Educational Administration: Theory and Practice
2024, 30 (8) 575-592
ISSN: 2148-2403
https://kuey.net/

Research Article

Stock market prediction using optimized long-short term memory based on improved salp swarm optimization

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

[1,2*]Department of Computer Science, Vellalar College for Women, Erode, Tamil Nadu, India
[1]K S Rangasamy College of Technology, Tiruchengode, Tamil Nadu, India

***Corresponding Author:** E.S. Samundeeswari
Department of Computer Science, Vellalar College for Women, Erode, Tamil Nadu, India

Citation: K.Kiruthika.et.al. (2024). Stock market prediction using optimized long-short term memory based on improved salp swarm optimization *Educational Administration: Theory and Practice, 30(8),* 575-592
Doi: 10.53555/kuey.v30i8.7642

ARTICLE INFO

ABSTRACT

The prediction of stock price volatility is thought to be one of the most fascinating and important study topics in the financial sector. Deep learning (DL) uses cutting-edge computing technology to analyze data, particularly in the discipline of finance, to deliver insightful analysis. Due to the advantages of sequential learning, the LSTM network has demonstrated notable performance when it comes to time series prediction. It might be very difficult for a researcher to build and choose the best computationally optimal LSTM network for stock market forecasting. The model's ability to learn is impacted by multiple hyperparameters that it must control due to the nature of the data. In addition, several earlier research used heuristics based on trial and error to guess these parameters, which may not always result in the best network. Furthermore, the hyper-parameter values have a significant impact on LSTM model accuracy. To increase the efficiency and precision of the stock prediction method, an improved salp swarm optimization (ISSA) algorithm is used in this research to find satisfactory LSTM model parameters. The ISSA approach uses partial opposition-based learning (POBL) to enhance the population diversity and avoid local optima problems. To verify the ability of the proposed ISSA-LSTM prediction method, four different stock market datasets are considered for experiments. The experimental results confirmed that the developed optimized ISSA-LSTM approaches produced high prediction accuracy and fast convergence rate.

Keywords: Long-short term memory; Salp swarm optimization; Partial opposition-based learning; Stock price prediction; Financial analysis

1. Introduction

The stock market is a marketplace where company shares and derivatives can be exchanged at set prices. The stock market is one of the fastest-growing divisions of any country, driven by supply and demand for shares. These days, a large number of people work directly or indirectly in this industry. As a result, keeping up with industry developments becomes essential. [1]. Predictions of stock prices get more and more interesting as the stock market moves. However, because stocks can fluctuate greatly in value, it can be difficult to predict stock values. Frequently, the stock market is a noisy, chaotic, non-parametric, non-linear system [2]. Moving averages (MA), exponential moving averages (EMA), and autoregressive integrated moving averages (ARIMA) are three statistical techniques that have been used to forecast stock prices in the past. However, statistical approaches are not performing well because of complex, noisy data that may not be sufficient for complex nonlinear models and because they are unable to handle the discontinuity of stock market data [3].

Deep learning approaches are therefore sophisticated machine learning approach that has the learning ability and the propensity to capture complicated non-linear models. The difficulties brought on by the complexity of financial time series can be resolved due to deep learning's strong data processing capabilities. Consequently, there are many opportunities for deep learning and finance. RNNs are effective for time series and other sequential data. They can identify long-term trends and temporal relationships in previous stock prices. RNNs are frequently used for stock price prediction, and one of the most well-known methods is LSTM. LSTM networks are a preferred option for stock price prediction due to their capacity to recognize long-term relationships in sequential data. Traditional RNNs have the vanishing gradient problem, which is fixed by LSTM, enabling them to successfully learn from lengthy data sequences. LSTM is especially helpful for tasks like stock price prediction when historical data greatly affects forecasts made in the future. The core of our study is an optimized LSTM network that integrates SI-based optimization techniques because it overcomes the processing limitations of heuristic estimates of architectural hyperparameters based on trial and error.

The SSA is an optimization method inspired by nature that takes signals from the group behavior of salps, and ocean animals that resemble jellyfish. To solve numerous optimization issues, SSA, a relatively new addition to the field of SI and optimization techniques, was developed. In comparison to other well-known techniques, SSA outperformed them on a variety of tasks, demonstrating its superiority. SSA also offers a variety of additional qualities, including flexibility and simplicity. Additionally, SSA showed its capacity to address both large- and small-scale issues and distinguished itself by its adaptability and highly stochastic character. However, SSA faces two problems such as population diversity and local optimum.

Numerous research in the literature has utilized the opposition-based learning (OBL) approach [4] to enhance various optimization techniques such as GA[5], artificial bee colony (ABC) [6], PSO[7], BCO[8], and SSA[9]. It was shown by an OBL that it may enhance these algorithms' population diversity and convergence rate. Nevertheless, OBL selects the best solutions and finds the worst options for every solution throughout each iteration; as a result, the search algorithm will get more complex [10]. This study, which was inspired by the concept of OBL, proposes a partial OBL called POBL, an enhanced OBL. The POBL is designed to compute partial opposite points of an estimate rather than only looking at the opposing point of a candidate. POBL is used to improve the SSA approach by initializing the population and updating position to enhance population diversity, and convergence rate and reduce the complexity. So, the original SSA algorithm's characteristics are carried over into the suggested ISSA. According to the suggested enhancements, ISSA is superior to the original SSA for its capacity to strike a balance between exploitation and exploration and avoid local optima. The proposed ISSA approach is used to fine-tune the hyperparameters of the LSTM approach for predicting stock prices. Contributions of research are as follows;

- The current study's new stock price prediction algorithm is based on an optimized LSTM.
- The ISSA is used in the optimized process to identify the best LSTM parameters.
- The ISSA uses the POBL method to enhance the population diversity and avoid local optima and convergence rates.
- To assess the effectiveness, the designed prediction technique is contrasted with certain well-known BPNN, standard LSTM, and variant LSTM methods.
- Results of the experiments done on four distinct financial datasets to evaluate the reliability of created prediction techniques

The paper is ordered as follows: section 2 covers related works; Section 3 covers LSTM, SSA, OBL, POBL, and ISSA algorithm; Section 4 covers optimized LSTM; Section 5 covers the results analysis; and Section 6 concludes the research work.

