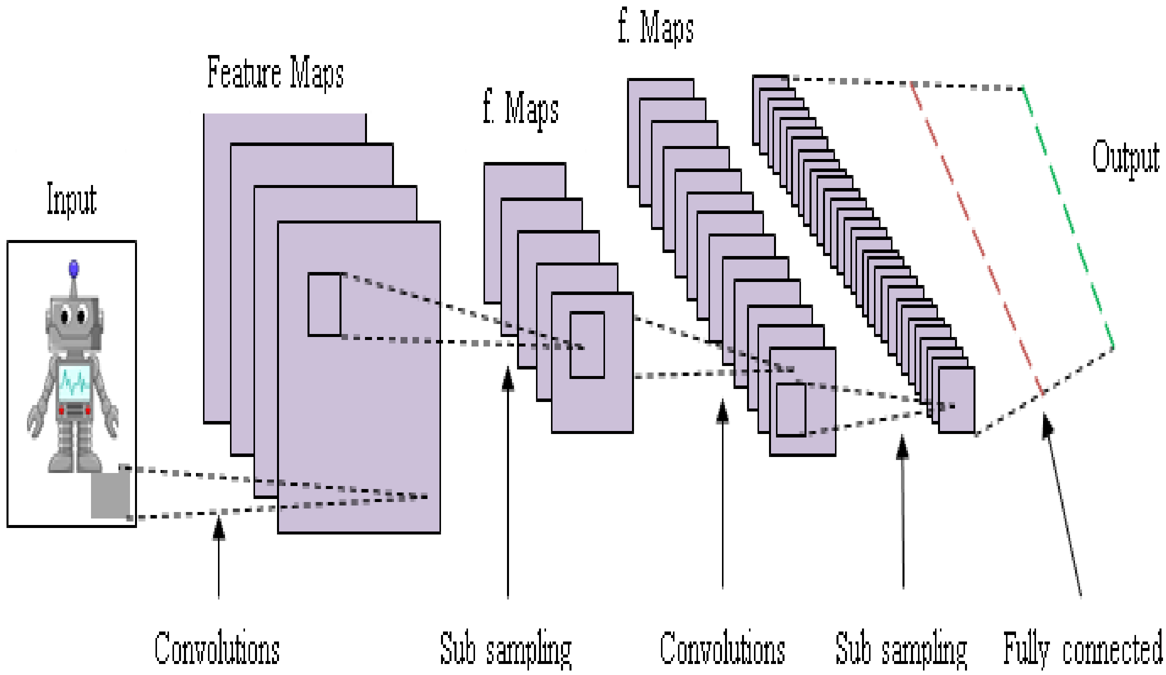


Figure 25 : Architecture of CNN

The softmax layer is used to calculate the classification probability, and the BN method is used by the classification layer to prevent changes in the data distribution. The probability is calculated as follows and is used to classify comments:

$$P_j = P(y = j \mid X, b) = \frac{e^{X^T W_j + b_j}}{\sum_{i=1}^{L} e^{X^T W_j + b_j}} \tag{30}$$

P_j- denotes the probability of j^{th}class text. X - is the input of the classification layers. W - is the weights matrix. b_i and $b_j$is the i^{th} offset element and j^{th} bias term. L - is the number of classes.

6.4 Word2vec:

A common method for natural language processing (NLP) tasks is called Word2Vec. It is especially useful for creating word embeddings, which are low-dimensional, dense vector illustrations of words in a constant vector space. These embeddings help approaches understand and process natural language quantifiable more efficiently by capturing the syntactical and semantic connections between words.

In 2013, Tomas Mikolov and other Google researchers developed Word2Vec. It is established on the concept that, within a corpus of text, words with similar imports commonly occur in comparable settings. The important idea of original Word2Vec is that a word's meaning can be assumed from the circumstances in which it appears. Word2Vec trains neural networks on vast text corpora to learn distributed representations of words.

- **Continuous bag of words (CBOW):** The target word in the CBOW architecture is predicted by the model using the context words that surround it. A window of context words surrounding the target word serves as the model's input. By running the context words through a hidden layer and producing the probability distribution throughout the vocabulary, the model gains the capacity to predict the target word. The word embedding is then derived from the hidden layer representations.
- **Skip-gram architecture**: Given the target word in the Skip-gram architecture, the model forecasts the surrounding context words. One target word serves as the model's input. By passing the target word through a hidden layer and producing the probability distribution over the lexicon, the model gains the ability to predict the surrounding context terms. The word embedding is derived from the representations of the hidden layer.

6.5 Enhanced ELM with sentiment analysis

The suggested SDELM method's flowchart is displayed in Figure 2, which details the two contributions of the proposed research project. First, a classification model based on CNN is used to calculate the SI. Second, DWM is used for optimizing ELM to forecast the stock price. One of the first objectives is to run a group that uses user sentiment evaluations for previous stock data as one factor in projecting the company's price. The SI is generated based on the quantity of daily bullish and bearish posts made by several individuals. The present research work suggests incorporating investor sentiment into stock prediction, which can significantly raise the accuracy of the algorithm. In particular, we use sentiment analysis on a vast number of stock market comments—which are reviews, including acronym modifications, spelling corrections, root renovation, and symbol substitutes, to increase the accuracy of sentiment classification.

The categorization results are used to construct the daily sentiment index, which is subsequently supplied as an input feature into the stock prediction module. Unlike previous sentiment analysis studies, we enhance classification performance by combining CNN with word2-vec. Additionally, the index's calculation allows for the emotional analysis of social groups, allowing for the evaluation of the feelings of as many investors as feasible.

As a result, we must first identify the appropriate sentiment classification of each stock review to compute the SI and ascertain the group sentiment tendency. Word2vec models can learn semantic information from large amounts of text in an unsupervised manner. For word2vec to function, words have to be mapped from the old space to the new space. Specifically, the SI is formed by learning the text, which creates the word vector to characterize the semantic information and assigns semantically associated words to similar distances.

Algorithm 4: CNN algorithm

Step 1: Define hyperparameters

Step 2: Define CNN architecture including convolution, pooling, and fully connected layers

Step 3: Initialize weights and biases

Forward pass

Step 4: For each epoch

Step 6: Calculated convolution layer output

Step 7: Calculated pooled layer output

Step 7: Calculated fully connected layer output

Step 9: Compute loss

Backward pass

Step 10: Updated weights and bias

Step 10: check termination status if yes, terminate the process. Otherwise, repeat the process.

Algorithm 5: S-DELM algorithm

Input: Stock market data and user comments

Output: Stock closing price

Step 1: Initializing the required parameters for ELM, DWM, and CNN algorithms

Step 2: Collecting the stock market data and user comments

Step 3: stock market data converted into technical indicators

Step 4: user comments are classified as bullish or bearish and calculated the sentiment score using CNN

Step 5: the technical indicators and sentiment scores are integrated as input samples

Step 6: Split the integrated input samples into training and testing phases

Step 7: Train the ELM with DWM

Step 8: Compare the performance

To learn high-dimensional vector illustrations of phrases from extensive stock comments, word2vec is first developed. If the phrase in the stock commentaries has to be considered, the results are implemented right away. If not, random is initialized by word2vec. The word vectors denote the pre-processed text, and will then be provided to CNN. Next, we create the SI using a CNN that has been improved using word2vec.

The SI for the day is calculated by adding up all of the comments that are made, whether good or negative. This can be shown in the group sentiment analysis, which may reveal the investor's overall emotional propensity. To calculate the stock's daily SI, we use the method recommended by Antweiler and Frank [74].

$$BI_t = In\frac{1+M_t^{bullish}}{1+M_t^{bearish}} \tag{31}$$

where BI_t is the SI at time $t$and $M_t^{bullish}$ and $1 + M_t^{bearish}$, the weights of bullish and bearish stocks are based on the number of bullish and bearish, respectively. The index takes into account the impact that bullish and bearish comments have had on investor sentiment over time. Deviations from the index's mean are directly proportional to the weights assigned to bullish and bearish views on stock valuation.

More bullish comments indicate a bullish general trend, as indicated by the positive index BI_t. However, if more comments express negative sentiment than positive sentiment, the mood is generally pessimistic and the SI BI_t is negative. The positive or negative tendency toward a given category is indicated by the magnitude of the SI. The group of sentimental orientation is reflected in BI_t. In the end, the S-DELM prediction model uses the SI's historical prices as input to forecast stock values in the market. The overall framework of the research work is shown in Figure 26.

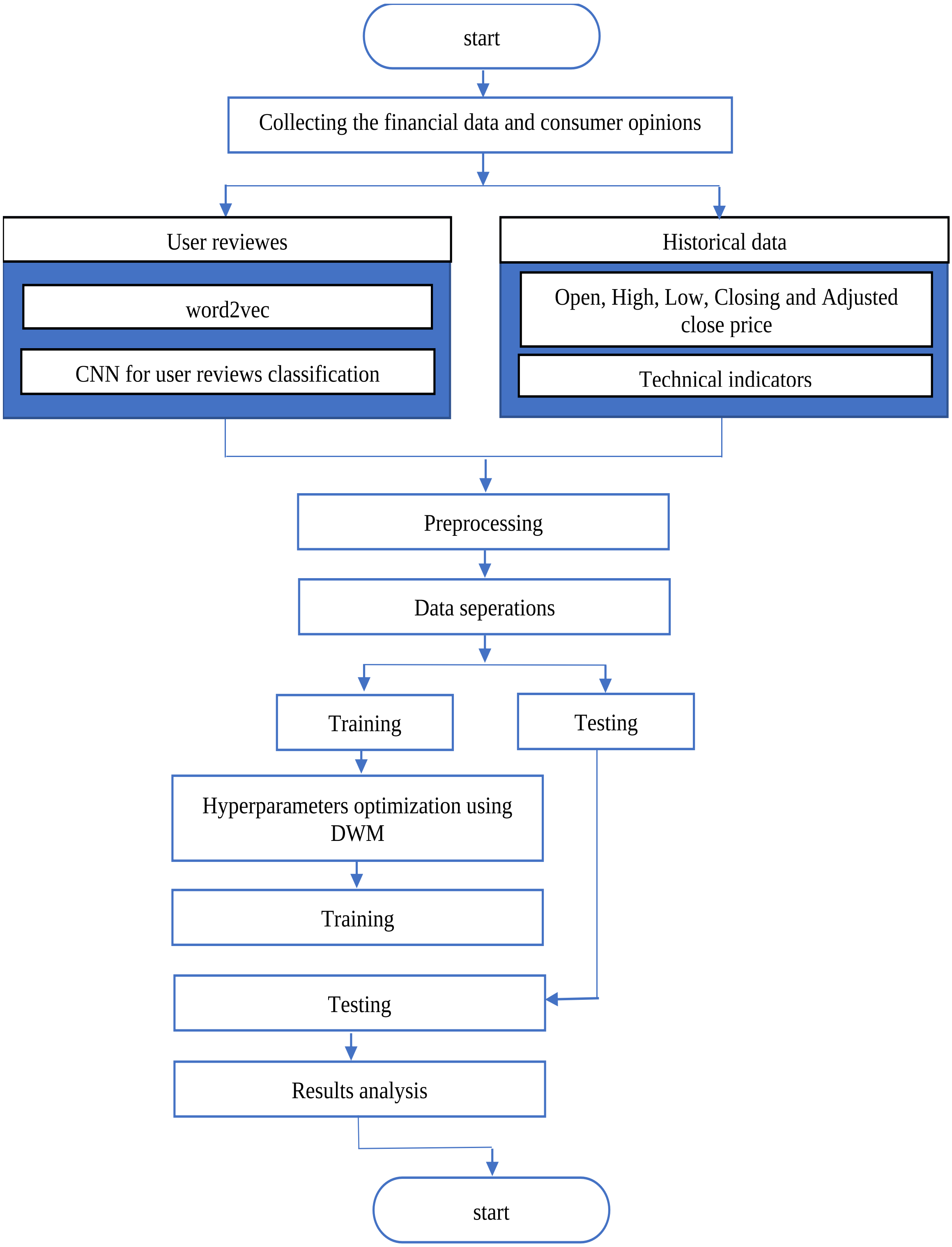
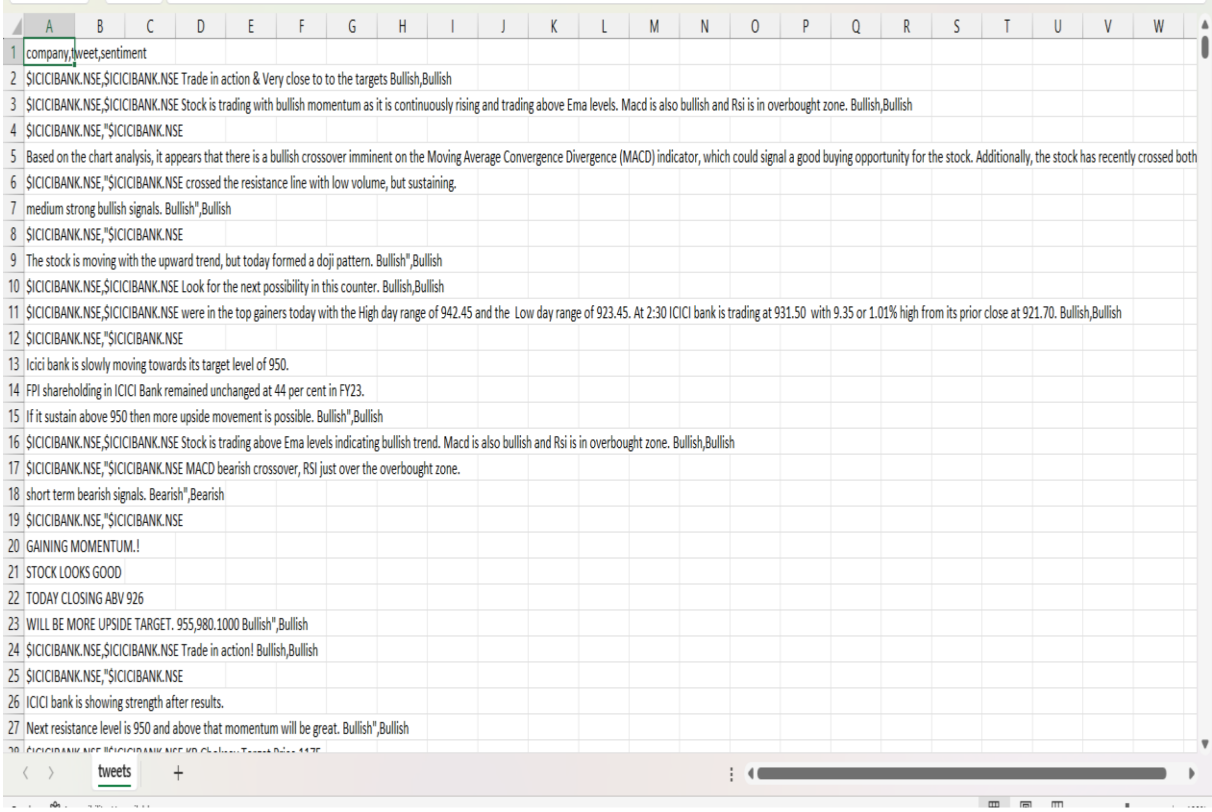


