IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Enhancing Stock Market Forecast Accuracy With Investor Sentiment And Advanced Optimized Lstm

[1]K. Kiruthika, [2]E.S. Samundeeswari

[1]Department of Computer Science

[1]Vellalar College for Women, Erode, Tamil Nadu, India

[2]K.S. Rangasamy College of Technology, Tiruchengode, Tamil Nadu, India

Abstract: A key challenge in the field of economics is forecasting the prices of stocks. This is made even more complex due to the unpredictable and turbulent nature of the stock market, making it one of the most challenging areas to accurately predict. Our proposal is to create a model for stock market forecasting that considers the sentiments of investors, employing deep learning to address these complexities. The present proposed work suggests including investors' emotion into stock forecasting, which could greatly enhance the reliability of the model's forecasts. Then, use of long short-term memory (LSTM) because of its benefits for using its memory function to analyses correlations between time series data. The findings of the experiment demonstrate that the optimized LSTM model may decrease time delay in addition to increasing prediction accuracy.

Keywords: Hyperparameter optimization, long-short term memory, salp swarm optimization, sentiment analysis, stock market price prediction

I. INTRODUCTION

A substantial amount of money is brought into the stock market by buying shares, which enhances the organic makeup of commercial funds by promoting resource awareness and greatly boosting the expansion of the commodity economy. The most difficult task is usually predicting stocks because of their volatility and noise. How to continuously predict stock movement is a very unresolved subject in the modern world of social economy and organization. Economic scientists have conducted a number of research in response to investor worries and the allure of large returns. In the past, methods such as generalized autoregressive conditional heteroskedasticity (GARCH), autoregressive moving average (ARMA), autoregressive integrated moving average (ARIMA), and autoregressive conditional heteroscedasticity (ARCH) were utilized to predict stock market information. [1-4].The linearity of the past and present variables is a prerequisite for all these models. The disordered and loud characteristics of financial time series information typically indicates that it lacks definable structure or linearity, which makes it difficult for statistical techniques to predict stock market indices. Nevertheless, a number of unfavourable aspects of the stock market make projections derived from conventional statistical methods inadequate [5].

Time-series analysis in the finance industry has been heavily reliant on machine learning models lately. Artificial neural networks (ANNs) and support vector regression (SVR) both produced notable outcomes. Furthermore, because deep learning is so good at mapping nonlinear relationships and adopting little prior knowledge, it is also emerging as a new machine learning trend. The intricacy of financial time series can be

overcome by deep learning's strong data processing skills. As a result, deep learning and finance have a lot of potential, but there is still more work to be done in this field.

However, because of human unreliability, the stock market does not always follow systematic ideologies. Instead, developed stock market prediction systems generally employ historical data as their input, neglecting other stock-impacting variables and their intricate talk into mechanisms [6]. Their behavioral, psychological, and emotional characteristics are vital in the economic system. Furthermore, new research has shown that investor attitude may have a significant impact on stock market returns. There is a strong hint that investors are not totally irrational, and as social networks become more importance in people's lives, shareholder connections in the stock market are getting easier and more common. Therefore, the sentiment and views expressed by other investors and on social media could influence an investor's mindset and decision-making processes. This could also, to some extent, have an effect on the stock market [7]. The suggested plan considers investor sentiment by computing binary sentiment indices for optimistic and negative sentiments.

We propose the use of LSTM to the forecasting of stock closing prices, which are subject to a wide range of important factors, show considerable degrees of uncertainty, and have nonlinear features. Few studies have utilized deep learning to forecast specific values in financial time series. While some researchers have applied deep learning to the financial sector, much of their work has concentrated on categorization difficulties. Furthermore, because to its advantages in evaluating associations between time-series data via its memory function, we select long short-term memory (LSTM) over other popular deep learning models, such as convolutional neural networks (CNN), deep belief networks, and others. The sentiment index (SI) is considered using a CNN-based classification model. In order to categorize stock market comments into bullish and bearish viewpoints, we present our CNN-based sentiment analysis algorithm in this paper. Next, in order to anticipate stock price, sentiment-based optimized LSTM is suggested in this article. The current study endeavors to decrease forecast inaccuracy, improve precision, and expedite learning. The following research contributions are made and presented:

- Four sets of stock market data are utilized to analyze the effectiveness of the enhanced LSTM model.
- Four unique evaluation tools are considered for assessing the robustness of the SA-ISSA-LSTM method.
- The refined LSTM model, developed under ISSA, is utilized for forecasting stock prices.

II. Related works

J. Shobana et al (2021) [8]created a novel technique for extracting text that uses skip-gram architecture to determine word contextual information and semantic links. Nonetheless, the primary contribution of this work is the LSTM for sentiment analysis based on the Adaptive Particle Swarm Optimization (APSO) algorithm. Presenting the Adaptive PSO algorithm improves weight parameters, which in turn improves the performance of the LSTM. The APSO classifier, which helps the LSTM choose the ideal weight for the environment in fewer iterations, is created when the PSO algorithm and the opposition-based learning (OBL) approach are combined. As a result, the APSO-LSTM's capacity to modify characteristics like learning rates and ideal weights, along with wise hyperparameter selections, improves accuracy and lowers losses.

D. Londhe et al (2022) [9]developed a new hybrid Deep Bidirectional LSTM (SoEo Algorithm-based Deep BiLSTM) to accurately forecast the sentiment. The process of transliteration identifies the languages and transforms them into a uniform format; features are then retrieved from this standardized data. The BiLSTM classifier is used to hybridize and carry out the adaption behavior of coyotes and the hunting strategy of bald eagles in both forward and backward orientations. According to the results of the simulation, the suggested model produced results of 91.572% accuracy, 89.19% precision, 91.551 % recall, and 89.019% F1 measure, all of which are higher than those of state-of-the-art techniques.

S. Wu et al. (2022) [11]suggested S_I_LSTM, a stock price prediction technique that takes into account the mood of investors as well as a number of data sources. First, we get data from various online sources by crawling them and preprocessing it accordingly. Technical indicators, non-traditional data sources like stock posts and financial news, and historical stock data are all included in this set of information. Then, for the non-

traditional data, we employ the convolutional neural network-based sentiment analysis method, which can determine the investors' sentiment index. In the end, we use the long short term memory network to anticipate the China Shanghai A-share market by combining sentiment index, technical indicators, and historical stock transaction data as the feature set of stock price prediction. According to the experiments, the mean absolute error can reach 2.386835, which is better than traditional approaches, and the predicted closing price of the stock is closer to the genuine closing price than the single data source.

Y. Shi et al (2021)[12]suggested a novel deep neural network-based sentiment analysis system for stock comments and used the estimated sentiment data to forecast stock movement. According to the empirical findings, our deep sentiment classification method outperformed the logistic regression algorithm by 9% and produced a sentiment extractor that was precise enough for the subsequent prediction stage. Furthermore, our novel hybrid features—which combine sentiment analysis with stock trading data—achieved a 1.25% improvement across 150 Chinese stocks in the testing sample. The sentiment data might lessen the prediction outcomes for American stocks. It was discovered that emotion traits gleaned from comments work well for Chinese stocks that have a lower beta risk value and a greater price to book value.

P. Koukaras et al (2022)[13]created an algorithm using SA on Twitter and StockTwits data to forecast stock movement. Using Microsoft stock, sentiment and stock movement data were utilized to assess and validate this strategy. Tweets from StockTwits and Twitter were collected, together with financial information from Finance Yahoo. Seven ML classification models were used and SA was applied to tweets. This work's primary innovation lies in its integration of several machine learning and artificial intelligence techniques, with a focus on obtaining additional features from social media, such as public sentiment, to enhance the accuracy of stock predictions. Utilizing SVM and the Valence Aware Dictionary and Sentiment Reasoner (VADER) to analyze tweets produced the best results. The highest Area Under Curve (AUC) value was 67%, and the highest F-score was 76.3%.

Z. Jin et al (2020) [14]presented a deep learning-based stock market forecast model that takes emotional inclinations of investors into account. Initially, it was suggested to incorporate investors' mood into stock prediction, which can significantly increase the accuracy of the model's predictions. Second, it is exceedingly difficult to make an accurate prediction since the stock pricing sequence is a complex temporal sequence with varying scales of fluctuations. The authors suggested using empirical modal decomposition (EMD) to gradually break down the complex sequence of stock price, which improves prediction accuracy. Third, use LSTM since it has the benefit of using its memory function to analyze relationships between time series data. refined it even further by implementing an attention mechanism to concentrate more on the most important data. The findings of the experiment demonstrate that the updated LSTM model can decrease time delay in addition to increasing prediction accuracy. It has been established that investors' emotional tendencies can effectively enhance expected outcomes; the addition of EMD can enhance inventory sequence predictability; and the attention mechanism can assist LSTM in effectively extracting pertinent information and current mission objectives from the information ocean.