2. Related works

Researchers and analysts can better comprehend the various models and approaches that have been employed in the past for stock market forecasting by doing a literature study. It offers details on the historical evolution of predictive models, as well as their advantages and disadvantages. Researchers can find benchmark models and methodologies that have done well in prior studies by evaluating the body of existing literature. This enables them to evaluate the effectiveness of their suggested models or techniques in comparison to industry standards.

K. Liu et al. (2021) [11] developed a brand-new sentiment-based LSTM to forecast stock prices using cutting-edge internet data sources. Individual investors frequently get trade information from internet social media sites for several growing nations. Consequently, stock features pulled from social media sites are likely to provide useful data. We created daily social networks using data we collected from EastyMoney, China's largest social media site, on individuals and the stocks they followed. After that, we added the network variable for each stock to the conventional variables and used the LSTM to forecast the close prices of the SSE 50 member stocks. Y. Wang et al. (2021) [12] introduced Direct input-to-output connections (DIOCs) into the Elman Neural Network (ENN) for stock price prediction. it is demonstrated that the developed ENN with these connections (Elman-DIOCs) performs noticeably better than the original ENN without them. Depending on the presence of hidden layer biases, output layer biases, and DIOCs, the author performs a systematic evaluation of 8 models. B. Yan et al. (2020)[13] developed the CEEMD-PCA-LSTM model for stock markets using a revolutionary deep-learning prediction approach. As a sequence smoothing and decomposition module in this model, complementary ensemble empirical mode decomposition (CEEMD) can step-by-step decompose the fluctuations or trends of various time series scales, producing a series of intrinsic mode functions (IMFs) with various characteristic scales. The deconstructed IMF component's dimension is then reduced while still preserving the majority of the raw data's information, removing any unnecessary data and enhancing forecast response time. The closing price of the following trading day for each component is then predicted using high-level abstract attributes that are independently input into LSTM networks. Finally, a final anticipated value is obtained by combining the predicted values of the constituent components.

A. Dezhkam et al. (2022) [14] developed a new stock price prediction method using a tri-state labeling method to categorize the underlying patterns into up, down, and no-action groups. The load of denoising the dataset as a preprocessing work is lessened by the addition of a no-action state in our unique technique. Using ML and DL models, we experiment with the performance of our labeling system. By using the Bayesian optimization approach to choose the optimal hyperparameter tuning values, the framework is enhanced. The necessary trading signals are produced by the price trend prediction module. J. Zhao et al. (2021) [15] developed a new method for developing stock price trend prediction models based on RNN/LSTM/GRU with the attention mechanism called AT-RNN-M, ATLSTM-M, and AT-GRU-M). According to the Authors, stock data has a long memory, meaning that changes in stock prices are tightly tied to past transaction data, and stock RNNs are effective at extracting characteristics from time series. S. Hochreiter et al. (2018)[16], RNNs are used to forecast stock closing prices. As a unique approach to forecasting the closing price of the stock market, an LSTM model, a sort of RNN, is developed. It is combined with stock fundamental transaction data and technical indicators. PCA is used to achieve dimension reduction for the technical indicators. Adaptive moment estimation (Adam) and Glorot uniform initialization are two optimization techniques that are used to train the model. Case studies are done on the NASDAQ, Apple (AAPL), and S&P 500.

L. Zheng et al. (2021) [17] have presented a hybrid prediction model that combines PCA and RNN to better forecast share prices for the aerospace industry sector and fully comprehend the influence of different factors on stock prices. The created model may function as an intelligent agent in an autonomous stock prediction system, allowing the financial sector to quickly decide on their economic plans and commercial ventures based on anticipated future share prices, increasing return on investment. Z. Berradi et al. (2019) [18] provided a PCA and simple RNN (SRNN)-based feature selection-based stock price prediction approach. The purpose of the work is to select the finest number of inputs from the eight characteristics using PCA, then apply the model SRNN to forecast the price of 29 days. F. Yang et al. (2021) [19] developed a prediction method using a neural network to enhance the performance based on the enhanced PSO. To enhance the global ability and the local search ability of PSO, the adaptive inertial weight is proposed. The proposed method is predicated on the notion of avoiding particles as much as possible from entering the same local solution and constantly maintaining the particles with a specified variety. S. Luo et al. (2020)[20] suggested a rapid sub-step Grid Search (SGS) method to optimize the parameters of the LSTM, which is used for time series analysis. This method led to improved efficiency and reduced prediction errors. J. Rasheed et al. (2020) [21] developed a new stock prediction method utilizing DL-based approaches. The one-dimensional convolutional neural network (1D-CNN) and the LSTM are two widely used techniques that were looked at. PCA was also used to examine the impact of dimensionality reduction on the prediction accuracy of both 1D-CNN and LSTM. Each approach underwent two distinct tests, one using PCA and the other not using PCA.

3. Research methods

The following subsection discusses the fundamentals of the research method used in this study such as LSTM, SSA, OBL, POBL, and ISSA.