Figure 26 : Overview of the proposed framework



	A
1	company,tweet,sentiment
2	$ICICIBANK.NSE,$ICICIBANK.NSE Trade in action & Very close to to the targets Bullish,Bullish
3	$ICICIBANK.NSE,$ICICIBANK.NSE Stock is trading with bullish momentum as it is continuously rising and trading above Ema levels. Macd is also bullish and Rsi is in overbought zone. Bullish,Bullish
4	$ICICIBANK.NSE,"$ICICIBANK.NSE
5	Based on the chart analysis, it appears that there is a bullish crossover imminent on the Moving Average Convergence Divergence (MACD) indicator, which could signal a good buying opportunity for the stock. Additionally, the stock has recently crossed both
6	$ICICIBANK.NSE,"$ICICIBANK.NSE crossed the resistance line with low volume, but sustaining.
7	medium strong bullish signals. Bullish",Bullish
8	$ICICIBANK.NSE,"$ICICIBANK.NSE
9	The stock is moving with the upward trend, but today formed a doji pattern. Bullish",Bullish
10	$ICICIBANK.NSE,$ICICIBANK.NSE Look for the next possibility in this counter. Bullish,Bullish
11	$ICICIBANK.NSE,$ICICIBANK.NSE were in the top gainers today with the High day range of 942.45 and the Low day range of 923.45. At 2:30 ICICI bank is trading at 931.50 with 9.35 or 1.01% high from its prior close at 921.70. Bullish,Bullish
12	$ICICIBANK.NSE,"$ICICIBANK.NSE
13	Icici bank is slowly moving towards its target level of 950.
14	FPI shareholding in ICICI Bank remained unchanged at 44 per cent in FY23.
15	If it sustain above 950 then more upside movement is possible. Bullish",Bullish
16	$ICICIBANK.NSE,$ICICIBANK.NSE Stock is trading above Ema levels indicating bullish trend. Macd is also bullish and Rsi is in overbought zone. Bullish,Bullish
17	$ICICIBANK.NSE,"$ICICIBANK.NSE MACD bearish crossover, RSI just over the overbought zone.
18	short term bearish signals. Bearish",Bearish
19	$ICICIBANK.NSE,"$ICICIBANK.NSE
20	GAINING MOMENTUM.!
21	STOCK LOOKS GOOD
22	TODAY CLOSING ABV 926
23	WILL BE MORE UPSIDE TARGET. 955,980.1000 Bullish",Bullish
24	$ICICIBANK.NSE,$ICICIBANK.NSE Trade in action! Bullish,Bullish
25	$ICICIBANK.NSE,"$ICICIBANK.NSE
26	ICICI bank is showing strength after results.
27	Next resistance level is 950 and above that momentum will be great. Bullish",Bullish

Figure 27 : Examples of stock twits' dataset

Two different investigational datasets are collected, including the historical and the comment dataset that calculates the SI. The stock market closing price is considered as a prediction value. In this research work. Three phases of work are carried out to achieve the optimum prediction results. The stock market closing price is considered as a prediction value. The closing price is considered as target value which is compared with the output values of the neural networks.

The stock comment contains comments for model training and a final comment that needs to be classified to find the SI. First, the model was trained using stockholder comments posted on StockTwits "(https://stocktwits.com/)". It is up to investors to mark a comment as bullish or bearish. Even if StockTwits is a well-known public depositor site. This allows for the collection of a large number of incredibly accurate remarks. We crawl the 96,903 comments on the site for training reasons.

The historical datasets contain four features such as open, high, low, and closing prices. StockTwits has nearly ten years' worth of data for twenty-five companies. The stock symbol, message, date, Time, message ID, and user ID for each communication are contained in the corresponding file [75]. Dependencies between stocks in the same sector or business can be attributed to similar market conditions, economic trends, regulatory effects, and industry-specific events, Technology equities, for instance, could be impacted by comparable technologies. Tables 23 and 24 show the details of user comments related to stock and examples are shown in Figure 27.

6.5.1 Parameter settings

The performance of a CNN for sentiment classification can be greatly impacted by several hyperparameters. Table 22 shows the parameters of the CNN method.

Table 22 : Parameters of CNN

PARAMETER NAME	VALUE
Number of layers	10
Number of neurons	100
Batch size	63
Epochs	100
Activation function	TanH and Sigmoidal
Optimizer	Adam
Training size	70 %

6.5.2 Pre-processing

When employing machine learning to anticipate the stock market, pre-processing is essential. Before being properly fed into machine learning models, raw financial data—such as stock prices, trade volumes, and other indications connected to the market—must be thoroughly prepared. The model's capacity to identify patterns, anticipate trends, and produce more accurate forecasts can all be greatly improved by proper pre-processing, which will ultimately improve financial market decision-making. The following sections discussed data cleaning, and normalization and comments pre-processing.

6.5.3 Data cleaning

Data preparation is crucial in machine learning applications since raw data often contains errors including noise, erroneous data, and missing values. Data preprocessing often includes data normalization and standardization, data binarization and type conversion, missing value padding, and denoising. The next step in machine learning pipelines is feature engineering which modifies or creates new features to enhance model performance. Additionally, feature engineering will prevent the training model from overfitting and lower the input dimension Which will significantly increase the training time.

Table 23: The explanation of comments

Dataset	Comments	Size	Function
Comments	Stock comments	96.903 comments	Training classifier
Historical data of the stock market	Comments from Yahoo Finance	80 comments a day for 1219 days	Predicting stock prices

6.5.4 Data normalization

Data normalization is a pre-processing method that is used for rescaling a dataset's arithmetical properties to a predefined range. To ensure that the process can converge more rapidly and to stop some features from dominating over others. The normalization objective to bring all variables to a comparable scale. Furthermore, this process enhances the prediction method’s functionality and raises their struggle to changes in the data.

Stock market data entails transforming financial information into a common range to kind it easier to analyze, assess, and compare data from several securities or periods. The normalization techniques are important to make sure the data is comparable and suitable for modeling. Stock prices can be normalized to a specified range usually between 0 and 1 by using min-max scaling. This method guarantees that all standards fit inside the same scale while keeping the qualified relationships between prices. The formula is for the min-max method as follows,

$$X_i = \frac{X_i - X_{\min}}{X_{\max} - X_{\min}} (X_{\max} - X_{\min}) + X_{\min} \tag{32}$$

Where. $\tilde{X}_{min}$- the minimum. $\tilde{X}_{max}$- maximum target value and X_i are real input data. The particular necessities of the study and the properties of the data regulate which normalizing procedure is best. Additionally, It's critical to assess the results taking into account any possible biases or distortions twisted by the normalization process.

Table 24: User comments

S.No	Bearish	Bullish
1.	"Crazy day so far! Will make a new ATH this week. Watch it"	This spikes the market tanks. Just like in 2021 when this squeezes and dipped 2-4%
2.	"Drop it like it's hot. I think the market is going to rip faces next week. I feel a wave of optimism coming"	It is what we've been waiting for.SHF desperation is unbounded
3.	Everything on my watchlist is still waaaaay too	Let's see how much more this junk pop!!..so this is going ATH again

6.6 Comments pre-processing

Preparing raw text data for analysis or additional natural language processing (NLP) activities entails cleaning and converting it before using it for stock market comments. Comments about the stock market are frequently boisterous and unstructured, especially when they are posted on news forums or social media sites like Twitter, Reddit, and StockTwits. Before using sentiment analysis or predictive modeling, data pretreatment is essential because these texts are frequently unstructured, loud, and full of colloquial jargon. Pre-processing is a critical step in transforming raw text into a structured format suitable for sentiment analysis, trend prediction, and machine learning models. Preprocessing is essential for sentiment analysis and predictive modelling in particular when examining stock market comments. Noise from tickers, URLs, special characters, and numerals is common in stock market comments. Emojis, slang, and acronyms are used by traders. Sentiment detection is enhanced when these terms are standardized. Extracting valuable insights from noisy, informal, and context-dependent material requires preprocessing stock market remarks. The accuracy of sentiment analysis and the predictive capacity of stock sentiment models are both increased by an appropriate preprocessing pipeline. Text cleaning, lowercasing, tokenization, stop words removal, lemmatization, and handling negations are the significant preprocessing techniques which are discussed as follows,

- **Text cleaning:** Remove irrelevant characters. HTML tags, URLs, and special symbols.
- **Lowercasing:** Stock market comments and reviews often mix uppercase and lowercase letters, leading to inconsistencies in analysis. Lowercasing ensures that words like *"Stock"*, *"STOCK"*, and *"stock"* are treated as the same.
- **Tokenization:** Tokenization is a fundamental step in NLP that involves breaking text into smaller units (tokens), such as words or phrases. In stock market comments and reviews, tokenization is crucial for sentiment analysis, trend prediction, and financial text processing. Divide the passage into discrete words or tokens.
- **Stop words removal:** Common words like "is," "the," "and," and "in" are known as stopwords since they don't significantly contribute meaning to financial forecasts or sentiment research. Eliminating them reduces noise, which enhances model performance and efficiency. Eliminate cliches that don't add much to the sentiment (e.g.. "is". "the". "and").
- **Lemmatization/Stemming:** Reduce words to their base or root form to ensure consistency. Stock market discussions contain multiple forms of the same word (e.g., buy, buying, bought). Lemmatization and stemming normalize them into a single form, reducing redundancy. The sentiment of the stock market depends on words like "bullish," "bearish," "crash," and "rally." Lemmatization guarantees that various word forms accurately contribute to sentiment detection.
- **Handling negations:** Detect and properly handle negations (e.g.. "not good" should be preserved contrarily from "good"). Negative remarks have the power to drastically alter the tone of stock-related discussions, which guarantees precise sentiment categorization for trading tactics. Discussions on the stock market frequently involve cautious remarks and negations, which need to be treated with caution. It assists in examining cautious market mood rather than incorrectly classifying it as positive.

As part of the current study, one of the initial steps is to incorporate users' sentimentality favorites for the stock with historical data on the stock as a predictor of the stock's closing price. The sentiment index is determined by counting how many integrated users post bullish or bearish comments each day. Thus, we must first determine the accurate sentiment categorization of the individual stock review to compute the sentiment index and obtain the group sentiment trend. In contrast to the CNN base model, we combine it with word2vec by altering the word vector's initialization.

Word2vec is a model for unsupervised learning of semantic information from a huge corpus of text. Mapping words from the original multidimensional space to the new one is the key to using word2vec. In particular. when learning the text. The word vector represents the semantic information of the word. and upon embedding the space. Semantically related words are mapped to comparable distances. We conduct softmax normalization and compute the cosine similarity between the input vector and the target word's output vector.

This research work introduces word2vec Which is used to learn high-dimensional vector representations of phrases by first training large-scale stock comment corpora. Next. word2vec computes the word vector representation of the stock comments that need to be categorized. The result is used immediately if the phrase in the stock comments that have to be classified is present in the trained large-scale stock comments corpus. If not, word2vec initializes it at random.

6.7 Hyper parameters analysis

An ANN's performance and capacity for generalization can be greatly improved through effective hyper parameter optimization which will increase the model's dependability and accuracy in real-world scenarios. In ANNs, optimizing hyper parameters is essential to enhancing model performance. Hyper parameters are configurations that are predetermined before the training process starts and are not learned from the data. Improving these can result in more effective training and improved generalization. The weights between the input and hidden layers of ELM is kind of feedforward neural network with a single hidden layer that is fixed and randomly given; only the output weights are learned.

This property makes ELMs exceptionally quick to train. But it also means that high performance requires hyper parameter adjustment. Particularly for complicated tasks like stock market prediction. The ELM algorithm consists of several levels Including input, Hidden, and output layers. The explanation of the input and output layer neurons is based on particular applications. Imagine that one output neuron, like the closing price or index has 10 input neurons, which are the ten technical indicators covered. In the SLFN model, the number of hidden layer neurons is significant, but in the ELM technique, there is just one hidden layer. Since under-fitting will occur from having too few neurons and over-fitting will occur from having too many. The prediction effect will be diminished. This study continuously assesses and updates the hidden layer node count of the network to determine an optimal number of nodes. The hyper parameters of ELM have a significant impact on their behaviour and performance. One of the characteristics of ELM, a kind of feedforward neural network