Y. Liu et al (2017) [15]developed a technique to evaluate online stock forum sentiment and utilize the data to forecast Chinese market stock volatility. The sentiment of the online financial posts has been classified, and the dataset is now open for public use in study. We construct a mechanism to calculate the sentimental score of each internet post about a given stock by creating a sentimental vocabulary based on financial terminology. Recurrent Neural Networks (RNNs) are utilized to fuse market data with two sentiment indicators, which contain emotive information, in order to anticipate stock volatility. An empirical investigation demonstrates that the model performs noticeably better with emotive indicators when compared to utilizing RNN alone.

L. Nemes et al (2021)[16] discuss the subject of stock value fluctuations and forecasts utilizing recently scraped business-related economic news. authors that concentrate on business news headlines. To analyze the sentiment of the headlines, they employ a wide range of methods. Using BERT as the baseline, compare the outcomes with those of three other tools: VADER, TextBlob, and a Recurrent Neural Network. Additionally, compare the sentiment results with the changes in stock prices over the same time period. The BERT and RNN,

in contrast to the other two instruments, were significantly more accurate; they could identify the emotional values without the need for neutral parts.By contrasting these findings with the movement of stock market prices over the same time periods, sentiment analysis of economic news headlines can be used to determine the exact moment at which a shift in stock values happened. Additionally, a noteworthy distinction was found among the models about the impact of affective values on the fluctuation of the stock market's value as determined by the correlation matrices.

W. Khan et al (2020)[17]proposed a list of ten machine learning models that are utilized on the concluding datasets to forecast the direction of the stock market. The trial outcomes reveal that the sentiment aspect enhances the forecasting precision of machine learning models by 0–3%, and the political scenario aspect augments the accuracy by approximately 20%. Moreover, the sentiment feature performs optimally on the seventh day, whereas the political scenario feature peaks on the fifth day. The Support Vector Machine (SVM) model demonstrates the highest efficacy, whereas the Adaptive Subset Classifier (ASC) and Bagging methods exhibit subpar results. The findings on interdependence suggest that stock markets within the same sector exhibit a moderate positive correlation with one another.

III.Long-Short Term Memory

LSTM stands for Long Short-Term Memory; a unique type of recurrent neural network (RNN) that excels at recognizing patterns over time, despite its own set of constraints compared to other RNNs. LSTM transforms input words from sentences into a distributed representation that can be used to map each word in a dictionary to a continuous value in a multi-dimensional space. Each term w in dictionary W is inserted into n-dimensional space $\in R^{nX|W|}$. Typically, a LSTM network contains a cell state C_i and hidden state h_i and it also contains of each three multiplicative units: forget F_i unit, input I_i unit and output O_i unit with weights W_F,W_i,W_o and bias B_F,B_i, B_o respectively. These units help the LSTM memory cell perform various operations such as reading, writing, resetting, and enable the memory cell to access and retain information over time. "σ"is Sigmoid function employed in input, forget, and output layers for producing values between 0 and 1. The following equations denote a LSTM memory cell which can be represented as:

$$I_i = \sigma(W_i|X_i, h_i| + B_I) \qquad (1)$$

Input gate's function $'I_i$ generates new memory state if the importance of the new word is significant. This decision is made by the input gate, which evaluates the value of retaining the new word, leading to the formation of a new memory.

$$F_i = \sigma(W_F|X_i, h_{i-1}| + B_F) \qquad (2)$$

Forget gate $'F_i'$acts as the entry point but it decides whether the previous memory cell is relevant for the computation of the present memory cell or not. The forget gate operates on the incoming word and the previously concealed state from the past, generatingF_i

$$\widetilde{C_i} = \tan h\,(W_c[X_i, h_{i-1}] + B_c) \qquad (3)$$

where $'\widetilde{C_i}'$is a fresh memory that relies on elements of a novel word $'x_i'$ and past concealed state $'h_{i-1}'$.

$$C_i = F_I \times C_{i-1} + \mathrm{I_i} \times \widetilde{C_i} \qquad (4)$$

Following the result of the forget gate$'\mathrm{F_i}'$it omits past memory $'C_{i-1}'$from this phase. It is also considering the result of the input gate$\mathrm{I_i}$ and new memory$'\widetilde{C_i}'$. Next, the model adds these two outcomes to create the ultimate memory$'C_i'$.

$$\mathrm{O_i} = \sigma(\mathrm{W_o}[\mathrm{X_i}, \mathrm{h_{i-1}}] + \mathrm{B_o}) \qquad (5)$$

$$\mathrm{h_i} = \mathrm{O_i} \times \tan \mathrm{h}\,(\mathrm{C_i}) \qquad (6)$$

Output gate $'O_i'$determines the timing for transferring the value held in the memory cell to the concealed layer.$'h_i'$is a newly discovered latent state calculated by multiplying the output state and the new cell state at each point.

IV. Salp swarm optimization (SSA)

Salp Swarm Optimization is a natural optimization technique that draws inspiration from the swarming behaviour of marine crustaceans known as salps. Like previous swarm intelligence algorithms, the salps' movement inside the algorithm is determined by the collective experience of the individuals as well as the global optimal solution discovered by the swarm. Converging towards optimal solutions while effectively exploring the solution space is the aim. The position of salps is defined in an *n*-dimensional search space where *n* is the number of variables of a given problem. Therefore, the position of all salpsare stored in a two-dimensional matrix called *x*. it is also as summed that there is a food source called *F* in the search space as the swarm's target. To update the position of the leader the following equation is proposed:

$$X_j^1 = \begin{cases} F_j + c_1((ub_j - lb_j)c_2 + lb_j) & c_3 \geq 0 \\ F_j + c_1((ub_j - lb_j)c_2 + lb_j) & c_3 < 0 \end{cases} \qquad (7)$$

Where X_j^1 shows the position of the first salp (leader) in the *j*th dimension. F_j is the position of the food source in the *j*thdimension,ub_j indicates the upper bound of *j*th dimension, lb_jindicates the lower bound of *j*th dimension, c_1,c_2, and c_3 are random numbers. The leader only updates its position with respect to the food source. The coefficient $c_1$is the most important parameter in SSA because it balances exploration and exploitation defined as follows:

$$c_1 = 2e^{-\left(\frac{4t}{L}\right)^2} \qquad (8)$$

Where l represents the present iteration and L stands for the highest possible number of iteration.

The parameter c_2 and $c_3$are unpredictable numbers evenly distributed across the range of [0, 1]. Indeed, they determine whether the subsequent spot in the jth dimension should move towards positive infinity or negative infinity, along with the increment size.

To update the position of the followers, the following equations is utilized (newton's law of motion):

$$X_j^i = \frac{1}{2}at^2 + v_0 t \qquad (9)$$

Were $i \geq 2, X_j^i$ shows the position of i_{th} follower salp in *j*th dimension, *t* is time, $v_0$is the initial speed, and $a = \frac{v_{final}}{v_0}$ where $= \frac{x - x_0}{t}$.

Because the time in optimization is iteration, the discrepancy between iterations is equal to 1, and considering$v_0 = 0$, this equation can be expressed as follows:

$$X_j^i = \frac{1}{2}\left(X_j^i + X_j^{i-1}\right) \qquad (10)$$

Where $i \geq 2$ and X_j^ishows the position of *i*th follower salp in *j*th dimension.

V. CONVOLUTIONAL NEURAL NETWORK (CNN)

The sentiment analysis module in this study uses the CNN model presented by Kim to categorize comments [19]. The CNN is separated into three parts such as input, convolution, and classification layer [20]. An $r \times u$ is the text word vector matrix that works as the input layer, where r is the number of distinctive phrases for each text and u is a result of data processing. To convolve the word vector matrix, the convolutional layer initially passes over the convolution kernel w of length h is specifically:

$$t_i = f(w * s_{i,i+h-1} + b) \qquad (11)$$

$s_{i,i+h-1}$ is a continuous text section made up of phrases i^{th} through $i+1$ phrase. $*$ is the convolution operation. f and b is the nonlinear function and bias term. Then, the training speed is normalized using the batch normalization (BN) algorithm. To lower the dimensionality and maintain a constant number of features, maximum value pooling is performed. The classification layer uses the BN algorithm to prevent changes in data distribution and the softmax layer to determine the classification probability. The probability is used to categorize stock market comments and is determined as follows:

$$P_j = P(y = j \mid X, b) = \frac{e^{X^T W_j + b_j}}{\sum_{i=1}^{L} e^{X^T W_j + b_j}} \qquad (12)$$

P_j - denote 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.