3.1 LSTM

LSTMs are distinct from conventional feed-forward models in that they have feedback connections [22]. The key to LSTM is an overview of a gating unit system that keeps prior information over the cell state unit's internal memory unit while incorporating numerous "gates" to allow the network to learn continually, i.e., when to forget previous knowledge or update cell state with current input. The internal memory, which stores all previous data up to the present, is controlled by three "gates". The internal memory unit's internal input gate controls how much fresh data is stored inside which is determined as follows,

$$g_i^t = \sigma(\sum_j U_{i,j}^t x_j^t + \sum_j W_{i,j}^g h_j^{t-1} + b_i^g) \tag{1}$$

$\sigma$is the activation function (sigmoid). x^t is the input vector. h^t is the present hidden layer vector, which contains all LSTM cells' output. b^g, U^g, and w^g are the input gate's bias, weights of input, and recurrent, correspondingly. The internal memory unit's internal memory unit is designated as the forget gate unit, which controls how much information from earlier times should be stored there which is determined as follows,

$$f_i^t = \sigma(\sum_j U_{i,j}^f x_j^t + \sum_j W_{i,j}^f h_j^{t-1} + b_i^f) \quad (2)$$

σ – is a sigmoid function that ranges from 0 and 1. b^f, U^f, and w^f are the bias, weights, and recurrent values, correspondingly. The LSTM cell's internal state unit S_i^t, which contains a conditional self-loop weight f_i^t, is updated, as follows:

$$S_i^t = f_i^t S_i^{t-1} + g_i^t \sigma(\sum_j U_{i,j} x_j^t + \sum_j W_{i,j} h_j^{t-1} + b_i) \quad (3)$$

Where b is the bias, w is the weights of input, and U is the recurrent. The cell state information measured by the forget gate at the most recent moment is on the right side of the equation above, and the input measured by the input gate is on the second fragment. The output gate controls the internal memory unit to produce and give the necessary data, which can be provided by:

$$h_j^t = tanh(S_i^t) O_i^t \quad (4)$$

$$O_i^t = \sigma(\sum_j U_{i,j}^0 x_j^t + \sum_j W_{i,j}^0 h_j^{t-1} + b_i^0) \quad (5)$$

Where, $b^0$is the output bias, U^0 is the inputs, and w^0 is the connection weights. The LSTM unit's output serves as a hidden layer in the RNN.

3.2 Salp swarm optimization

SSA, created by Mirjalili et al. in 2017 [23], is a topical optimization algorithm to be developed. Algorithm 1 shows the steps of SSA. In their nature, SSA behaves like Salps. Salps are a large mammal species of the Salpidae family. Salps have translucent bodies that resemble jellyfish and barrel-shaped bodies. Additionally, salp moves resemble jellyfish's method of movement. The salps' swarming behavior is the most intriguing strategy; in the deep waters, the salps construct a chain known as the salp chain by swarming together. The population is initially split into the leader and followers' groups to mathematically model the salp chains. The salp at the front of the chain is the leader, while the other salps are referred to as followers. As their name suggests, these salps have a leader who directs the swarm while the followers follow the leader either directly or indirectly. Consequently, a two-dimensional matrix named x is used to hold the positions of all salps. Additionally, it may be deduced that the swarm is aiming for a food supply with the designation$F$ in the search space. The update on the leader's position is as follows,

$$X_j^1 = \begin{cases} F_j + c_1((ub_j - lb_j)c_2 + lb_j) & c_3 \geq 0 \\ F_j + c_1((ub_j - lb_j)c_2 + lb_j) & c_3 < 0 \end{cases} \quad (6)$$

Where X_j^1 – represents the leader position. F_j – represents the food source position, ub_j is the upper bound and lb_j is the lower bound, c_1,c_2, and c_3 are random numbers. The coefficient c_1 is the most significant in SSA for balancing exploration and exploitation which is as follows:

$$c_1 = 2e^{-\left(\frac{4t}{L}\right)^2} \quad (7)$$

Where$l$ and L are present and maximum iterations. The c_2 and c_3 are uniformly produced random numbers between [0, 1]. Updating the followers' positions:

Algorithm 1: SSA
Salp $x_i(i = 1,2, \dots, , \mathrm{n})$ population initialization
While (condition is not satisfied)
Determine each search agent's fitness
F= best search agent
Update the value of c_1
For each salp
If (i= =1)
Reposition the leading salp
Else
Reposition the follower salp
End if
End for
Salps position should be adjusted based on the upper and lower boundaries of the variables

$$X_j^i = \frac{1}{2}\left(X_j^i + X_j^{i-1}\right) \quad (8)$$

Where $i \geq 2$ and X_j^i represent the position of i^{th} follower salp in j^{th} dimension.

3.3 Opposition-Based Learning

The OBL intelligence method was developed by H R Tizhoosh to obtain an estimate that is the opposite of the present estimate to enhance the effectiveness of the offered solution [24]. Population-based optimization methods often start by generating a collection of solutions. Using previous information or a random generator, the population may be created. The optimization approach is then used to update the populations. However, if there is no prior knowledge of the answer, the supplied solution cannot converge to a global solution. Additionally, the convergence of the global solution requires extra calculation time. Numerous research has been conducted using the advantages of the OBL for initializing and updating the population to overcome these drawbacks. The OBL has gained popularity as a method for improving the efficiency of many machine learning algorithms in recent years. According to reports, the OBL can increase population variety and boost the efficiency of worldwide searches. For example, numerous SI optimizations [25-30] and ANN [31, 32] methods use the OBL to enhance their convergence rate. An approach for creating optimization methods that look in the opposite direction of the existing solution is provided by the OBL algorithm. The best of the two solutions is picked as the current option after being compared to the alternative. This OBL technique produces a solution that is nearer to the ideal solution with rapid convergence. The next subsections explain the OBL method.

a) Opposite number

Assume, x is the real number. It falls within the range of $m$and n: $x \in [m, n]$. The opposite value $x$is given by $\acute{x}$. The opposite solution $\acute{x}$is defined as follows,

$$\acute{x} = m + n - x \quad (9)$$

The opposite value is assumed as follows, let $x = (x_1, x_2, \dots, x_D)$ be a data sample in D- dimensional space. $x_1, x_2, \dots, x_D \in R$ and$x_i[m_i, n_i]$ $\forall i \in \{1,2, \dots D\}$ Hence, the opposite number is defined by $\acute{x}_i, \dots \acute{x}_D$in D - dimensional space,

$$\acute{x}_i = m_i + n_i - x_i, i = 1,2, \dots, D \quad (10)$$

b) Opposition-based optimization

Let $f(x)$ be the function and $g(.)$is a proper evaluation function. If $x \in [a, b]$ is a random opening value and $\acute{x}$ is an opposite point of x. Calculate $f(x)$and $f(\acute{x})$in every iteration. The process of learning stays with $x$if $g(f(x) \geq g(f(\acute{x})$, otherwise with $\acute{x}$. According to the provided algorithm, the evaluation function $g(.)$ is denoted. It may have functions for weight, fitness, and error.