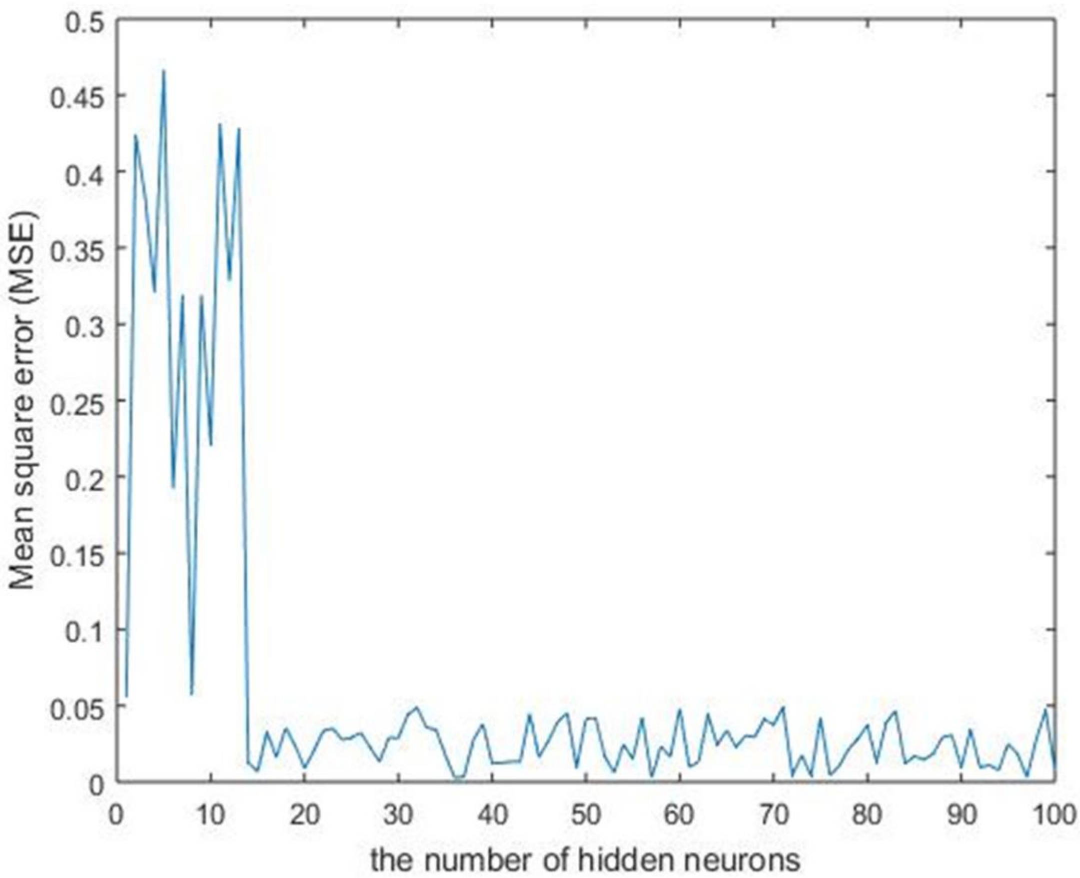


Figure 28: Hidden neurons vs MSE

with a single hidden layer. is that the weights between the input and hidden layers are assigned at random and are not modified during training. Rather, Only the weights that are shared by the output and the hidden layer are learned. The following are the main hyper parameters in ELM and what they mean:

6.7.1 Hidden neurons

The ability of the model to learn from the data is dependent on the quantity of hidden neurons. Although additional neurons can detect more intricate patterns that Having too many of them about the number of training examples might cause overfitting. While the ELM's learning capacity is often enhanced by increasing the number of hidden neurons. Doing so can also raise processing costs and increase the chance of overfitting. The maximum number of hidden neurons in the ELM network is 100. Figure 28 shows that the MSE of the learning model is lowest when there are 35 hidden neurons. A 0.0005 stopping error has been reported.

6.7.2 Activation function

The model gains non-linearity from the activation function of the hidden neurons which enables it to capture more intricate relationships in the data. Sigmoid, TanH, and ReLU are popular options. The convergence and effectiveness of the ELM are impacted by the selection of the activation function. Certain tasks and data types may be better suited for certain activation functions.

6.7.3 Weights and biases

The network's initial setup may be impacted by the random seed selection because the weights between the input and hidden layers are assigned at random. Performance outcomes can vary depending on the random seed used. The repeatability of the outcomes is ensured by using a set random seed.

The findings are calculated using an average of thirty separate runs. The ELM's activation function plays a crucial part in determining prediction accuracy as well. As input and output phase activation functions, the sigmoid function is used. The maximum epoch count is fixed at 1000. The forecast accuracy is also determined by the two parameters in the DWM algorithm when selecting the optimal values. Hence, the value of both is set as $\lambda = 0.2$. $\beta = 0.7$ based on [21].

6.8 Performance measures

The accuracy and efficacy of stock market prediction models in predicting future stock prices or trends are evaluated through performance criteria. Performance indicators offer important information about how robust, accurate, and reliable stock market prediction algorithms. To fully assess model performance, many indicators may be utilized in combination or prioritized based on the particular goals and specifications of the prediction task. The performance is measured based on MSE, RMSE, MAE, MAPE, and R-Square (R^2) which are discussed as follows.

6.8.1 MSE

A technique called the MSE calculates the difference between the estimated and real values.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \bar{y}_i)^2 \tag{33}$$

6.8.2 RMSE

The RMSE is a useful statistic that is frequently used to validate experimental results. It can have positive or negative residuals depending on whether the predicted value overestimates or underestimates the actual values. It is defined as follows.

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \bar{y}_i)^2} \tag{34}$$

6.8.3 MAE

One of the most effective ways to gauge forecast accuracy is to use MAE and it is a frequently utilized error metric since it is used to formulate learning issues to optimistic challenges.

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|(y_i - \bar{y}_i)| \tag{35}$$

6.8.4 MAPE

The average of the percentage of mistakes is known as the MAPE which is considered by dividing the total of the individual absolute errors by the demand. It is also one of the measures that is frequently used to evaluate the degree of forecast accuracy.

$$MAPE = \frac{100}{m}\sum_{i=1}^{m}\left|\frac{(y_i - \bar{y}_i)}{y_i}\right| \tag{36}$$

6.8.5 R-square (R^2)

The percentage of a dependent variable's variance that is clarified by more than one independent variable is expressed statistically as R-square (R^2). R-square can be used to assess the goodness-of-fit of a regression. Regression models aim to forecast stock prices or returns based on a variety of independent variables including technical indications, economic indicators, corporate fundamentals, and so on.

$$R^2 = 1 - \frac{(\sum_{i=1}^{N}(y_i - \bar{y}_i)^2)/N}{(\sum_{i=1}^{N}(\hat{y}_i - \bar{y}_i)^2)/N} \tag{37}$$

Where. y_i – target value. $\bar{y}_i$ - predicted output respectively. N –number of data samples. $\hat{y}_i$ - mean of real values. The value of $R^2$is closer to 1. stronger ability and better the model.

The ML community will be better able to choose and use the appropriate metrics if they have a thorough understanding of the many error measures available. Determining and quantifying errors facilitates problem analysis that improves decision-making from a business perspective and aids in managing bottlenecks.

6.9 Experimental Results and Analysis

Phase III uses sentiment analysis to predict the stock market price based on customer sentiment. The CNN algorithm is used to classify the user comments into bullish/bearish. The present section considers both technical indicators and the sentiment of users as input to the developed DELM is called S-DELM. The present sub-section discusses the ability of the developed S-DELM method. Tables 25, 27, 29, 31, and 33 show the training performance of MSE, RMSE, MAE, MAPE, and R-Square values, respectively.

Tables 26, 28, 30, 32, and 34 show the testing performance of MSE, RMSE, MAE, MAPE, and R-Square values, respectively. Figures 29, 31, 33, 35, and 37, show the training performance of MSE, RMSE, MAE, MAPE, and R-Square values, respectively. Figures 30, 32, 34, 36, and 38, show the testing performance of MSE, RMSE, MAE, MAPE, and R-Square values, respectively.

In the training phase, the suggested technique obtained low MSE values when compared to other prediction algorithms, such as 0.000521, 0.000894, 0.000621, 0.000227, 0.000302, and 0.000487 for S&P, Nifty 50, SBIN, ICICI, HDFC, and MSFT, respectively.

The RMSE value of the suggested technique is low when compared to other prediction algorithms such as 0.02282, 0.02989, 0.02491, 0.01506, 0.01737, and 0.02206 for S&P, Nifty 50, SBIN, ICICI, HDFC, and MSFT, respectively. The suggested technique obtained low MAE values when compared to other prediction algorithms, such as 0.0034, 0.0058, 0.0027, 0.0029, 0.0034, and 0.0043 for S&P, Nifty 50, SBIN, ICICI, HDFC, and MSFT, respectively.

The suggested technique obtained low MAPE values when compared to other prediction algorithms, such as 0.0004, 0.0058, 0.0004, 0.0004, 0.0003, and 0.0006 for S&P, Nifty 50, SBIN, ICICI, HDFC, and MSFT, respectively. The suggested technique obtained high R-square values when compared to other prediction algorithms, such as 0.9998, 0.9997, 0.9999, 0.9998, 0.9956, and 0.9893 for S&P, Nifty 50, SBIN, ICICI, HDFC, and MSFT, respectively.

Table 25 : Training performance comparisons for MSE

Methods	S-DELM	DELM-T	DELM-H	GAELM	KELM	ELM	SVM	BPNN
S & P	0.000521	0.000628	0.000738	0.000843	0.001291	0.00482	0.006293	0.009061
Nifty 50	0.000894	0.001024	0.000129	0.001365	0.001517	0.001738	0.001995	0.002155
SBIN	0.000621	0.000792	0.000875	0.000937	0.001045	0.001591	0.001836	0.002067
ICICI	0.000227	0.000596	0.000758	0.000863	0.000983	0.001135	0.001455	0.001864
HDFC	0.000302	0.000552	0.000691	0.000821	0.000973	0.001847	0.002851	0.003195
MSFT	0.000487	0.000587	0.000653	0.000694	0.000792	0.000984	0.001587	0.001918

Table 26: Testing performance comparisons for MSE

Methods	S-DELM	DELM-T	DELM-H	GAELM	KELM	ELM	SVM	BPNN
S & P	0.00084	0.000959	0.00105	0.00122	0.00294	0.00346	0.00598	0.00718
Nifty 50	0.00111	0.00369	0.00582	0.00768	0.00892	0.01029	0.01529	0.01989
SBIN	0.000191	0.000479	0.00068	0.000865	0.000942	0.001191	0.001395	0.001759
ICICI	0.000127	0.000356	0.00048	0.000658	0.000893	0.001221	0.001426	0.001852
HDFC	0.000199	0.000489	0.00059	0.000798	0.000931	0.001194	0.001352	0.001698
MSFT	0.00156	0.00467	0.00061	0.00695	0.00749	0.00897	0.01099	0.0201

Table 27: Training performance comparisons for RMSE

Methods	S-DELM	DELM-T	DELM-H	GAELM	KELM	ELM	SVM	BPNN
S & P	0.02282	0.02505	0.02716	0.02903	0.03593	0.06942	0.07932	0.09518
Nifty 50	0.02989	0.03253	0.01135	0.03694	0.03894	0.04168	0.04466	0.04642
SBIN	0.02491	0.02814	0.02958	0.03061	0.03232	0.03988	0.04284	0.04546
ICICI	0.01506	0.02441	0.02753	0.02937	0.03135	0.03368	0.03814	0.04317
HDFC	0.01737	0.02349	0.02628	0.02865	0.03119	0.04297	0.05339	0.05652
MSFT	0.02206	0.02422	0.02555	0.02634	0.02814	0.03136	0.03983	0.04379

Table 28: Testing performance comparisons for RMSE

Methods	S-DELM	DELM-T	DELM-H	GAELM	KELM	ELM	SVM	BPNN
S & P	0.02898	0.03096	0.03240	0.03492	0.05422	0.05882	0.07733	0.08473
Nifty 50	0.03331	0.06074	0.07628	0.08763	0.09444	0.10144	0.12365	0.14103
SBIN	0.01382	0.02188	0.02607	0.02941	0.03069	0.03451	0.03735	0.04194
ICICI	0.01126	0.01886	0.02190	0.02565	0.02988	0.03494	0.03776	0.04303
HDFC	0.01410	0.02211	0.02429	0.02824	0.03051	0.03455	0.03677	0.04120
MSFT	0.03949	0.06833	0.02469	0.08336	0.08654	0.09471	0.10483	0.14177

Table 29 : Training performance comparisons for MAE

Methods	S-DELM	DELM-T	DELM-H	GAELM	KELM	ELM	SVM	BPNN
S & P	0.0034	0.0069	0.0079	0.0095	0.0106	0.0381	0.0451	0.0621
Nifty 50	0.0058	0.007	0.0084	0.0092	0.0192	0.0281	0.041	0.0515
SBIN	0.0027	0.0045	0.0051	0.0059	0.00631	0.0072	0.0095	0.0103
ICICI	0.0029	0.0045	0.0058	0.0069	0.0081	0.0172	0.0377	0.0412
HDFC	0.0034	0.0052	0.0063	0.0072	0.0079	0.0085	0.0093	0.01016
MSFT	0.0043	0.0055	0.0067	0.0075	0.0084	0.0095	0.0101	0.0115

Table 30 : Testing performance comparisons for MAE

Methods	S-DELM	DELM-T	DELM-H	GAELM	KELM	ELM	SVM	BPNN
S & P	0.0079	0.0098	0.0102	0.0106	0.0119	0.0125	0.0141	0.0152
Nifty 50	0.0308	0.0652	0.0752	0.0886	0.0938	0.1022	0.1256	0.1495
SBIN	0.0101	0.0115	0.0119	0.0124	0.0142	0.0172	0.0199	0.0208
ICICI	0.0081	0.0099	0.0122	0.0153	0.0291	0.0315	0.0515	0.0821
HDFC	0.0117	0.0134	0.0138	0.0145	0.0156	0.0176	0.0181	0.0201
MSFT	0.0096	0.0099	0.0143	0.0127	0.0372	0.0511	0.0676	0.0859

Table 31: Training performance comparisons for MAPE

Methods	S-DELM	DELM-T	DELM-H	GAELM	KELM	ELM	SVM	BPNN
S & P	0.0004	0.0016	0.0024	0.0035	0.0059	0.0069	0.0088	0.0105
Nifty 50	0.0058	0.0064	0.0079	0.0092	0.0102	0.0221	0.0281	0.0315
SBIN	0.0004	0.0029	0.0046	0.0056	0.0064	0.0071	0.009	0.011
ICICI	0.0004	0.0006	0.0008	0.0009	0.0018	0.0026	0.0039	0.0090
HDFC	0.0003	0.0045	0.0057	0.0068	0.0072	0.0079	0.0094	0.0101
MSFT	0.0006	0.0053	0.0059	0.0063	0.0071	0.0078	0.0099	0.0105

Table 32: Testing performance comparisons for MAPE

Methods	S-DELM	DELM-T	DELM-H	GAELM	KELM	ELM	SVM	BPNN
S & P	0.0058	0.0063	0.0068	0.0072	0.0080	0.0086	0.0098	0.0095
Nifty 50	0.0037	0.0049	0.0053	0.0065	0.0078	0.0086	0.0094	0.0105
SBIN	0.0016	0.0024	0.0031	0.0035	0.0051	0.0072	0.0099	0.0109
ICICI	0.0011	0.0025	0.0037	0.0041	0.0067	0.0098	0.0139	0.0193
HDFC	0.0011	0.0026	0.0039	0.0055	0.0069	0.0081	0.0095	0.0111
MSFT	0.0014	0.0056	0.0061	0.0065	0.0076	0.0081	0.0095	0.0112