5.1 ISSA based on POBL

The proposed ISSA method has two steps such as (i) opposition-based initialization, and (ii) generation jumping. These steps are discussed as follows,

a) Opposition-based initialization

The position of each leader salp (j^{th} element of the first salp) is initialized as follows,

$$X_j^1(t)|_{(t=0)} = X_j^{min} + \left(X_j^{max} - X_j^{min}\right).r_{ij}^u(t)|_{(t=0)} \qquad (13)$$

Similarly, j^{th}element of i^{th} follower position X_j^i is initialized as follows,

$$X_j^i(t)|_{(t=0)} = X_j^{min} + \left(X_j^{max} - X_j^{min}\right).r_{ij}^u(t)|_{(t=0)} \qquad (14)$$

$r_{ij}^u(t)|_{(t=0)}$ - stands for the evenly distributed random number between 0 and 1. After initialization of leader position $X_j^1(t)|_{(t=0)}$ and follower position $X_j^i(t)|_{(t=0)}$ of *i*th salp, the opposite of positions is calculated. The original swarm and its opposite swarm are then used to choose the best NP number of places with speeds.

b) Generation jumping

The *j*th element $x'_{ij}(t)$ of opposite position $x'_i(t)$ of i^{th} salp is described in the search space as follows:

$$x'_{ij}(t) = a_j(t) + b_j(t) - \alpha_{ij}(t).x_{ij}(t) \qquad (15)$$

Where $[a_j(t), b_j(t)]$ is the dynamic search space series whichare derived as follows:

$$a_j(t) = \left\{\min_{\forall_i} x_{ij}(t)\right\} \qquad (16)$$

$$a_j(t) = \left\{\max_{\forall_i} x_{ij}(t)\right\} \qquad (17)$$

$\alpha_{ij}(t)$ is the dynamic tightening feature for i^{th} salp in j^{th}dimension which is used to increase the convergence rate and efficiently escape from local minima leading to enhancing the ability of global searching. $\alpha_{ij}$is defined as follows:

$$\alpha_{ij}(t) = 1 - \eta.r_{ij}^c(t) \qquad (18)$$

Where $r_{ij}^{c}(t)$ is the Cauchy factor that distributes a random number centered at the source with a scale parameter one. The Cauchy density function centered at the origin is defined by

$$f(x) = \frac{1}{\pi}.\frac{s}{s^2+x^2}, \quad -\infty < x < \infty \qquad (19)$$

Where S - represents the scale parameter. The Cauchy distributed function is well-defined as follows:

$$F_s(x) = \frac{1}{2} + \left(\frac{1}{\pi}\right) arctan\left(\frac{x}{s}\right) \qquad (20)$$

$\eta = \beta \times \eta$. The initial value of η is 1.0 and β is aevenly spread out random number that spans between 0.01 and 0.9. Three partial opposition positions of various orders are generated after the computation of the opposing position i^{th}salp. From the original, opposite, and partially opposite positions, the best NP number of solutions is computed as follows:

$$k = \lceil rand() \times (D-1) \rceil \qquad (21)$$

Where $1 < k < D$ since $k = 0$ represents the complete opposite and $k = D$ represents the original position. $rand()$ is a uniformly distributed random value that ranges between 0 and 1. For k times, the indices $l(1 \leq l \leq D)$ of the original values $x_{il}(t)$ in the partial opposite positions are considered as follows:

$$l = \lceil rand() \times D \rceil \qquad (22)$$

The i^{th} element of the opposite vector the $x_{il}(t)$ of the original vector $x_i(t)$ replaces the $\acute{x}_i(t)$ to derive the partial opposite vector $p\acute{\chi}_i^k(t)$ as follows:

$$\acute{x}_{il}(t) = x_{il}(t) \qquad (23)$$

6. PROPOSED SENTIMENT ANALYSIS based on ISSA-LSTM

The proposed research work has two contributions. First, the SI is calculated using a CNN-based classification model. Second, the optimized LSTM is used ISSA for predicting the stock price. An initial goal is to perform a group that incorporates user sentiment opinions for the historical data about the stock as one element in predicting the price of the company. Based on the amount of daily bullish and bearish made by several users, the SI is created. Consequently, to determine the group sentiment tendency and calculate the SI, we must first recognize the proper sentiment categorization of the individual stock review. Models called Word2vec can un-supervised learn semantic information from a lot of text. Words must be mapped from the old space to the new space for word2vec to perform. In particular, by learning the text, the word vector is created to represent the semantic information of each word, and the semantically related words are assigned to similar distances. In this study, conduct softmax normalization and compute the cosine similarity using the Skip-gram in word2vec.First, word2vec is developed to learn high-dimensional vector representations of phrases from large-scale stock comment corpora. The outcome is immediately applied if the phrase in the stock comments is to be categorized. If not, word2vec initializes it randomly. Following that, CNN will be given the word vectors, which denote the preprocessed text. Next, using a CNN enhanced by word2vec, we generate the SI for the group's sentiment analysis. Based on the total number of positive and negative comments made each day, the SI for the day is determined. The group sentiment analysis, which might indicate the investor's general emotional inclination, can show this. To determine the daily SI for the stock, we employ the approach suggested by Antweiler and Frank [21].

$$BI_t = \mathrm{I}\,n \frac{1+M_t^{bullish}}{1+M_t^{bearish}} \qquad (24)$$

where BI_t is the SI at time t and $M_t^{bullish}$ and $1+M_t^{bearish}$, which are determined by the number of bullish and bearish, respectively, are the weights of bullish and bearish stocks. The index considers how bullish and bearish remarks have affected investor sentiment over time, and changes in the index's direction are proportionate to the weights given to bullish and bearish stock valuation views. The index BI_t is positive and the general tendency is characterized as bullish when more comments reflectbullishly.

On the other hand, if more comments are expressing bearish sentiment than positive sentiment, the SI BI_t is negative and the overall mood is pessimistic. The size of the SI reveals the degree of inclination to a particular category, and the positive or negative. BI_t reflects the group of sentimental orientation. Finally, the SI is considered as input to the prediction with their historical prices for predicting stock prices of the stock market using optimized LSTM.

7. Proposed SA based optimized LSTM

The ISSA model is employed in this study to optimize the LSTM approach, which forecasts the price of stock indices. Figure 1 shows the flowchart of proposed ISSA-LSTM approach and algorithm 2 shows the step-by-step process of proposed ISSA-LSTM. The number of partial opposite locations for each salp in the ISSA algorithm is chosen by trial and error. Additionally, each salp's half-opposing locations' degrees or orders are chosen at random. By adding a control mechanism for choosing the number and degree of partial opposite positions that promote effective exploration and exploitation of salp in the search space, the performance of the ISSA algorithm may be significantly enhanced. The initial stage of the ISSA-LSTM model is data preparation, which comprises dividing and normalizing data. The ISSA method is employed in the second step to look for the LSTM network's optimal parameters. The stock market price outcomes are finally predicted using the LSTM network with the best parameters.

The core of our study is a hybrid mechanism that combines an RNN-LSTM network with an optimization method like ISSA. This mechanism overcomes processing limitations caused by a heuristic estimate of architectural hyperparameters that is dependent on trial and error. This technique assists us in formulating a methodical approach for creating an ideal deep-learning model through the automated construction of an optimized LSTM network. To create an accurate model for intraday stock market forecasting, we are investigating the five most impactful hyper-parameters of this network. Time lag, number of hidden layers, number of hidden neurons, batch size, and epochs are the first five hyper-parameters on the list.