3.4 POBL

The POBL is a new form of the OBL approach developed by Z. Hu et al. (2014) for enhancing population diversity and avoiding local optima for adaptive differential evolution [33]. An exact opposite point contains the opposites of the original values for each dimension. The partial opposite points may be defined as follows if $P\acute{X}$is the set of partial opposing points of a given point X in D-dimensional space:

$$P\acute{X}^1 = \begin{bmatrix} P\acute{X}_1^1 \\ P\acute{X}_2^1 \\ P\acute{X}_3^1 \end{bmatrix}_{D\times 1} = \begin{bmatrix} x_1 & \acute{x}_2 & \ldots\acute{x}_D \\ \acute{x}_i & x_2 & \ldots\acute{x}_D \\ \vdots & \vdots & \ddots \vdots \\ \acute{x}_1 & \acute{x}_2 & \ldots x_D \end{bmatrix}_{D\times D} \quad (11)$$

The level of partial opposition is indicated by the superscript 1 in $P\acute{X}$. Because each point only includes one original number in one dimension, the partial opposing points mentioned above are of order or degree one. Therefore, there are D number of partial opposing points to a point X in X dimensional space.

3.5 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,

Algorithm 2: ISSA algorithm based on POBL
Salp $x_i(i = 1,2,\ldots,,\mathrm{n})$ population initialization
Calculate the salp opposite population $\acute{x}_{il}(i = 1,2,\ldots,,\mathrm{n})$
Select the n fittest salps from $\{x \cup \acute{x}_{il}\}$ which is considered as the initial population of SSA
While (condition is not satisfied)
Determine each search agent's fitness
F=the best search agent
Update the value of c_1
For each salp
Calculate the opposite $\acute{x}_{il}$ of the present position and
Calculate the partial position of each salp
Calculate its fitness values and compare the present position and its partial opposite position of salp
Select the fittest position as the present position then update the position of both leader and followers
For each salp (x_i)
If (i= =1)
Reposition the leading salp
Else
Reposition the follower salp
End if
End for
Salps position should be adjusted based on the upper and lower boundaries of the variables
End while
Return F

Opposition-based initialization

The position of each leader salp is initialized as follows,

$$X_j^1(t)|_{(t=0)} = X_j^{min} + \left(X_j^{max} - X_j^{min}\right).r_{ij}^u(t)|_{(t=0)} \quad (12)$$

$r_{ij}^u(t)|_{(t=0)}$ - 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 and its opposite swarm are then used to choose the best NP number of places with speeds.

a) Generation jumping

The jth element $\acute{x_{ij}}(t)$ of opposite position $\acute{x_i}(t)$ of i^{th} salp is described in the search space as follows:

$$\acute{x_{ij}}(t) = a_j(t) + b_j(t) - \alpha_{ij}(t).x_{ij}(t) \quad (13)$$

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

$$a_j(t) = \left\{\begin{matrix} min \\ \forall_i \end{matrix} x_{ij}(t)\right\} \quad (14)$$

$$b_j(t) = \left\{\begin{matrix} max \\ \forall_i \end{matrix} x_{ij}(t)\right\} \quad (15)$$

$\alpha_{ij}(t)$ 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:

$$\alpha_{ij}(t) = 1 - \eta . r_{ij}^{c}(t) \tag{16}$$

Where $r_{ij}^{c}(t)$ is the Cauchy factor that distributes a random number. The Cauchy density function is defined by

$$f(x) = \frac{1}{\pi} . \frac{s}{s^2 + x^2}, \quad -\infty < x < \infty \tag{17}$$

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

$$F_s(x) = \frac{1}{2} + \left(\frac{1}{\pi}\right) arctan\left(\frac{x}{s}\right) \tag{18}$$

$\eta = \beta \times \eta$. The initial value of η is 1.0 and β is a uniformly distributed random value that ranges 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:

$$k = \lceil rand() \times (D - 1) \rceil \tag{19}$$

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:

$$l = \lceil rand() \times D \rceil \tag{20}$$

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:

$$\acute{x}_{il}(t) = x_{il}(t) \tag{21}$$

4 The proposed ISSA-LSTM

The ISSA model is employed to optimize the LSTM approach, which forecasts the price of stock indices. Figure 1 shows the flowchart of the proposed ISSA-LSTM approach and algorithm 2 shows the 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 LSTM network with an optimization method like ISSA. This mechanism overcomes processing limitations caused by a heuristic estimate of architectural hyper parameters 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 as follows,

- **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 size, 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.