Table 33: Training performance comparisons for R-Square

Methods	S-DELM	DELM-T	DELM-H	GAELM	KELM	ELM	SVM	BPNN
S & P	0.9998	0.9762	0.9672	0.9467	0.9174	0.8999	0.8544	0.8197
Nifty 50	0.9997	0.9624	0.9461	0.9165	0.8954	0.8542	0.8192	0.7615
SBIN	0.9999	0.9628	0.9494	0.9154	0.8926	0.8762	0.8462	0.8164
ICICI	0.9998	0.9686	0.9572	0.9286	0.9051	0.8725	0.8246	0.7916
HDFC	0.9956	0.9652	0.9408	0.9358	0.9027	0.8726	0.8291	0.8165
MSFT	0.9893	0.9541	0.9437	0.9248	0.9027	0.871	0.8389	0.8034

Table 34: Testing performance comparisons for R-Square

Methods	S-DELM	DELM-T	DELM-H	GAELM	KELM	ELM	SVM	BPNN
S & P	0.9958	0.9539	0.9437	0.9294	0.9027	0.8834	0.8564	0.8164
Nifty 50	0.9983	0.9662	0.9601	0.9468	0.9121	0.8598	0.8262	0.8062
SBIN	1	0.9954	0.9785	0.9564	0.9256	0.8718	0.8302	0.7861
ICICI	0.9959	0.9622	0.9495	0.9387	0.8836	0.8591	0.8261	0.7864
HDFC	1	0.9866	0.9782	0.9527	0.9293	0.8901	0.8519	0.7909
MSFT	0.9861	0.9621	0.9467	0.9325	0.9027	0.8852	0.8561	0.8259

In the testing phase, when compared to previous prediction algorithms the recommended technique produced low MSE values of 0.00084, 0.00111, 0.000191, 0.000127, 0.000199, and 0.00156 for S&P, Nifty 50, SBIN, ICICI, HDFC, and MSFT, respectively. When compared to the aforementioned prediction algorithms, the suggested technique achieved low RMSE values of 0.02898, 0.03331, 0.01382, 0.01126, 0.01410, and 0.03949 for S&P, Nifty 50, SBIN, ICICI, HDFC, and MSFT, respectively. When compared to previous prediction algorithms, the recommended technique produced low MAE values of 0.0079, 0.0308, 0.0101, 0.0081, 0.0117, and 0.0096 for S&P, Nifty 50, SBIN, ICICI, HDFC, and MSFT, respectively.

When compared to the aforementioned prediction algorithms, the suggested technique achieved low MAPE values of 0.0058, 0.0037, 0.0016, 0.0011, 0.0011, and 0.0014 for S&P, Nifty 50, SBIN, ICICI, HDFC, and MSFT, respectively. The recommended technique outperformed other prediction algorithms with values of 0.9958, 0.9983, 1, 0.9959, 1, and 0.9861 for S&P, Nifty 50, SBIN, ICICI, HDFC, and MSFT, respectively.

As a result, the values obtained for MSE, RMSE, MAE, and MAPE are low compared to other prediction algorithms which show that the suggested S-DELM performance is better. The R-Square values of the proposed technique are high compared to other prediction algorithms which again proves that the proposed S-DELM method is producing better accuracy and low error rate.