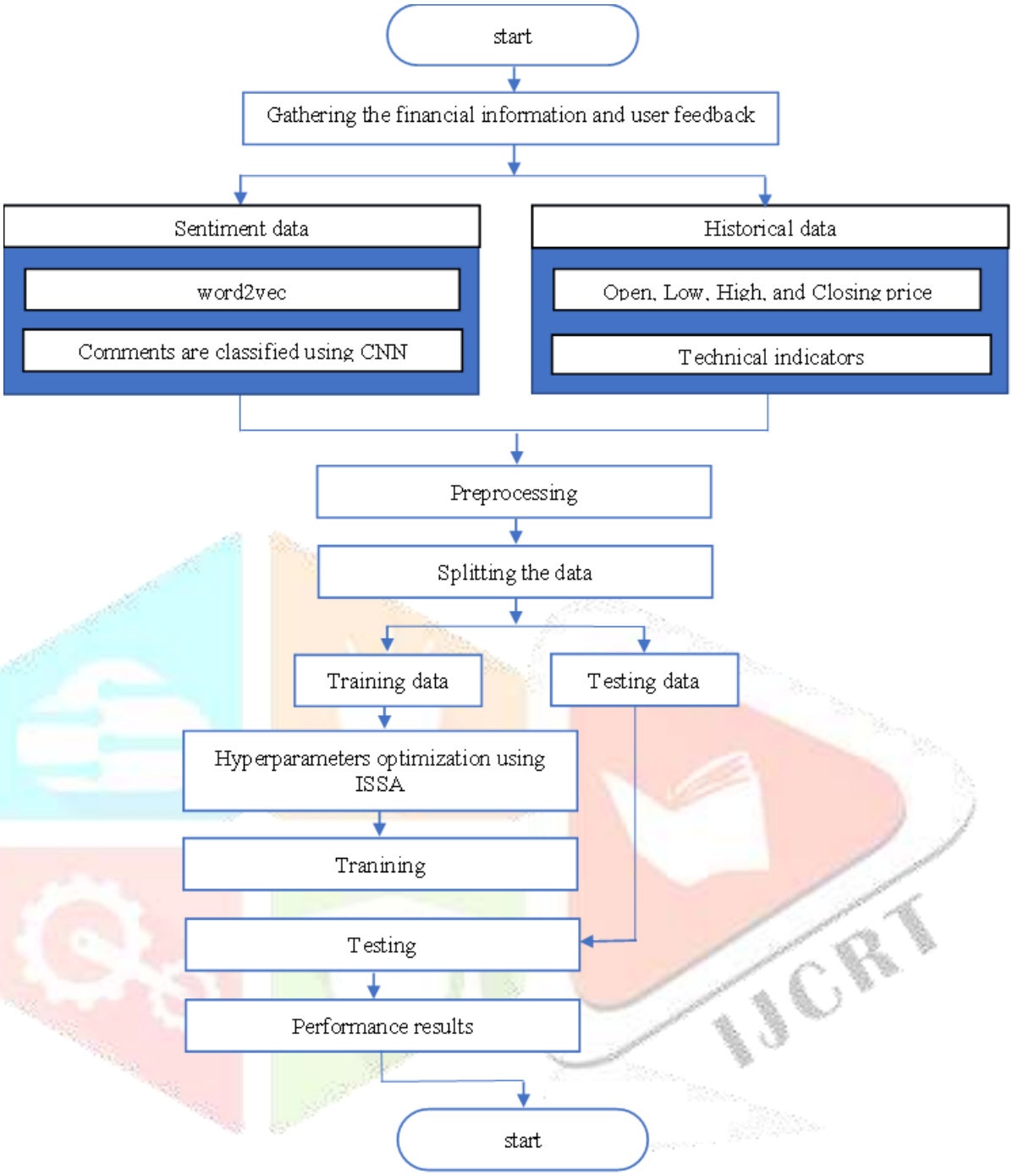

Figure 1 : Overview of proposed optimized LSTM

- ***Time lag:*** the model may capture conditional relationships across subsequent periods by rejecting the obsolete data with the proper time lag (time steps). A dimensionality problem is exacerbated, the model is overfitting, and there are numerous lagged data.
- ***Several hidden layers:*** A topology with either too few or too many numbers of layers cause the model to overfit, learns from the training data, and then fails to generalize fresh, untested data.
- ***Number of hidden neurons:*** A network with a small number of neurons struggles to find the signal in a challenging dataset, underfitting the model. The limited amount of data in the training set prevents all of the hidden layer neurons from being trained, which leads to the model overfitting. Additionally, a network with too many neurons increases the information processing capacity, and having enough training data lengthens the training time to the point where it is impossible to train the network.
- ***Batch size:*** When the gradient descent stochasticity of a network is too high or too low, it has a detrimental impact on the forecasting model during training.

- ***Number of epochs:*** The drawback of a topology with too many epochs is that it does not permit early stopping, which leads to overfitting of the model to the training data and aids in data memorization rather than learning.

8. Experimental results and analysis

The main contributions made in this paper—sentiment index and LSTM optimized with ISSA, for example—will be examined and validated in this part. According to experimental data, the predictor with the highest MAPE and R2 and the lowest RMSE and MAE is thought to perform the best. MATLAB R2015b is used to implement the analysis findings. The SA-ISSA-LSTM, ISSA-LSTM, SSA-LSTM, IPSO-LSTM, PSO-LSTM, GA-LSTM, and LSTM, as well as the performance of the SA-ISSA-LSTM, are correlated with each other across four distinct datasets: the S&P Sensex, Nifty 50, SBIN, and ICICI bank datasets. Four distinct performance metrics are taken into account while analyzing the proposed approaches' effectiveness. The following subsections are discussed in detail.

8.1 Datasets

The historical data and the comment dataset, which computes the SI, are two distinct experimental datasets collected. The stock comment dataset includes comments for model training and a final remark that has to be categorized to find the SI. First, the model was trained using shareholder comments posted on Stock Twits "(https://stocktwits.com/)".It is up to investors to label a comment as optimistic or bearish, even if Stock Twits is a well-known public depositor site. This allows for the collection of a large number of incredibly accurate remarks.

Table 1 : Technical Indicators

S.No	Name of Technical Indicators	Formulas	Descriptions
1	Simple Moving Average (SMV)	$MV = \frac{x_1 + x_2 + \cdots + x_n}{n}$	The average value of specific $'t'$ days
2	10-days Moving Average	$MV_{10} = \frac{x_1 + x_2 + \cdots + x_n}{n}$	The average value of the last 10 transaction days
3	Momentum	$M = C_t - C_{t-4}$	It is measuring the sum of prices over given period length
4	Stochastic (K%)	$STCK = \frac{C_t - LL_{t-n}}{HH_{t-n} - LL_{t-n}} \times 100$	Stochastic determining the velocity of price variation.The comparative position of the current closing price in a certain period is calculated
5	Stochastic (D%)	$STCD = \frac{\sum_{i=0}^{n-1} K_{t-1\%}}{n}$	Specify the three days moving average
6	Relative Strength Index (RSI)	$= 100 - \frac{100}{1 + (\sum_{t=0}^{n-1} \mathrm{UP}_{t-1}/\mathrm{n})/(\sum_{t=0}^{n-1} \mathrm{DW}_{t=1}/\mathrm{n})}$	It measures the speed and movement of the price which ranges between 0 to 100.
7	Williams (%R)	$LW = \frac{H_n - C_t}{H_n - L_n} \times 100$	It computes overbought and oversold levels and is used to control market entry and exit opportunities.
8	Moving Average Convergence Divergence(MACD)	$= MACD(n)_{t-1} + \frac{2}{n+1}\ (Diff_t - MACD(n)_{t-1})$	MACD is to match up to the short-term and long-term momentum of a stock to calculate approximately its future direction
9	Commodity Channel Index(CCI)	$CCI = \frac{M_t - SM_t}{0.015 D_t} \times 100$	It assesses the present price level relative to an average price level over a certain length of time to determine a new movement or warn of severe conditions.
10	Price Oscillator (PO)	$\mathrm{PO} = \frac{\mathrm{MA}_5 - \mathrm{MA}_{10}}{\mathrm{MA}_5}$	PO is showing the relationship between two moving averages.