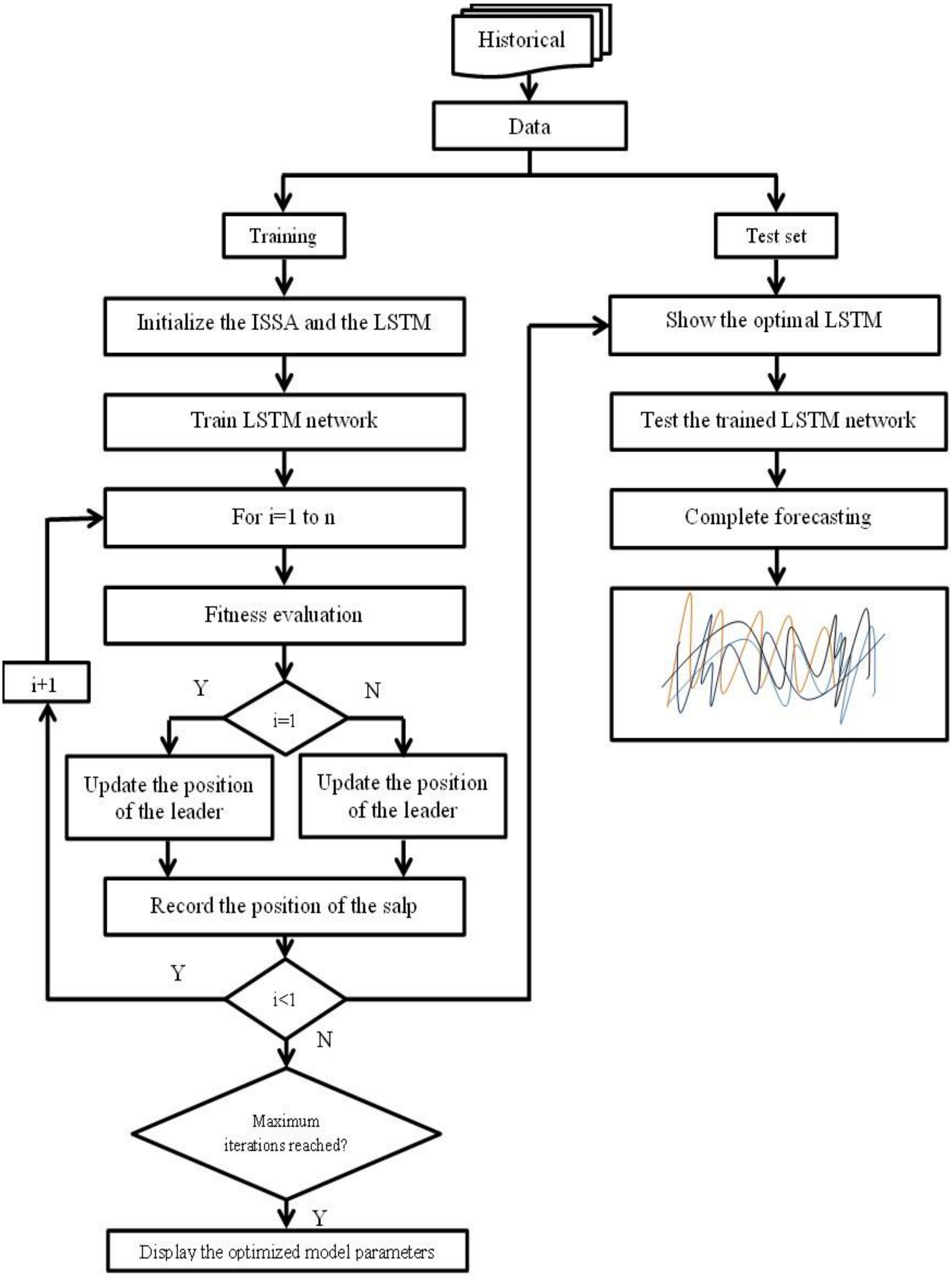

Fig. 1: Overview of the proposed prediction method

Algorithm 3: Proposed ISSA-LSTM

Step 1: Use the Min-Max normalization method to convert the stock price into a real value between [0, 1]. The training and the test datasets are then formed from the historical data.
Step 2: The population size, number of iterations, learning factor, and other required parameters are among the ISSA that should be initialized. Set the epoch, batch size, and the number of nodes as the initial values for the LSTM parameters.
Step 3: The ISSA method optimizes the appropriate hyperparameters in the LSTM model.
Step 3.1: LSTM network training. The MSE is the selected loss function. Update the loss function by repeating the training iterations.
Step 3.2: The fitness of each salp should be evaluated, and the global and individual ideal fitness values should be determined.
Step 3.3: The salp's location is continually modified during the iterative procedure by the POBL. The salps will change with the partial opposite local of the current location, increasing the likelihood that they will seek new places.
Step 3.4: The ideal LSTM hyperparameters are captured when the ISSA's end conditions are satisfied. If

5 Experimental results and discussions

Researchers and practitioners can assess how well various prediction models and algorithms work based on experimental data. They shed light on how effectively a specific model outperforms others in predicting stock values. The present research work develops a new stock prediction method based on LSTM and ISSA for stock market prediction. The developed ISSA-LSTM method is compared with some well-known conventional BPNN and variant LSTM methods such as LSTM [34], GA-LSTM [35], PSO-LSTM, IPSO-LSTM [36], and SSA-LSTM [37]. The models are implemented on a laptop with Intel Core i3-10510U CPU running at 1.8 GHz, and 8GB of RAM. MATLAB 2015R is used for development, while Windows 10 is the host operating system.

5.1 Data collections

Four stock market datasets were obtained from Yahoo [38] such as the S&P BSE Sensex and the Nifty 50 are two stock indexes and Two Indian stock bank prices: State Bank of India (SBIN) and ICICI Bank (ICICI). The data sets span the period from January 2015 to December 2022. Training and testing phases are separated for the acquired datasets. 70% of the datasets are taken into account for training and the remaining 30% are reviewed for testing. In the stock market, technical indicators are essential for assisting traders and investors in making wise choices. These estimates, which are based on previous price, volume, or open interest data, are known as indicators. They offer perceptions of market patterns, prospective turning points, and the general health of a stock or market. Technical indicators are heuristic or mathematical computations that are based on the examination of previous trading activity, notably price fluctuations of securities and trading volume, to complete technical analysis for predicting future price movement. Table 1 shows the detail of technical indicators.

The scaling is finished as follows,

$$\bar{x} = \left(\frac{x - \min(x)}{\max(x) - \min(x)}\right) \tag{22}$$

Where, X_i, X_{min}, X_{max} are present, minimum and maximum values of input data samples. After preprocessing datasets, the datasets split into two sets such as training and testing sets. 70 % is considered as training and 30 % is testing sets from overall datasets.