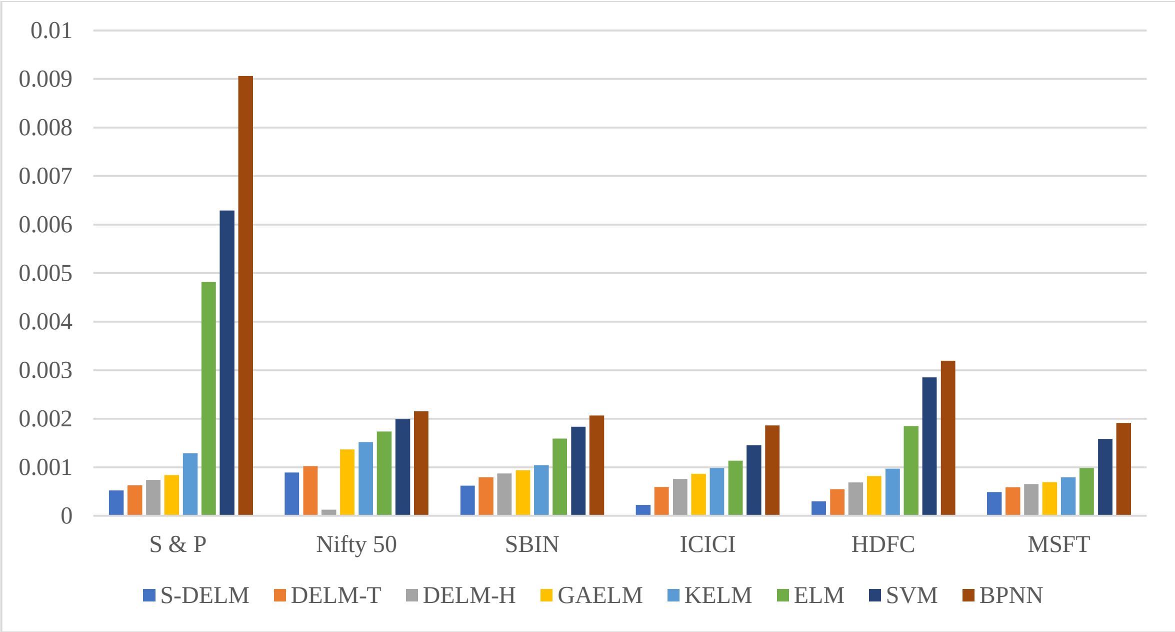


Figure 29 : Training performance comparisons for MSE

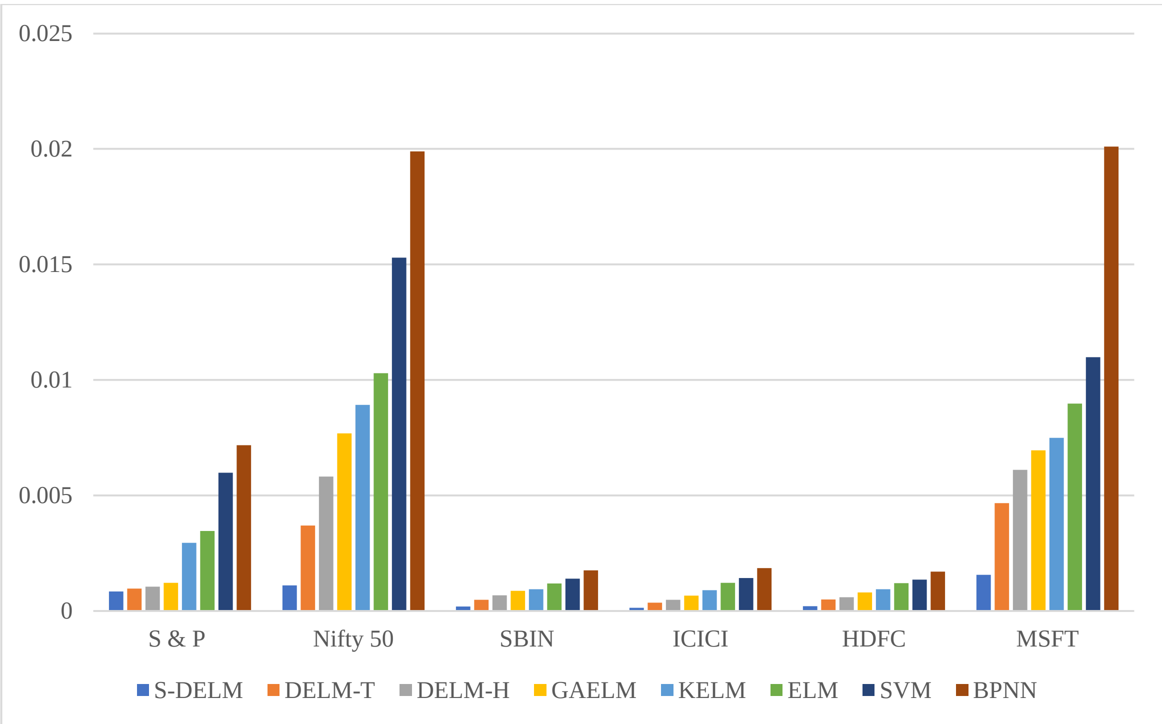


Figure 30 : Testing performance comparisons for MSE

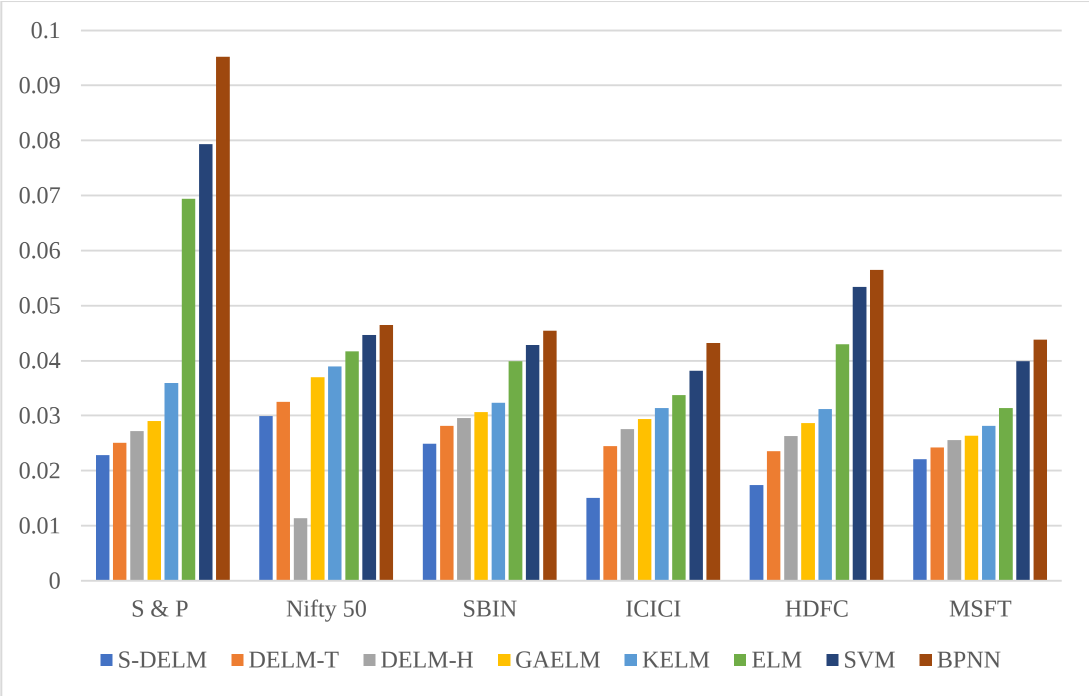


Figure 31 : Training performance comparisons for RMSE

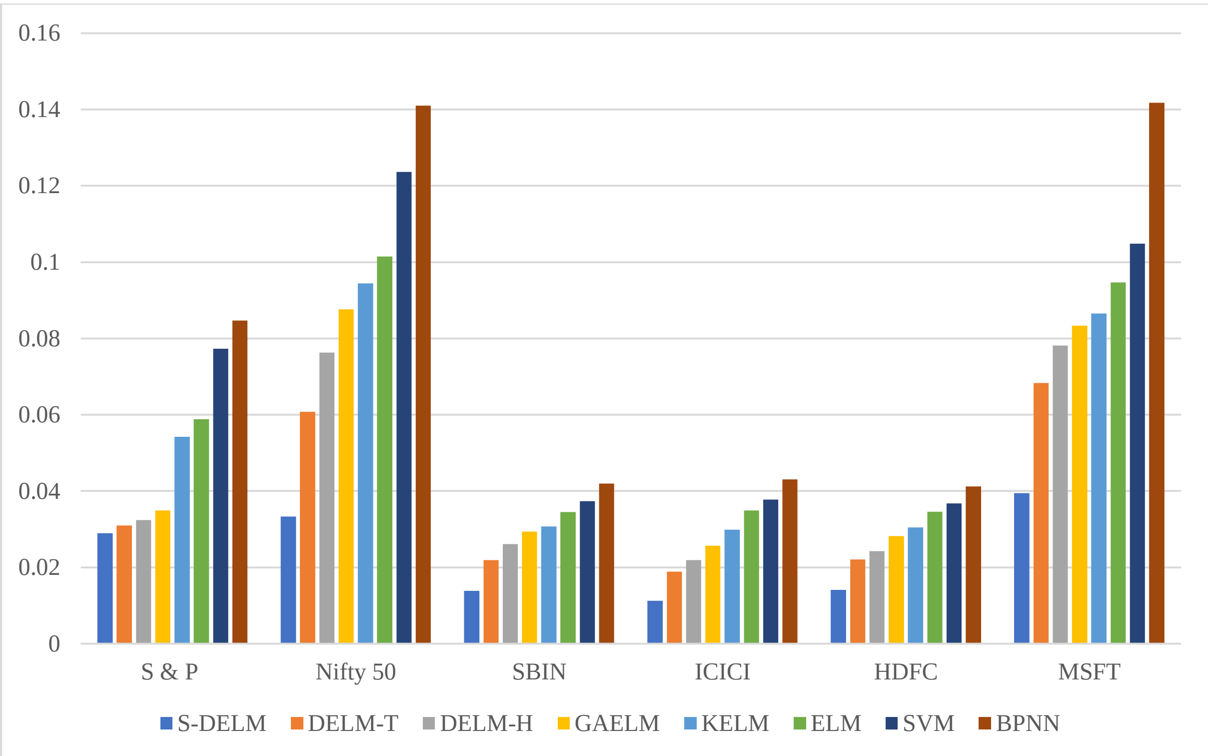


Figure 32 : Testing performance comparisons for RMSE

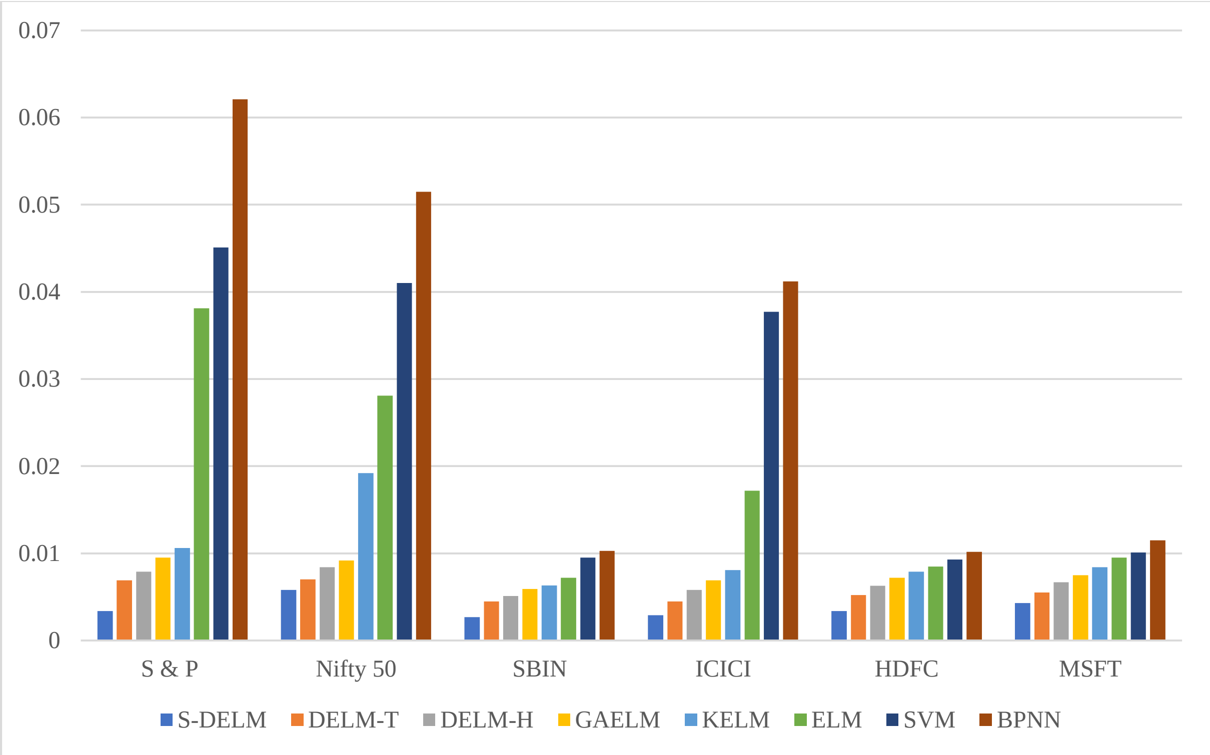


Figure 33 : Training performance comparisons for MAE

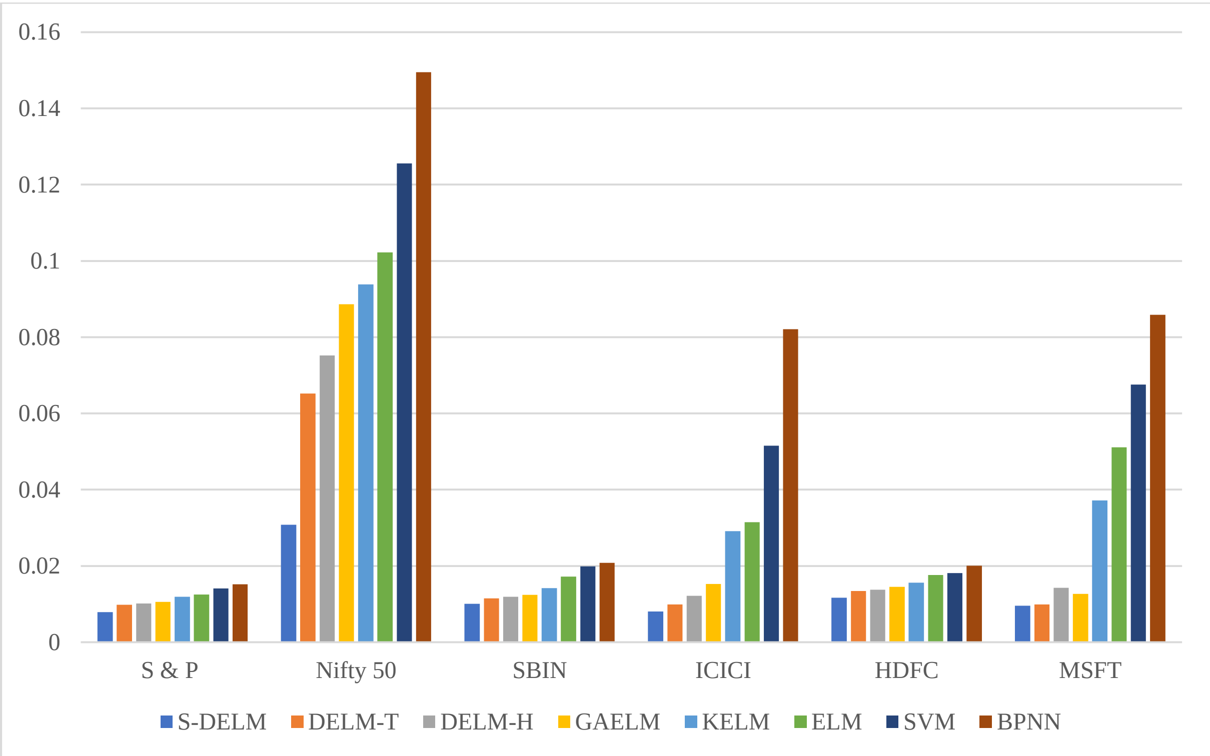


Figure 34 : Testing performance comparisons for MAE

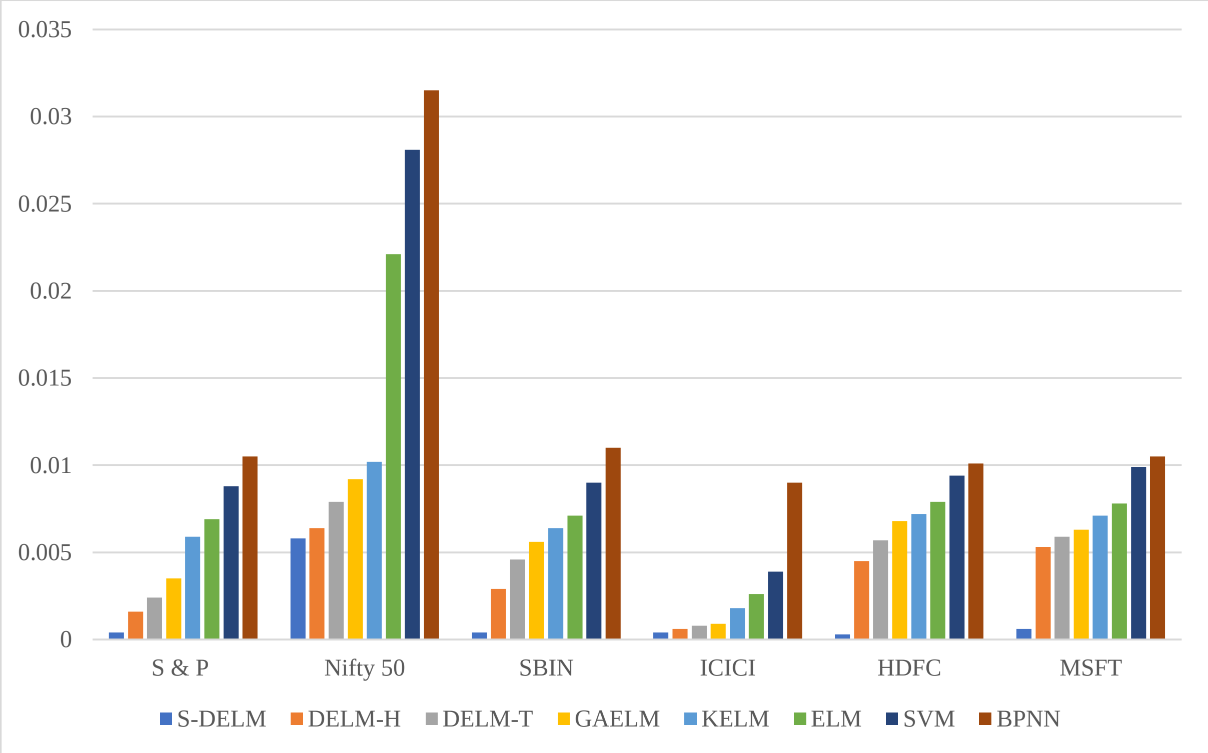


Figure 35 : Training performance comparisons for MAPE

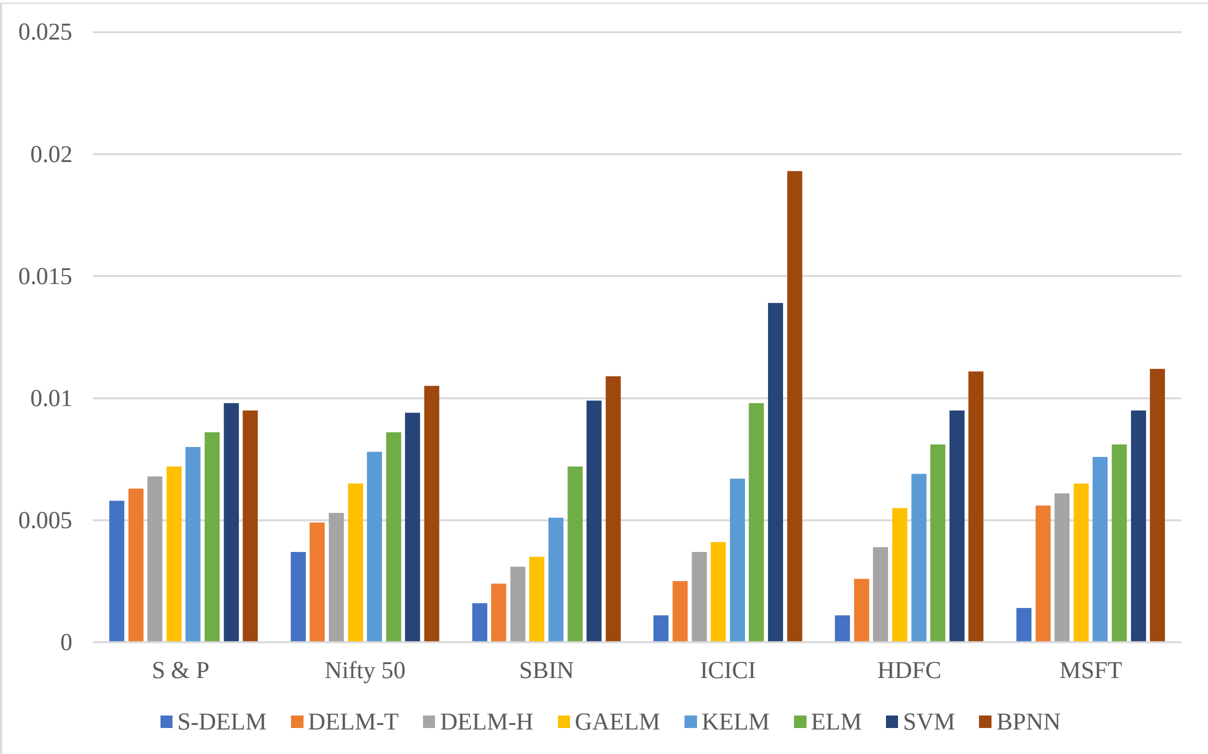


Figure 36 : Testing performance comparisons for MAPE

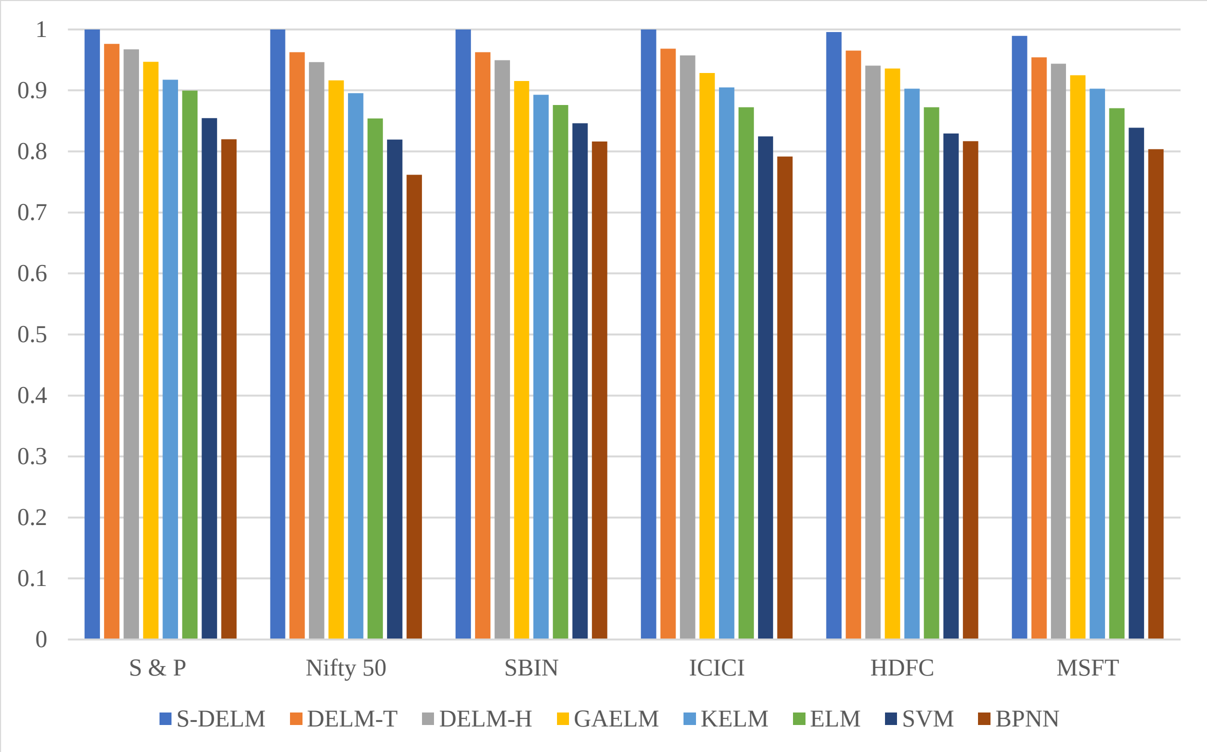


Figure 37 : Training performance comparisons for R-Square