C_t- Closing price
H_t- High price
HH_t-Highest high price
L_t- Lowest price
LL_t- Lowest Low
UP_t- Upward price
-DW_t Downward price

8.2 Performance measures

Stock market price prediction is a difficult undertaking that requires a variety of performance metrics to assess the precision and potency of prediction models. The performance is measured with four methods namely Root Mean Square Error (RMSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and R-Square (R^2) which are determined as follows,

- **RMSE:** RMSE is the MSE squared, which returns the error measure to the initial data scale which is defined as follows,

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \bar{y}_i)^2} \qquad (23)$$

- **MAE:** MAE determines the average absolute difference between the actual and anticipated values, providing a sense of the size of the forecast mistake which is defined as follows,

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|(y_i - \bar{y}_i)| \qquad (26)$$

- **MAPE:**MAPE simplifies interpretation and comparison between datasets by expressing the prediction error as a percentage of the actual values which is defined as follows,

$$MAPE = \frac{100}{m}\sum_{i=1}^{m}\left|\frac{(y_i - \bar{y}_i)}{y_i}\right| \qquad (27)$$

- **R-Square:**R^2 measures how well the actual values' variability is explained by the projected values. Perfect prediction is denoted by an R^2 of 1, whereas no predictive capacity is indicated by an R^2 of 0 which is defined as follows,

$$R^2 = 1 - \frac{\left(\frac{1}{N}\sum_{i=1}^{N}(y_i - \bar{y}_i)^2\right)/N}{\left(\frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - \bar{y}_i)^2\right)/N} \qquad (28)$$

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

8.3 Results analysis

Sentiment-based optimized LSTM the focus of the present research effort, is discussed in this section. Utilizing historical datasets as input, a CNN-based classification algorithm is used to produce the sentiment index value. This method is then used to forecast the stock price using optimized LSTM, which is further augmented by ISSA. This section discusses the analysis of the experimental data. Tables 2, 3, 4, and 5 present the findings. Figures 2,3,4and 5 provide the graphical representation for the S&P Sensex, Nifty 50, ICICI, and SBIN bank datasets, respectively. When compared to prediction techniques found in literature, the findings show that the SA-ISSA-LSTM approach provided a high performance for all the datasets.

Table 2: Performance results for the Nifty 50 datasets

Approaches	RMSE	MAE	MAPE	R2
SA-ISSA-LSTM	0.0071	0.0273	8.2273	0.9983
ISSA-LSTM	0.0084	0.0388	7.9327	0.9842
SSA-LSTM	0.0091	0.0424	7.3724	0.9798
IPSO-LSTM	0.0117	0.0577	6.8189	0.9541
PSO-LSTM	0.0132	0.0664	5.7256	0.9396
GA-LSTM	0.0144	0.0753	4.6495	0.9007
LSTM	0.0179	0.0091	3.0221	0.8691
BPNN	0.0188	0.0097	2.8786	0.7925

Table 3 : Performance results for the S & P Sensex datasets

Approaches	RMSE	MAE	MAPE	R2
SA-ISSA-LSTM	0.0305	0.2734	8.5612	0.9974
ISSA-LSTM	0.0398	0.3021	7.9469	0.9841
SSA-LSTM	0.0479	0.3259	7.3412	0.9635
IPSO-LSTM	0.0561	0.0387	6.5737	0.9526
PSO-LSTM	0.0734	0.0462	4.6577	0.9338
GA-LSTM	0.0810	0.0569	3.6449	0.8931
LSTM	0.0989	0.0732	2.1251	0.8378
BPNN	0.0950	0.0839	1.4978	0.8279

Table 4 : Performance results for the SBIN datasets

Approaches	RMSE	MAE	MAPE	R2
SA-ISSA-LSTM	0.0067	0.0045	8.9510	0.9974
ISSA-LSTM	0.0079	0.0051	8.1209	0.9837
SSA-LSTM	0.0088	0.0059	7.5508	0.9651
IPSO-LSTM	0.0118	0.0068	6.9431	0.9321
PSO-LSTM	0.0187	0.0075	6.2676	0.9158
GA-LSTM	0.0210	0.0080	5.8558	0.8829
LSTM	0.0271	0.0069	4.8671	0.8420
BPNN	0.0345	0.0086	3.2468	0.7932

Table 5 : Performance results for the ICICI datasets

Approaches	RMSE	MAE	MAPE	R2
SA-ISSA-LSTM	0.0085	0.0024	5.1832	0.9852
ISSA-LSTM	0.0097	0.0033	5.3819	0.9696
SSA-LSTM	0.0129	0.0048	5.2151	0.9458
IPSO-LSTM	0.0291	0.0059	4.9165	0.8822
PSO-LSTM	0.0292	0.0068	4.1718	0.8360
GA-LSTM	0.0371	0.0079	3.7382	0.8015
LSTM	0.0422	0.0094	3.1625	0.7448
BPNN	0.0495	0.0111	2.9349	0.7200