Table 1 : Details of 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 mean value of a particular $'t'$ day
2	10-days Moving Average	$MV_{10} = \frac{x_1 + x_2 + \cdots + x_n}{n}$	The simple moving average's average over the previous 10 days
3	Momentum	$M = C_t - C_{t-4}$	It calculates the amount of the stock price over a specified period.
4	Stochastic K%	$STCK = \frac{C_t - LL_{t-n}}{HH_{t-n} - LL_{t-n}} \times 100$	Stochastic offers a way to determine the speed of price fluctuation. $K\%$ Measures how the current closing price has changed about other prices throughout time.
5	Stochastic D%	$STCD = \frac{\sum_{i=0}^{n-1} K_{t-1\%}}{n}$	$D\%$ requires the three days moving averages of K%
6	Relative Strength Index (RSI)	$= 100 - \frac{100}{1 + (\sum_{t=0}^{n-1} \mathrm{UP}_{t-1}/n)/(\sum_{t=0}^{n-1} \mathrm{DW}_{t=1}/n)}$	It is a movement oscillator that measures the rate and direction of price movement on a scale from 0 to 100.
7	Williams %R	$LW = \frac{H_n - C_t}{H_n - L_n} \times 100$	It is a momentum indicator that establishes entry and exit points in the market by measuring overbought and oversold levels. It evaluates a stock's close about its high-low range for a specific amount of time
8	Moving Average Convergence Divergence(MACD)	$= MACD(n)_{t-1} + \frac{2}{n+1} \left(Diff_t - MACD(n)_{t-1}\right)$	The purpose of MACD is to approximate a stock's future direction by matching up to its short- and long-term momentum.
9	Commodity Channel Index(CCI)	$CCI = \frac{M_t - SM_t}{0.015 D_t} \times 100$	It compares the current price level to the average price level over a specific period to spot emerging trends or signal severe situations.
10	Price Oscillator (PO)	$\mathrm{PO} = \frac{\mathrm{MA}_5 - \mathrm{MA}_{10}}{\mathrm{MA}_5}$	A technical indicator called PO demonstrates the connection between two moving averages.

C_t- The closing price of the day, L_t- Lowest price of the day, H_t- High price of the day, LL_t- Lowest Low price of the day, HH_t-Highest high price of the day, UP_t- Upward price of the day, and DW_t- Downward price of the day.

5.2 Parameter settings

A critical stage in creating a deep learning model for various objectives, such as stock market prediction, is setting the parameters of an LSTM network. The performance of the model can be considerably impacted by proper parameter adjustment. The following values are used for LSTM such as hidden size = 10, output size = 1, layer num = 1, Activation function = Relu, Optimization function = Adam, learning rate = 0.05, betas = (0.9, 0.999), eps = 1×10^{-8}; loss function = MSE, maximum training epoch = 200 and error value = 0.005.

5.3 Performance measures

Performance measurements are essential for stock market forecasting because they offer a quantitative evaluation of the precision and effectiveness of predictive models. They support decision-making by assisting academics, investors, and traders in assessing the accuracy of their forecasts. The performance of the developed method is measured with four different measures namely Root Mean Square Error (RMSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and R-Square (R^2) which are defined as follows,

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

$$\mathrm{MAE} = \frac{1}{m}\sum_{i=1}^{m}|(y_i - \bar{y}_i)| \tag{24}$$

$$\mathrm{MAPE} = \frac{100}{m}\sum_{i=1}^{m}\left|\frac{(y_i - \bar{y}_i)}{y_i}\right| \tag{25}$$

$$R^2 = 1 - \frac{\left(\frac{1}{m}\sum_{i=1}^{m}(y_i - \bar{y}_i)^2\right)/m}{\left(\frac{1}{m}\sum_{i=1}^{m}(\hat{y}_i - \bar{y}_i)^2\right)/m} \quad (26)$$

Where,y_i - target value, $\bar{y}_i$ - predicted output, m - total number of data points. $\hat{y}_i$ - mean of real values. Stronger ability and a better model are indicated by values that are closer to 1.

5.4 Results analysis

Results analysis and discussion are essential when utilizing deep learning algorithms to predict the stock market. The main goal of outcomes analysis is to evaluate how well the stock market price prediction models performed. This entails assessing the model's generalization performance by contrasting its forecasts with real market data. It aids analysts and investors in assessing the accuracy of the model's forecasts. The present research work uses the ISSA-LSTM approach for predicting stock market prices. The present section shows the ability of the developed approach by comparing various variants of LSTM based on performance measures.

Tables 2, 3, 4, and 5 show the performance results analysis prediction models for Nifty 50, S & P Sensex, SBIN, and ICICI datasets respectively. The graphical representation of results comparisons is shown in Figures 2,3,4 and 5 based on performance metrics such as RMSE, MAPE, MAE, and R2, respectively. Figures 6,7,8 and 9 show the performance comparisons based on prediction outcomes vs actual outcomes. From overall experimental results, the developed prediction approach produced higher generalization performance when compared with other prediction approaches. The ISSA-LSTM, SSA-LSTM, IPSO-LSTM, PSO-LSTM, GA-LSTM, LSTM, and BPNN prediction models have the highest to lowest fitting degrees based on the predictions of the latter models. On all stock data sets, the ISSA-LSTM model's prediction and evaluation criterion is less than those of other models, and R2 values are closer to 1.