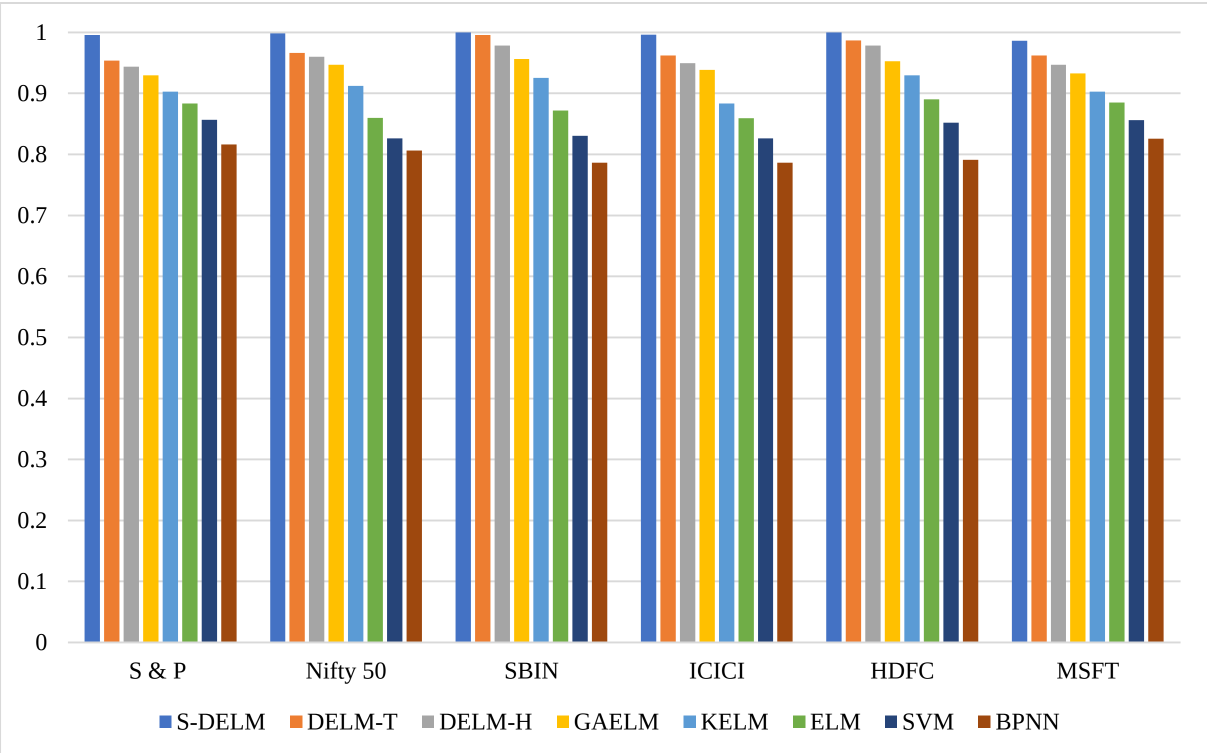


Figure 38 : Testing performance comparisons for R-Square

Table 35 : Training computation time

Methods	S-DELM	DELM-T	DELM-H	GAELM	KELM	ELM
S & P	132.85	135.94	149.27	158.25	0.0418	0.0587
Nifty 50	140.94	142.94	150.71	167.84	0.0618	0.0637
SBIN	135.94	140.51	146.94	155.39	0.0527	0.0539
ICICI	138.52	143.27	146.73	150.86	0.0505	0.0552
HDFC	138.64	142.68	145.72	148.55	0.0559	0.0608
MSFT	134.67	139.81	140.31	143.95	0.0517	0.0495

Table 36 : Testing computation time

Methods	S-DELM	DELM-T	DELM-H	GAELM	KELM	ELM
S & P	0.0146	0.0267	0.0289	0.0348	0.001582	0.001907
Nifty 50	0.0128	0.0253	0.0263	0.0294	0.001957	0.002158
SBIN	0.0142	0.0268	0.0297	0.0318	0.001658	0.001895
ICICI	0.0184	0.0237	0.0253	0.0287	0.001837	0.002067
HDFC	0.0192	0.0255	0.0271	0.0302	0.001737	0.001995
MSFT	0.0183	0.0239	0.0247	0.0299	0.001673	0.001871