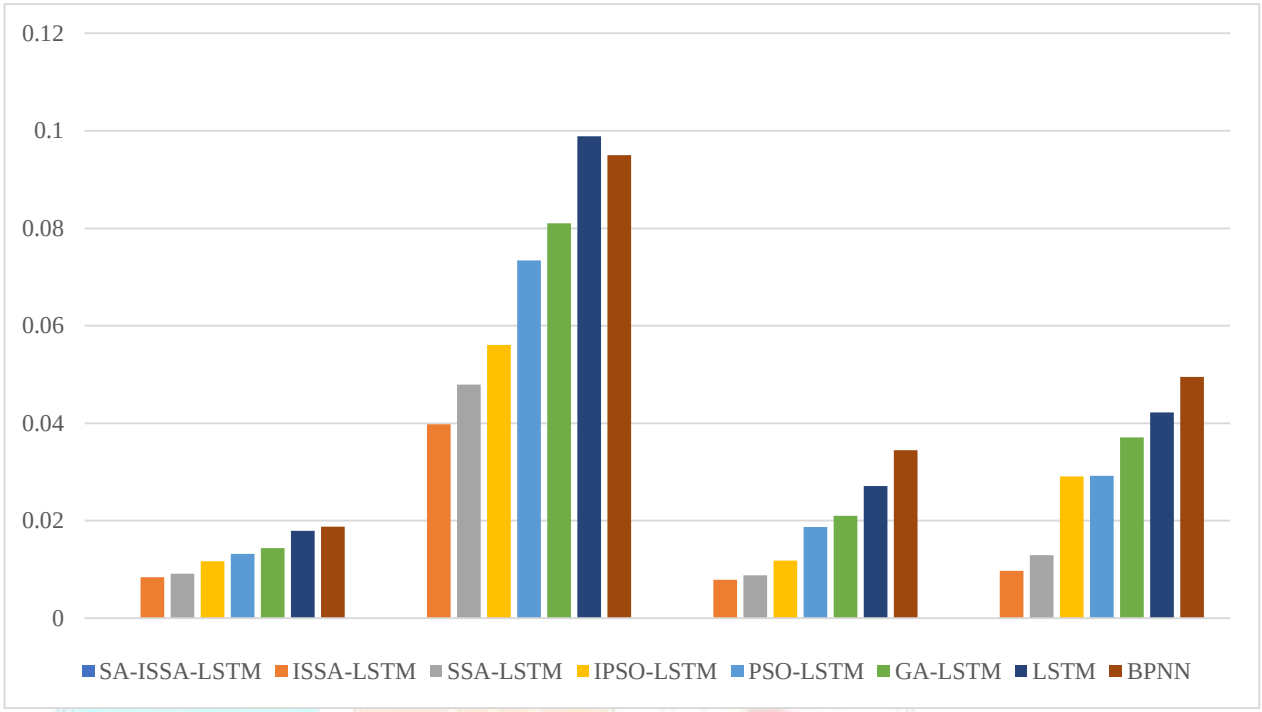

Figure 2 : Performance analysis of prediction model based on RMSE

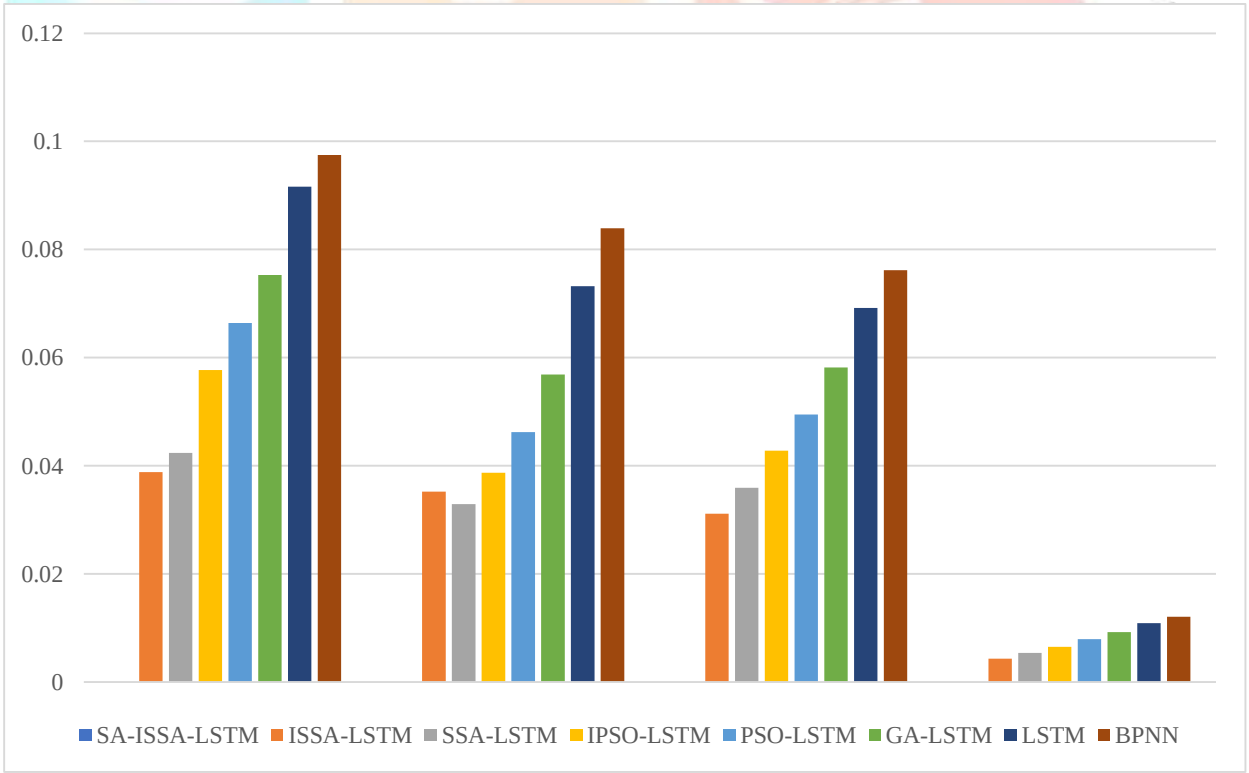

Figure 3 : Performance analysis of prediction model based on MAPE

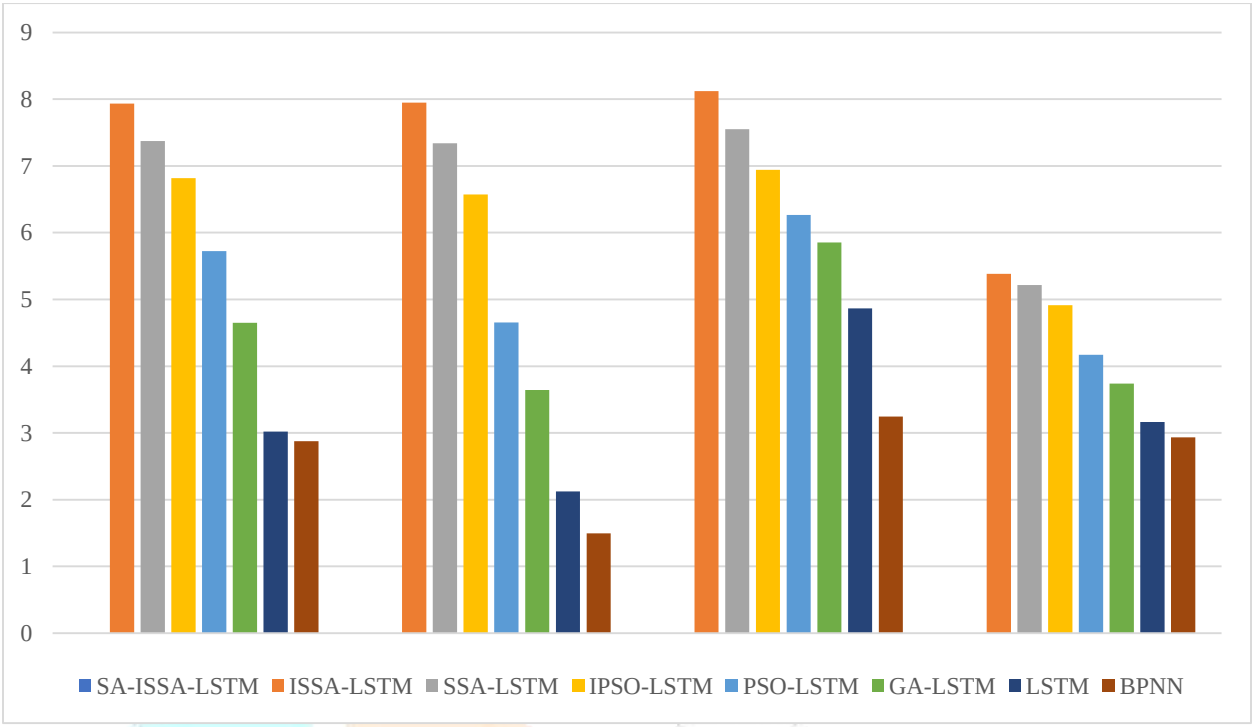

Figure 4 : Performance analysis of prediction model based on MAE

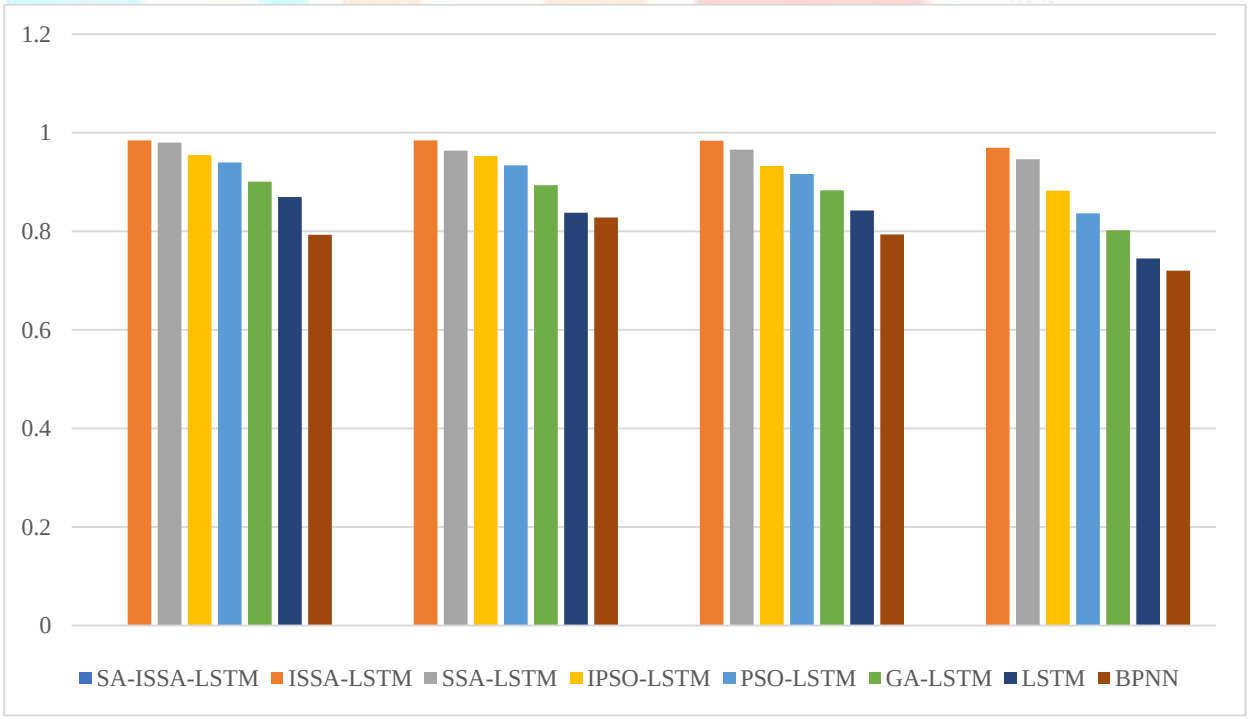

Figure 5 : Performance analysis of prediction model based on R2

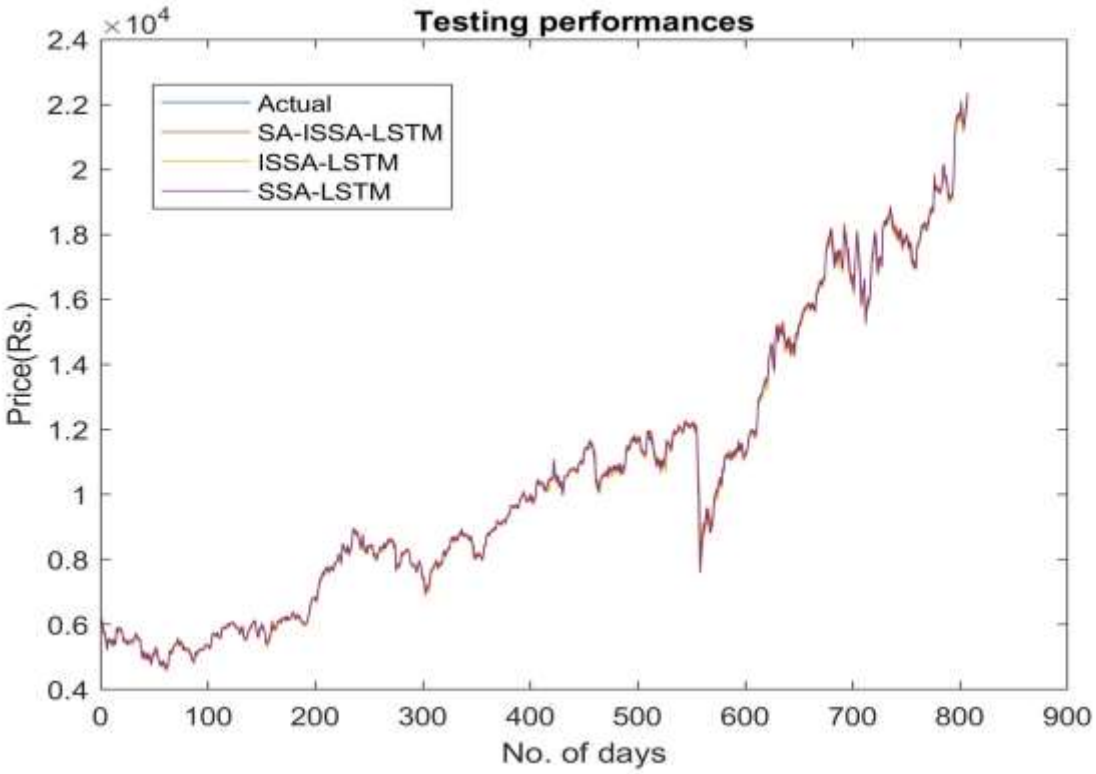

Figure 6 : The prediction vs actual results for Nifty 50 dataset

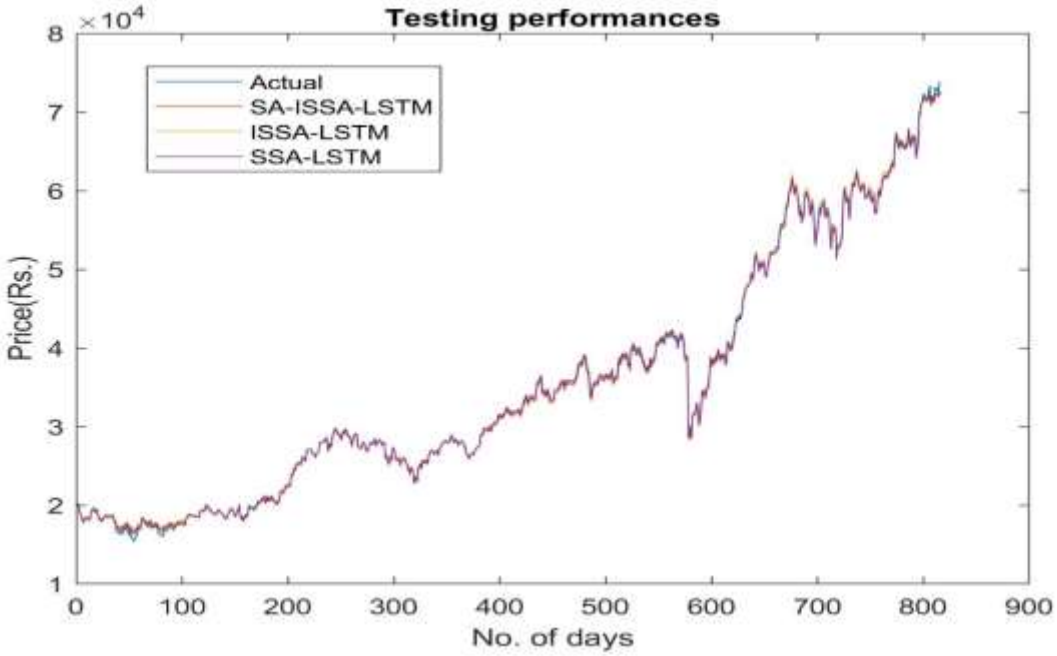

Figure 7 : The prediction vs actual results for S & P BSE dataset

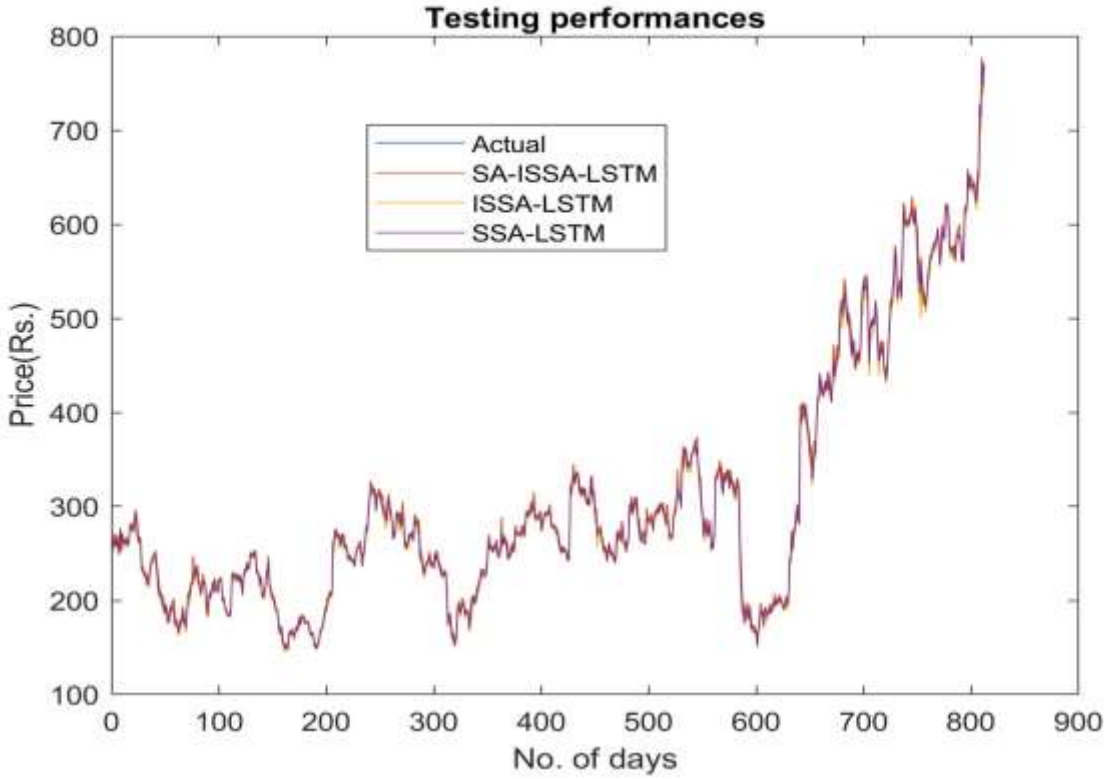

Figure 8 : The prediction vs actual results for SBIN dataset

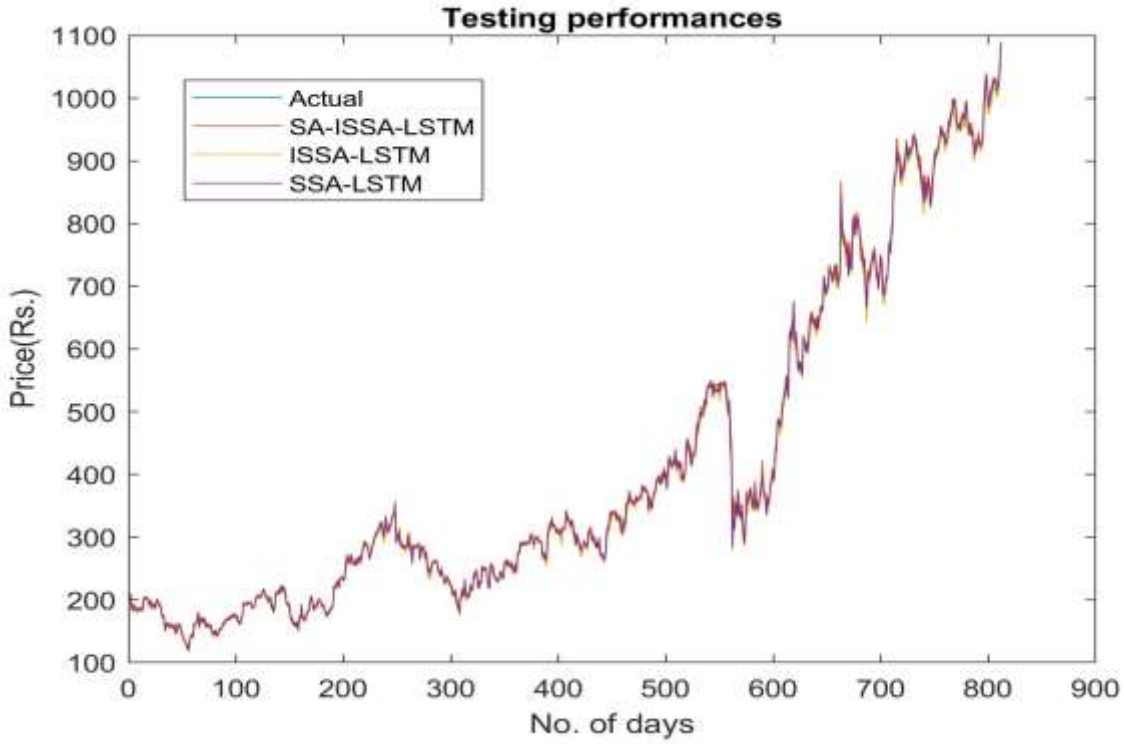

Figure 9 : The prediction vs actual results for ICICI dataset

The actual price and the expected price are important factors in stock market price prediction as they influence decision-making processes and help assess how well predictive algorithm's function. Figures6, 7,8 and9 are shows the performance analysis for actual versus and predicted values based on compared method for Nifty 50, S & P BSE, SBIN, and ICICI datasets, respectively.

To sum up, the proposed method consistently achieves the highest level of precision, the least amount of time lag, and the most accurate predictive value when predicting the stock market by using a mix of the best long short-term memory and sentiment index. Research on the increase and decrease of stock prices has been thoroughly examined; however, the exact cost and timing of stock transactions have been given less focus. The selection of investors and the steadiness of the country's economic situation depend heavily on timely and accurate stock price forecasts. More timely and accurate stock price predictions, timely and appropriate regulations, and sound stock market guidance may all be supplied, all of which make sense in supporting the economy's steady, sustainable growth.

VI. Conclusions

This article introduces an innovative framework for forecasting stock market trends through the use of long short-term memory. Specifically, the sentiment index is utilized to account for the emotional inclination of the investor. In addition, ISSA optimizes the LSTM-based model by determining the proper hyperparameters. Experiments carried out on the four stock datasets have confirmed the suggested scheme's performance. The experimental findings demonstrate that the suggested system regularly outperforms the alternative methods in three important aspects: reduced time offset, greater rise classification accuracy, and closer projected closing price.