In the ISSA-LSTM model, the Nifty 50’s RMSE, MAPE, MAE, and R2 values are 0.0085, 0.0392, 7.9327, and 0.9937, respectively. The ISSA-LSTM model predicts stock prices that are higher than the average of the several versions and the traditional LSTM and BPNN models. In the ISSA-LSTM model, the S & P Sensex’s RMSE, MAPE, MAE, and R2 values are 0.0401, 0.3026, 7.9473, and 0.9837, respectively. The ISSA-LSTM model predicts stock prices that are higher than the average of the several versions and the traditional LSTM and BPNN models. The SBIN's RMSE, MAPE, MAE, and R2 values in the ISSA-LSTM model are 0.0083, 0.0056, 8.1209, and 0.9837, respectively. The ISSA-LSTM model forecasts stock prices higher than the mean of its several iterations as well as the conventional LSTM and BPNN models. The ICICI's RMSE, MAPE, MAE, and R2 values in the ISSA-LSTM model are 0.0091, 0.0037, 5.3823, and 0.9836, respectively. The ISSA-LSTM model forecasts stock prices higher than the mean of its several iterations as well as the conventional LSTM and BPNN models.

Table 2: Performance results for the Nifty 50 datasets

Approaches	RMSE	MAPE	MAE	R^2
ISSA-LSTM	**0.0085**	**0.0392**	**7.9327**	**0.9937**
SSA-LSTM	0.0096	0.0421	7.3729	0.9794
IPSO-LSTM	0.0107	0.0573	6.8192	0.9537
PSO-LSTM	0.0132	0.0659	5.7261	0.9391
GA-LSTM	0.0154	0.0749	4.6492	0.9002
LSTM	0.0169	0.0087	3.0218	0.8687
BPNN	0.0178	0.0093	2.8782	0.7925

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

Approaches	RMSE	MAPE	MAE	R2
ISSA-LSTM	**0.0401**	**0.3026**	7.9473	0.9837
SSA-LSTM	0.0483	0.3264	7.3417	0.9631
IPSO-LSTM	0.0565	0.0391	6.5741	0.9522
PSO-LSTM	0.0738	0.0467	4.6581	0.9333
GA-LSTM	0.0816	0.0574	3.6456	0.8927
LSTM	0.0993	0.0736	2.1247	0.8373
BPNN	0.0955	0.0844	1.4982	0.8274

Table 4 : Performance results for the SBIN datasets

Approaches	RMSE	MAPE	MAE	R^2
ISSA-LSTM	**0.0083**	**0.0056**	**8.1209**	**0.9837**
SSA-LSTM	0.0091	0.0063	7.5512	0.9647
IPSO-LSTM	0.0121	0.0071	6.9435	0.9317
PSO-LSTM	0.0192	0.0079	6.2680	0.9153
GA-LSTM	0.0215	0.0085	5.8561	0.8826
LSTM	0.0275	0.0073	4.8675	0.8415
BPNN	0.0349	0.0089	3.2472	0.7928

Table 5 : Performance results for the ICICI datasets

Approaches	RMSE	MAPE	MAE	R^2
ISSA-LSTM	**0.0091**	**0.0037**	**5.3823**	**0.9836**
SSA-LSTM	0.0172	0.0051	5.2153	0.9261
IPSO-LSTM	0.0225	0.0063	4.9169	0.8727
PSO-LSTM	0.0297	0.0071	4.1721	0.8364
GA-LSTM	0.0375	0.0083	3.7387	0.8019
LSTM	0.0425	0.0097	3.1629	0.7452
BPNN	0.0498	0.0114	2.9352	0.7204