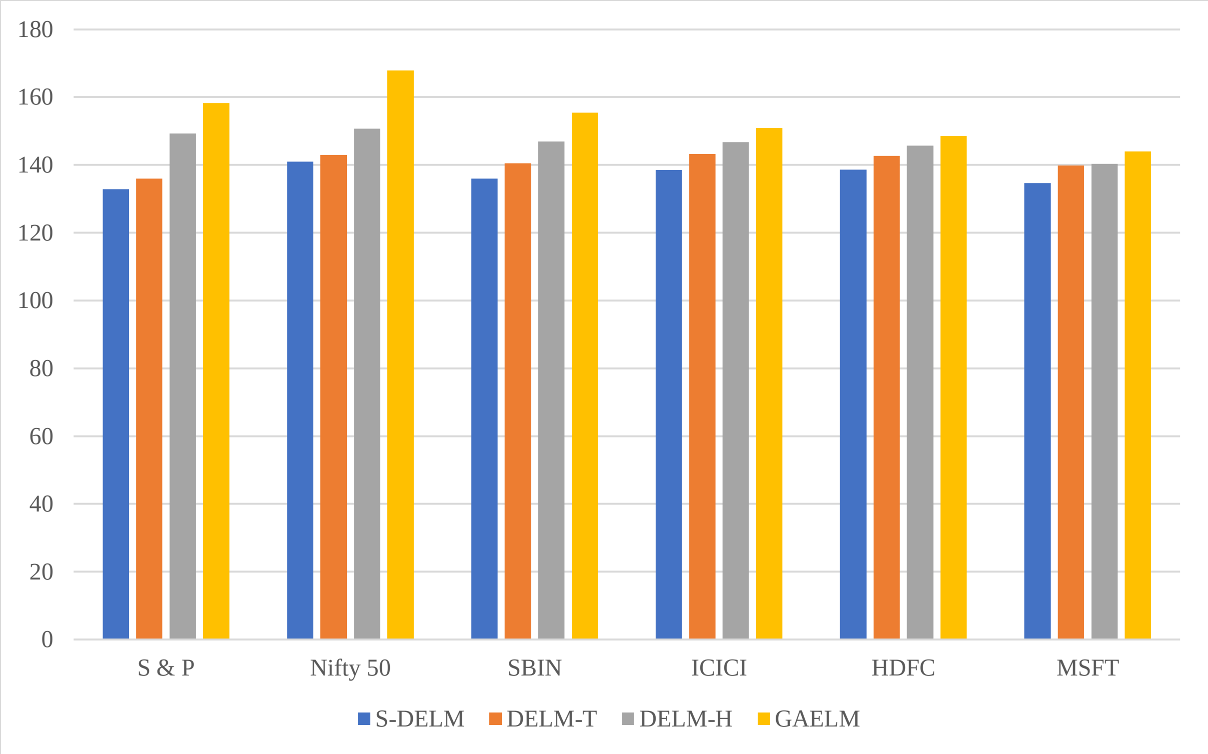


Figure 39 : Training computation time

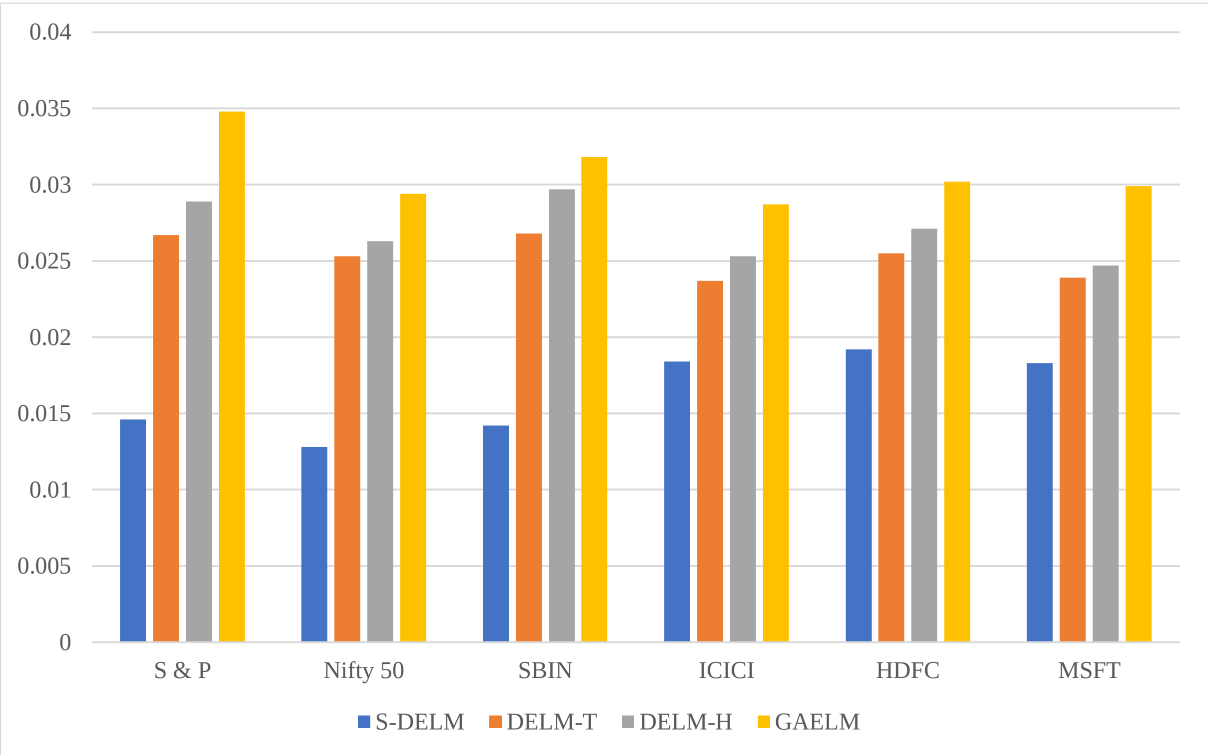


Figure 40 : Testing computation time

Figure 41 : Training performance comparisons for the S&P Sensex dataset

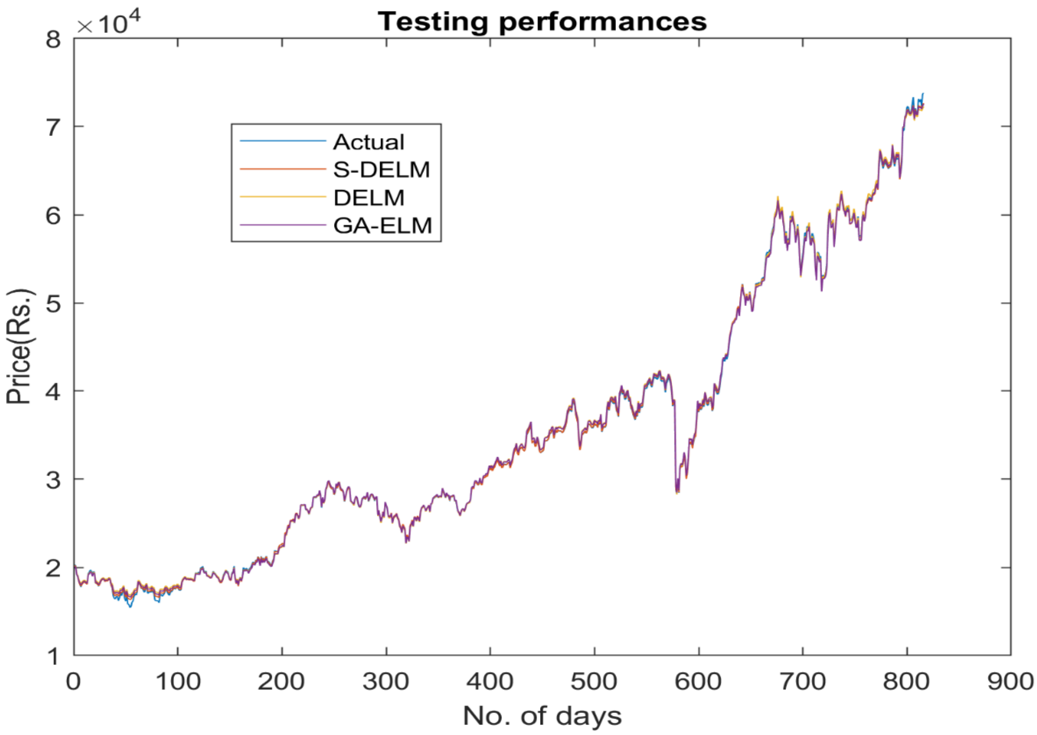


Figure 42 : Testing performance comparisons for the S&P Sensex dataset

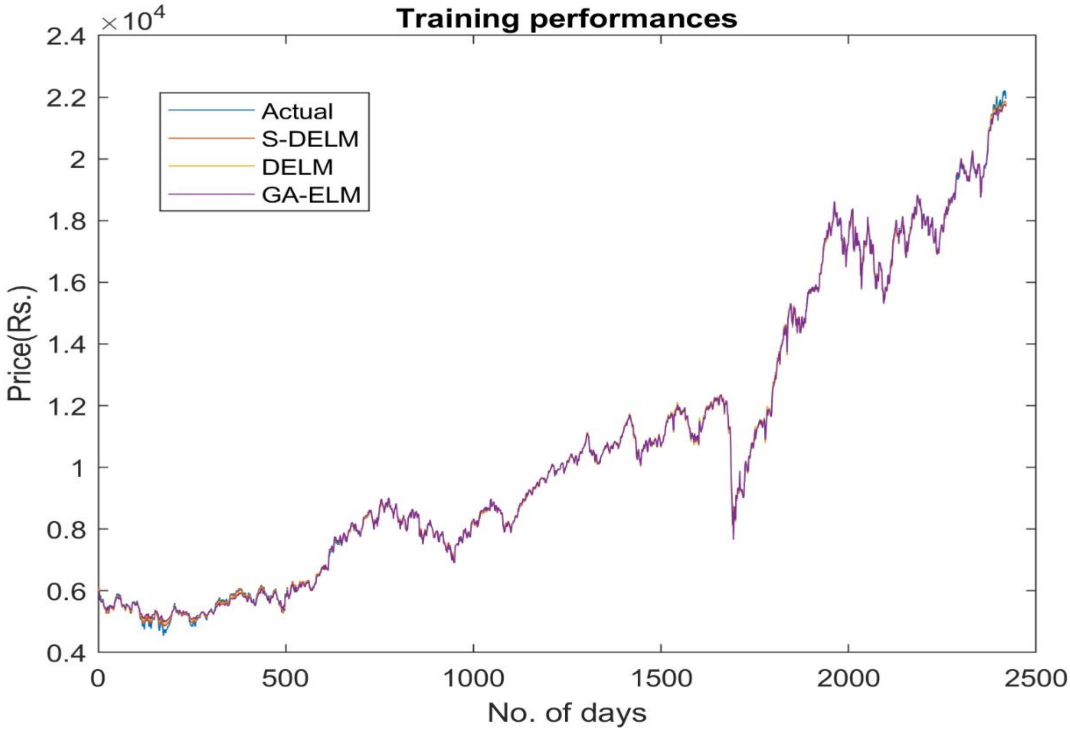


Figure 43: Training performance comparisons for Nifty 50 dataset

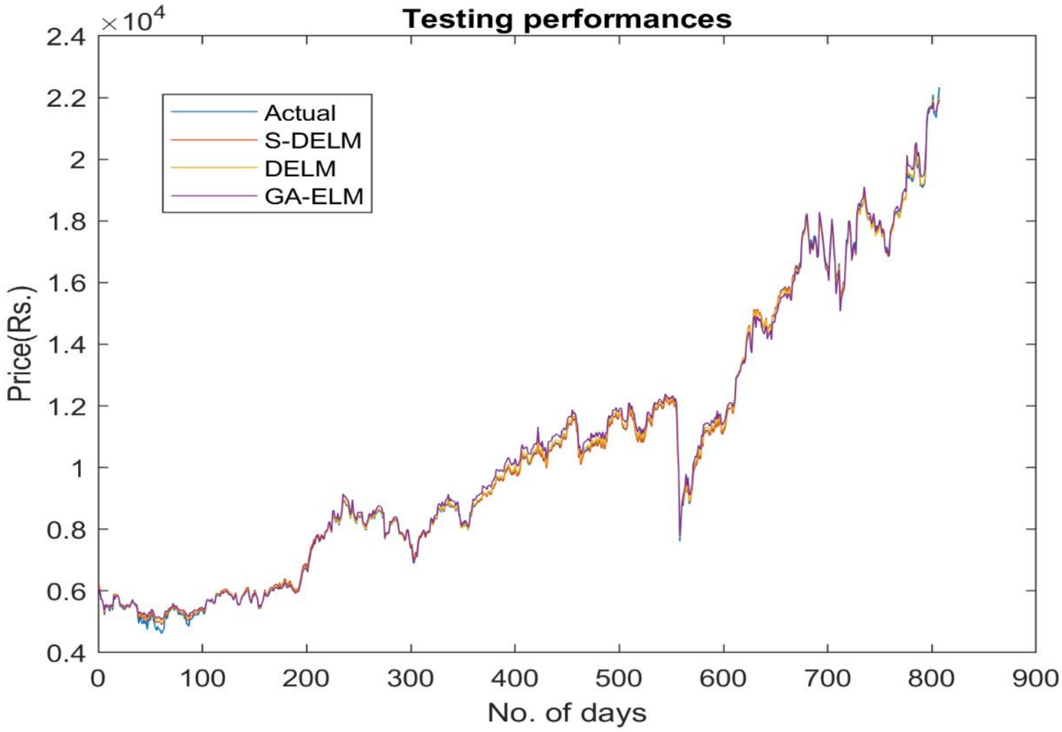


Figure 44: Testing performance comparisons for Nifty 50 dataset

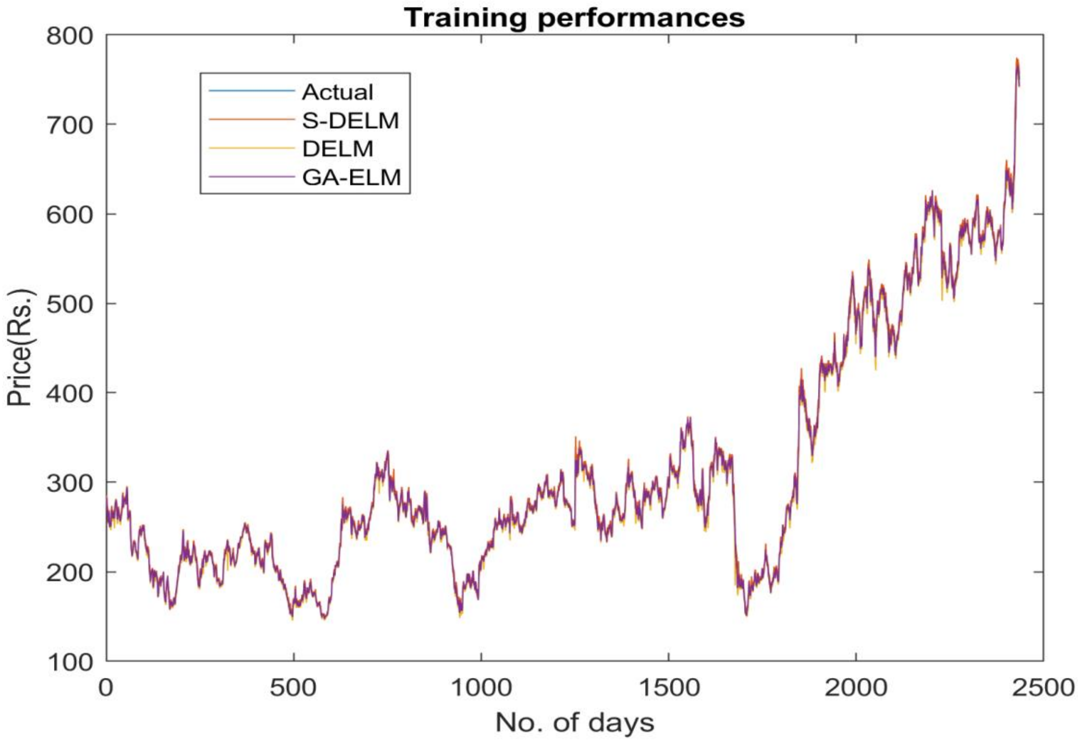


Figure 45: Training performance comparisons for the SBIN dataset

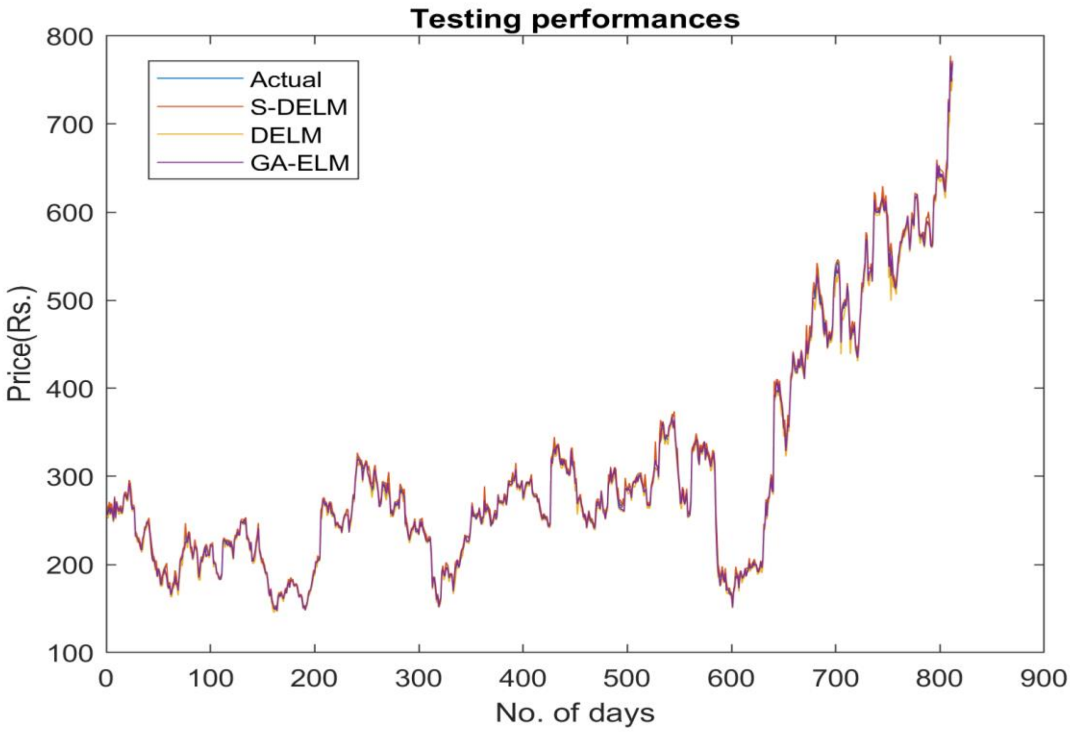


Figure 46: Testing performance results comparisons for the SBIN dataset

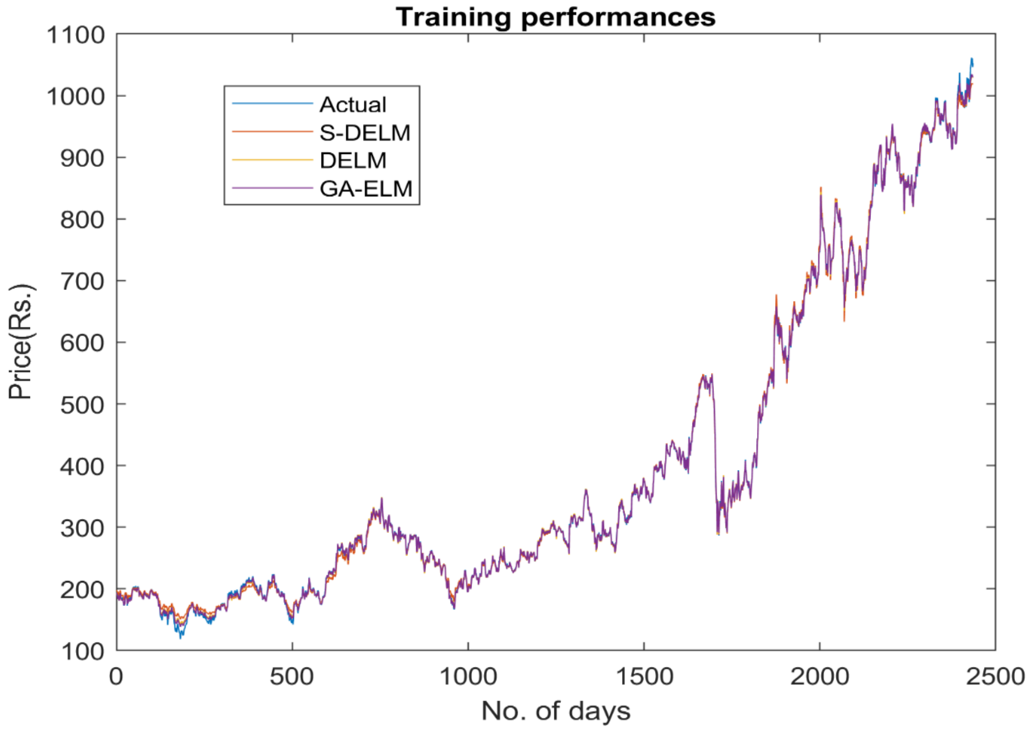


Figure 47: Training performance comparisons for the ICICI dataset

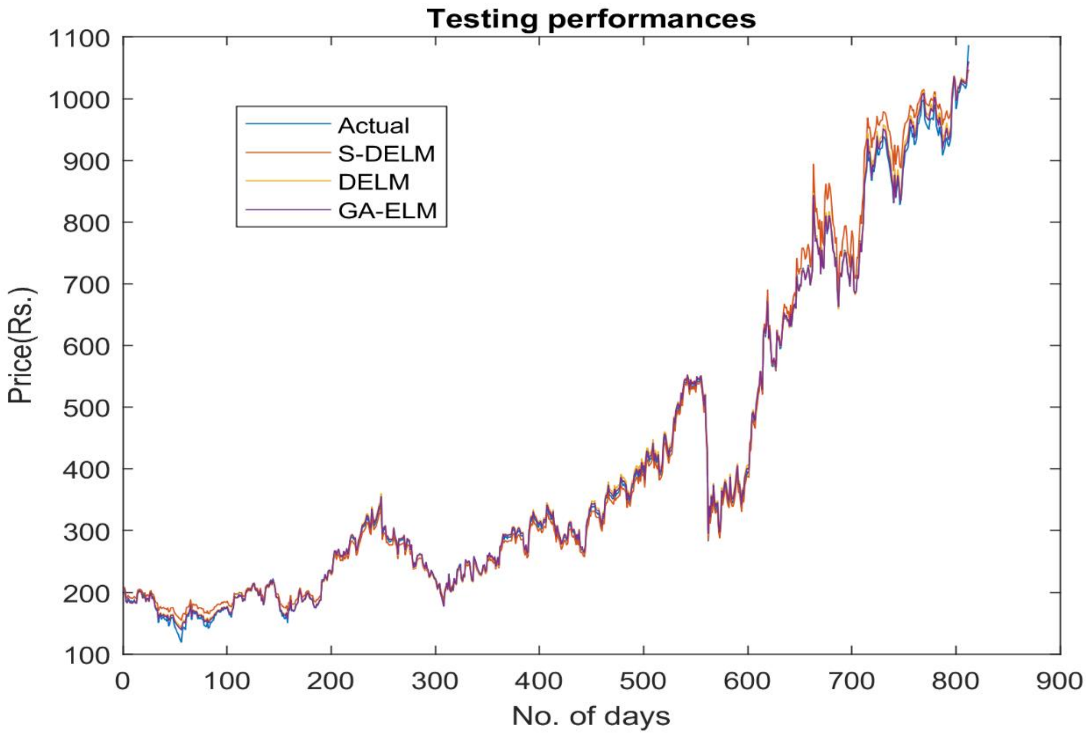


Figure 48: Testing performance comparisons for the ICICI dataset

Figure 49: Training performance comparisons for the HDFC dataset

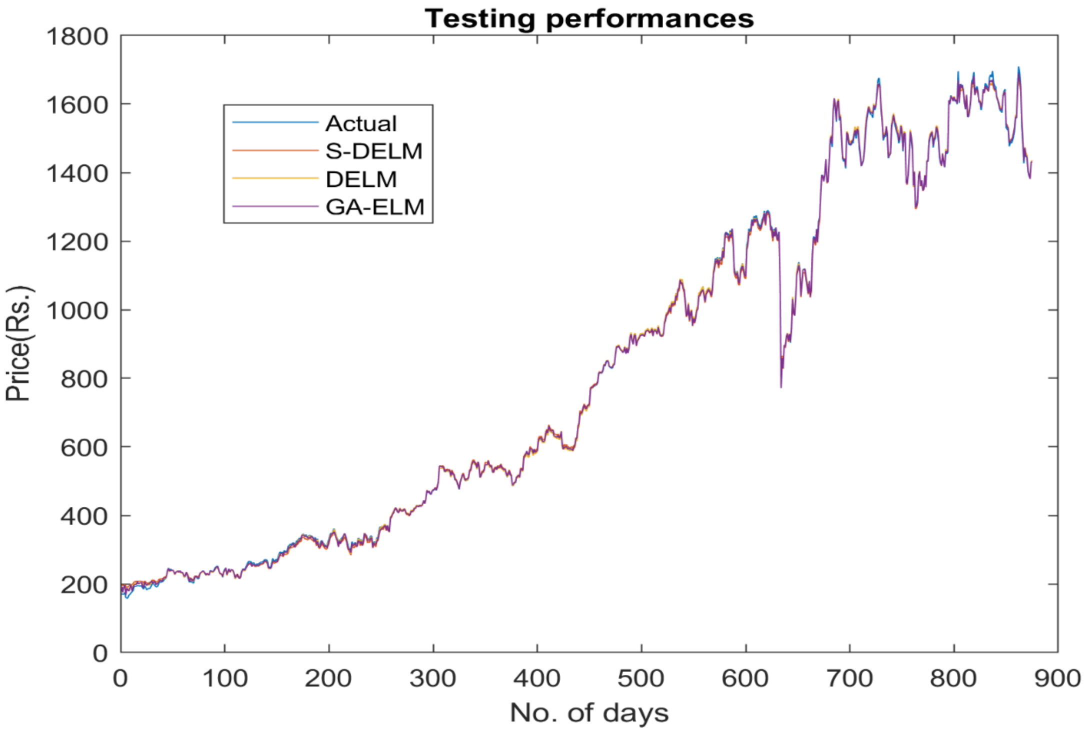


Figure 50: Testing performance comparisons for HDFC dataset

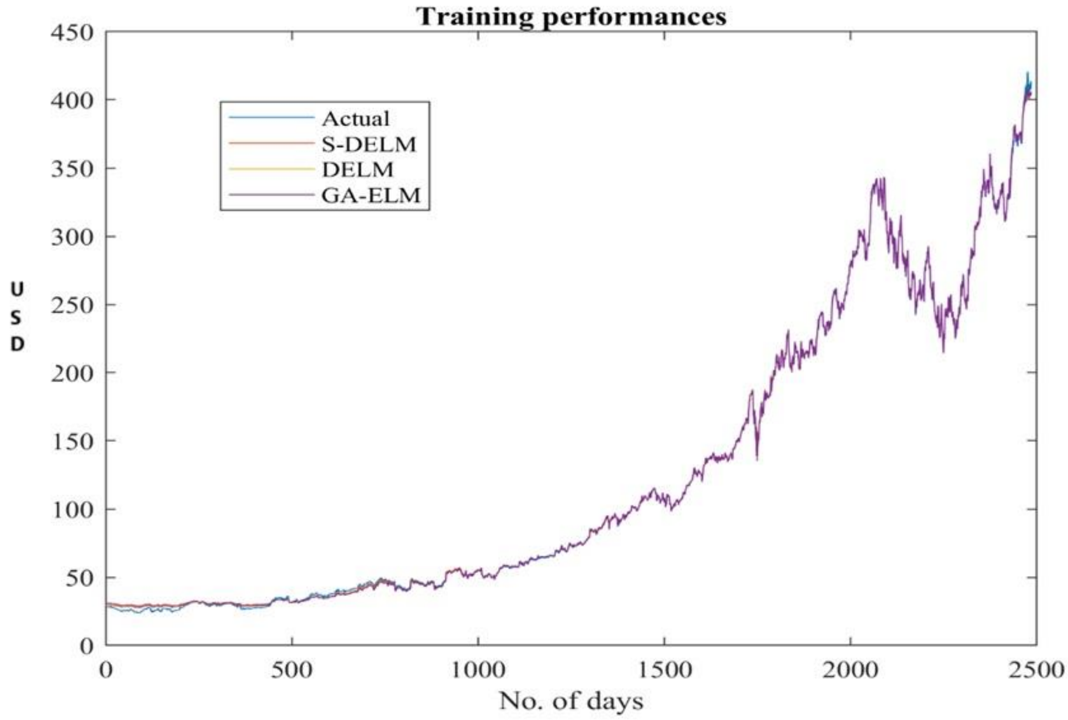


Figure 51: Training performance comparisons for Microsoft dataset

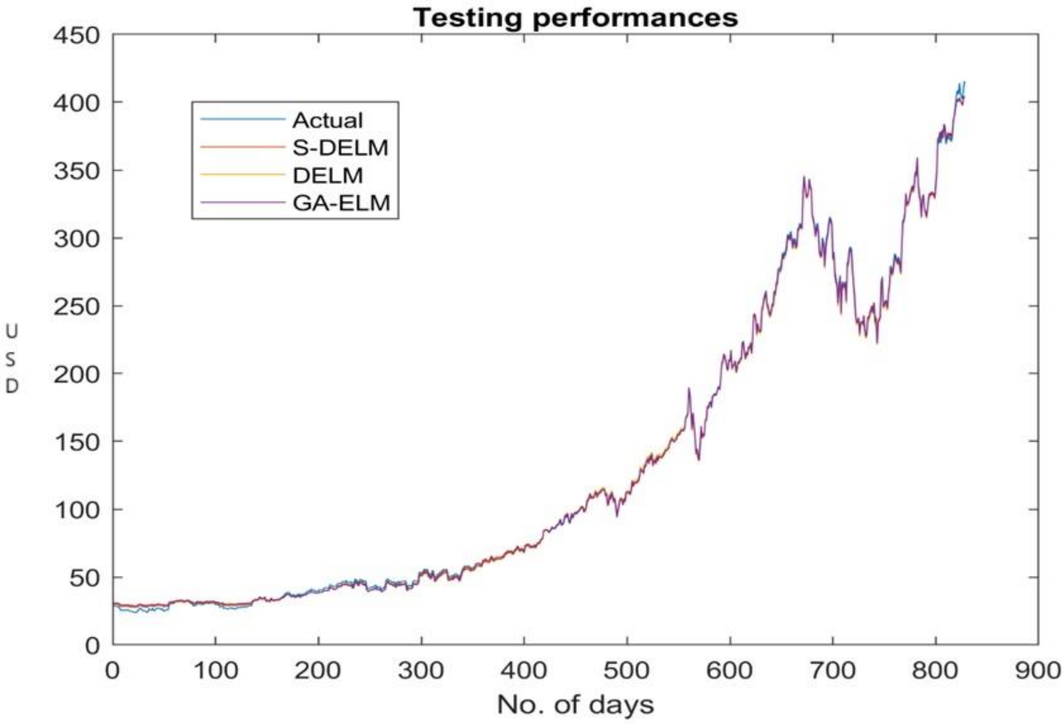


Figure 52: Testing performance comparisons for Microsoft dataset

According to overall experimental results, the proposed prediction method , the S-DELM method produced high performance such as 0.9998 when compared with other prediction methods. Compared with standard DELM-T, DELM-H, GA-ELM, KELM, ELM, SVM, and BPNN, the R^2 of proposed S-DELM is improved by 3.24 %, 4.66 %, 6.93%, 9.47%, 12.29 %, 16.195%, and 19.58% on average, respectively.

In conclusion. the suggested scheme regularly obtains the highest correctness, the lowest computation time, and the closest predictive value using ELM in conjunction with the sentiment index and DWM. Few studies concentrate on the precise price and timeliness of stocks even though many have examined the rise and fall of equities. Accurate and timely stock price forecasting is essential for both investor decision-making and the stability of the national economy. It will make sense to support the steady, sustainable growth of the economy with more timely and appropriate regulation, timely and accurate stock price predictions, and sound stock market guidance.

ELM is often chosen for its fast training speed making it suitable for large-scale datasets and real-time applications. The efficient computation time allows ELM to handle large volumes of data more quickly than some other ML algorithms. Hence. the present research work takes computation time also for performance comparisons. Tables 35 and 36-- and Figures 39 and 40 show the computation time for all compared methods and respective datasets. Figure 41-52 shows the predicted vs actual price comparisons for both train and testing phases. Hence, the developed method produces a low computation time when compared with some variants of ELM with SI and a high computational time when compared with conventional ELM and KELM.

6.10 Computational complexity

The present research work proposes the combinations of the three algorithms such as the DWM algorithm optimizes the Modified ELM algorithm for investment decisions based on sentiment classification using a CNN. Hence, the computational complexity of the proposed model is defined by the combinations of these algorithms. There are three main parts to the model:

a) Sentiment classification is separated from stock market comments using a CNN. CNN method uses text word embeddings and convolutional layers. $\mathrm{O}(\mathrm{L} \times \mathrm{M} \times \mathrm{N} \times \mathrm{F} \times K^2)$. Where, L is the number of layers. $M \times N$ is the input text matrix size (sentence length × embedding dimension). F is the number of filters. $K \times K$ is the kernel size. Pooling and activation functions add negligible complexity $O(P)$.

b) Positive, neutral, and negative sentiments are classified by the Modified Extreme Learning Machine (ELM). ELM has three main steps: Random Weight Initialization ($O(1)$), Hidden Layer Activation Calculation ($O(N \times H)$), and Moore-Penrose Inverse for Output Weights ($O(N \times H^3)$). Where, N is the number of training samples, H is the number of hidden neurons ($H << N$). Hence, the total computation complexity of ELM is $O(N \times H) + O(H^3))$.

c) Deterministic Weight Modification (DWM) Optimization improves ELM's performance. The time of complexity of DWM is defined as $O(H^3) + O(N \times H)$.

d) Hence, the total computation complexity of the proposed S-DELM is determined $\mathrm{O}(\mathrm{L} \times \mathrm{M} \times \mathrm{N} \times \mathrm{F} \times K^2) + O(H^3) + O(N \times H)$.

The computational complexity of the proposed approaches' is Balanced which is efficient for real-time trading decisions. Comparing the suggested CNN + Modified ELM + DWM model to DWM-ELM, GA-ELM, ELM, SVM, and BPNN, the former offers the optimum balance between accuracy and computational efficiency for real-time stock market sentiment analysis. This comparison assesses how computationally complicated different models for sentiment-based investment decisions are. SVM is more sophisticated than ELM-based models, though, and it does not scale well for huge datasets. Multiple epochs are needed for the iterative training of BPNN, which increases its computing cost. Therefore, slower than the suggested model and ELM.

6.11 Compared methods