References

[1] R. Efendi, N. Arbaiy, and M. M. Deris, "A new procedure in stock market forecasting based on fuzzy random auto-regression time series model," Information Sciences, vol. 441, pp. 113-132, 2018.

[2] L.-Y. Wei, C.-H. Cheng, and H.-H. Wu, "A hybrid ANFIS based on n-period moving average model to forecast TAIEX stock," *Applied Soft Computing*, vol. 19, pp. 86-92, 2014.

[3] Y.-S. Lee and L.-I. Tong, "Forecasting time series using a methodology based on autoregressive integrated moving average and genetic programming," *Knowledge-Based Systems*, vol. 24, no. 1, pp. 66-72, 2011.

[4] C.-F. Chen, W.-H. Ho, H.-Y. Chou, S.-M. Yang, I.-T. Chen, and H.-Y. Shi, "Long-term prediction of emergency department revenue and visitor volume using autoregressive integrated moving average model," *Computational and mathematical methods in medicine*, vol. 2011, 2011.

[5] R. Dash, P. K. Dash, and R. Bisoi, "A self adaptive differential harmony search based optimized extreme learning machine for financial time series prediction," *Swarm and Evolutionary Computation*, vol. 19, pp. 25-42, 2014.

[6] Z. Jin, Y. Yang, and Y. Liu, "Stock closing price prediction based on sentiment analysis and LSTM," *Neural Computing and Applications*, vol. 32, no. 13, pp. 9713-9729, 2020.

[7] F. Wang, Y. Zhang, Q. Rao, K. Li, and H. Zhang, "Exploring mutual information-based sentimental analysis with kernel-based extreme learning machine for stock prediction," *soft computing*, vol. 21, no. 12, pp. 3193-3205, 2017.

[8] J. Shobana and M. Murali, "Adaptive particle swarm optimization algorithm based long short-term memory networks for sentiment analysis," *Journal of Intelligent & Fuzzy Systems*, vol. 40, no. 6, pp. 10703-10719, 2021.

[9] D. Londhe and A. Kumari, "Multilingual Sentiment Analysis Using the Social Eagle-Based Bidirectional Long Short-Term Memory," *International Journal of Intelligent Engineering & Systems*, vol. 15, no. 2, 2022.

[10] J. Shobana and M. Murali, "An efficient sentiment analysis methodology based on long short-term memory networks," *Complex & Intelligent Systems*, vol. 7, no. 5, pp. 2485-2501, 2021.

[11] S. Wu, Y. Liu, Z. Zou, and T.-H. Weng, "S_I_LSTM: stock price prediction based on multiple data sources and sentiment analysis," *Connection Science*, vol. 34, no. 1, pp. 44-62, 2022.

[12] Y. Shi, Y. Zheng, K. Guo, and X. Ren, "Stock movement prediction with sentiment analysis based on deep learning networks," *Concurrency and Computation: Practice and Experience*, vol. 33, no. 6, p. e6076, 2021.

[13] P. Koukaras, C. Nousi, and C. Tjortjis, "Stock market prediction using microblogging sentiment analysis and machine learning," in *Telecom*, 2022, vol. 3, no. 2: MDPI, pp. 358-378.

[14] Z. Jin, Y. Yang, and Y. Liu, "Stock closing price prediction based on sentiment analysis and LSTM," *Neural Computing and Applications*, vol. 32, pp. 9713-9729, 2020.

[15] Y. Liu, Z. Qin, P. Li, and T. Wan, "Stock volatility prediction using recurrent neural networks with sentiment analysis," in *Advances in Artificial Intelligence: From Theory to Practice: 30th International Conference on Industrial Engineering and Other Applications of Applied Intelligent Systems, IEA/AIE 2017, Arras, France, June 27-30, 2017, Proceedings, Part I 30*, 2017: Springer, pp. 192-201.

[16] L. Nemes and A. Kiss, "Prediction of stock values changes using sentiment analysis of stock news headlines," *Journal of Information and Telecommunication*, vol. 5, no. 3, pp. 375-394, 2021.

[17] W. Khan, U. Malik, M. A. Ghazanfar, M. A. Azam, K. H. Alyoubi, and A. S. Alfakeeh, "Predicting stock market trends using machine learning algorithms via public sentiment and political situation analysis," *Soft Computing*, vol. 24, pp. 11019-11043, 2020.

[18] F. Wang, Y. Zhang, Q. Rao, K. Li, and H. Zhang, "Exploring mutual information-based sentimental analysis with kernel-based extreme learning machine for stock prediction," *soft computing*, vol. 21, pp. 3193-3205, 2017.

[19] K. O'Shea and R. Nash, "An introduction to convolutional neural networks," *arXiv preprint arXiv:1511.08458*, 2015.

[20] J. Gu *et al.*, "Recent advances in convolutional neural networks," *Pattern recognition*, vol. 77, pp. 354-377, 2018.

[21] W. Antweiler and M. Z. Frank, "Is all that talk just noise? The information content of internet stock message boards," *The Journal of finance*, vol. 59, no. 3, pp. 1259-1294, 2004.