The performance of the developed stock prediction model is compared with some variants of ELM such as DELM, GAELM, KELM, and ELM and conventional methods such as SVM and BPNN. BPNN can model nonlinear stock market patterns and has good generalization performance. If given sufficient training, it can generate high accuracy and capture intricate relationships between stock history information. However, it's slow training, which needs several iterations to achieve good performance, might cause it to become trapped in local minima that impact predictions, and if not regularized appropriately, it is prone to overfitting.

In terms of ELM, the randomized hidden layer can handle high-dimensional stock data, prevent backpropagation, compute weights directly, and reduce computation and training time. However, choosing hidden nodes carefully is necessary since the number of hidden neurons determines performance. Stability fluctuates and there is no weight fine-tuning because weights are random, which makes it vulnerable to noisy stock data.

The reason KELM is superior to ELM in nonlinear modeling is that it employs kernel tricks to increase accuracy, is more stable than ELM, overcomes instability caused by random weights, and performs well with sizable feature sets in stock markets. Unfortunately, it requires kernel matrix inversion, which increases time complexity, making it computationally expensive. Performance depends on selecting the appropriate kernel function, because kernel operations demand a lot of memory for large datasets. However, in both ELM and KELM, the most important one is the selection of the optimal hyperparameters such as the number of hidden nodes, connection weights and biases, and learning rate which are usually done by trial and error.

Generally speaking, the KELM and ELM require more hidden neurons than more predictable tuning-based learning methods[76]. GA improves performance for noisy data by fine-tuning the network for robustness, avoiding random initialization issues, and optimizing hidden nodes and weights to get more accurate prediction accuracy. However it's expensive to compute, takes longer to optimize, and necessitates changing GA parameters like population size and generations[77]. The DWM approach guarantees a steady and reliable prediction model, the weight changes are deterministic, and unlike GA-ELM, it does not require evolutionary iterations reduces the model's sensitivity to noise in financial data and eliminates needless iterations, in contrast to GA-ELM, which uses several generations.

Traders risk making the mistake of overfitting their trading methods to historical data, which results in tactics that perform well on historical data but are not suitable for upcoming market conditions. Technical indicators offer valuable information about market patterns, price movements, and possible future price directions, which is useful when examining stock market data. Technical indicators are useful for verifying market trends or price changes. Conversely, the developed stock market prediction algorithms ignore other factors that affect stocks and their complex internal workings in favor of primarily using historical data and technical indicators as their input. Because people are inconsistent, the stock market does not always operate according to systematic rules.

Their behavioral, psychological, and emotional characteristics are vital in the economic system. Furthermore, new research has shown that investor attitude may have an important effect on stock market returns. There is a strong hint that investors are not irrational, and as social networks become more important in people's lives, shareholder connections are getting easier and more common. Thus, an investor's perspective and decision-making procedures may be influenced by the feelings and opinions of other investors as well as those shared on social media [63].

Sentiment classification-based stock market investment decisions necessitate highly effective models that can swiftly and precisely interpret enormous volumes of data[78]. The suggested prediction method considers investor sentiment by computing binary sentiment indices for optimistic and negative sentiments. An algorithm for sentiment analysis based on CNN that divides stock market remarks into bullish and negative perspectives. Next, the predicting the stock price using enhanced ELM optimized by DWM. Finally, technical indicators and user comments datasets are considered as inputs to the prediction method.

6.12 Summary

Analyzing stock market prediction results entails assessing predictive model performance, analyzing the results, and drawing practical conclusions. The present research work focused on various performance evaluations conducted for evaluating the performance of developed stock market prediction methods such as training and testing, convergence analysis, and predicted values with actual stock closing prices.

When utilizing sophisticated models, result analysis is crucial to sentiment analysis-based stock market price prediction such as DWM-ELM. In this phase, the model's results are interpreted, the forecasts are verified, and the model's conclusions are checked for accuracy and relevance in financial forecasting. Better decision-making in the financial markets results from this process, which not only helps the model be refined but also increases confidence in its capacity to anticipate stock prices reliably and accurately. Sentiment-based DELM, the focus of the current research effort, is discussed in this section. Utilizing historical datasets as input, a CNN is exploited to generate the SI, which is then used to forecast the stock price using improved DELM that is enhanced by DWM. The three phases of research work were carried out to predict the stock prediction. The first phase is using the optimized DELM using historical datasets. The second phase is using the technical indicators for optimized DELM. Finally, the third phase uses investor comments to predict the stock market price with DELM.