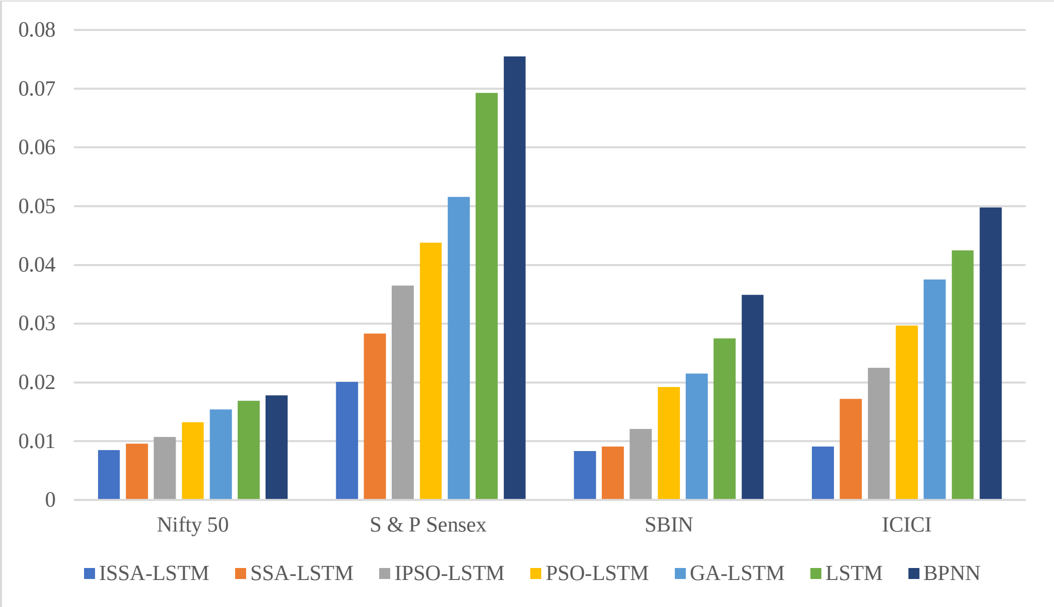

Fig. 2 : Performance analysis of prediction model based on RMSE

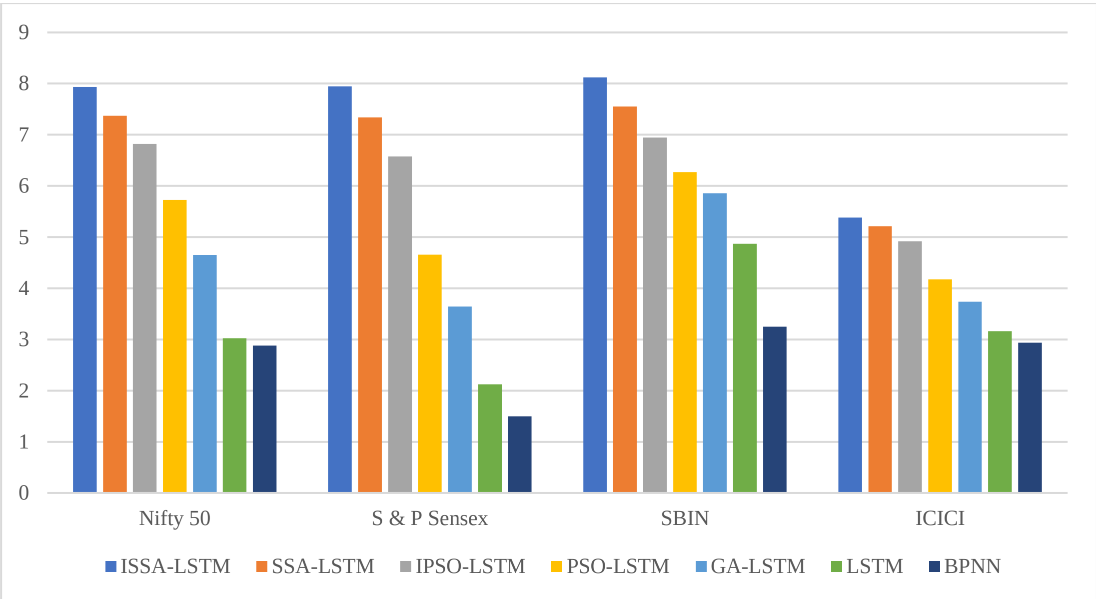

Fig. 3 : Performance analysis of prediction model based on MAPE

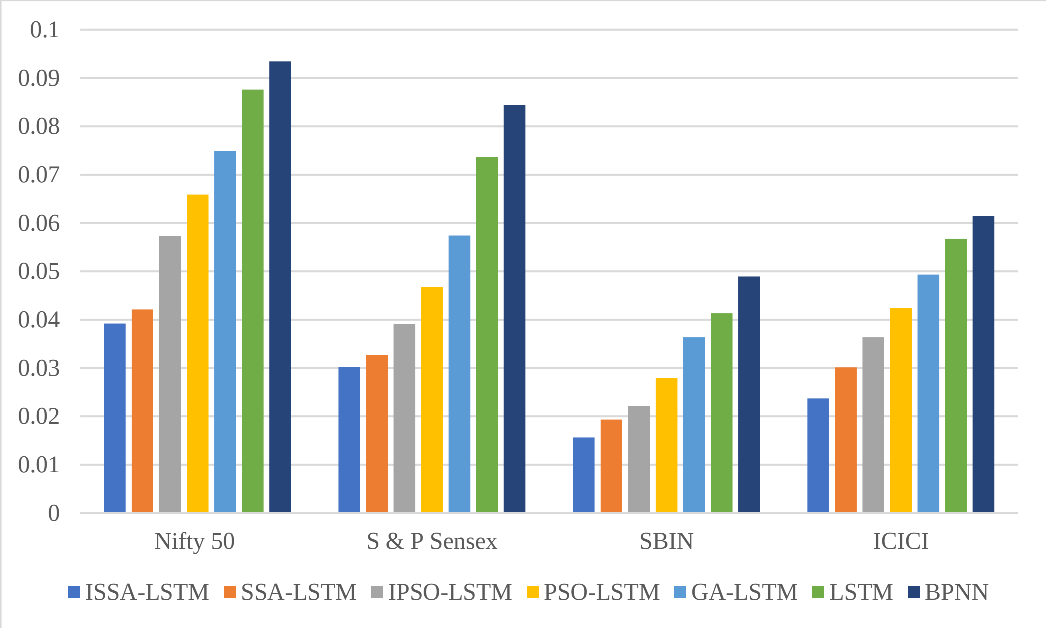

Fig. 4 : Performance analysis of prediction model based on MAE

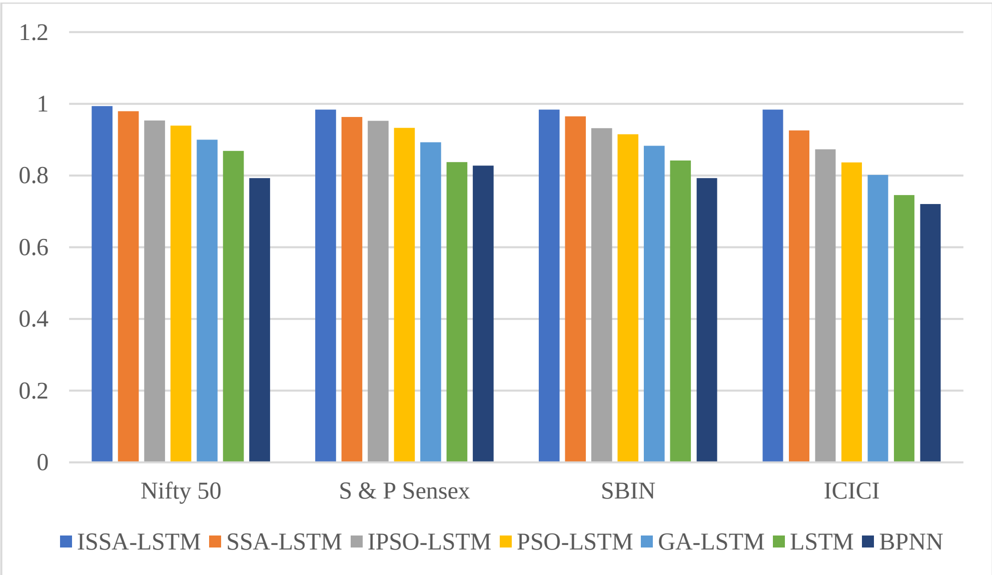

Fig. 5 : Performance analysis of prediction model based on R2

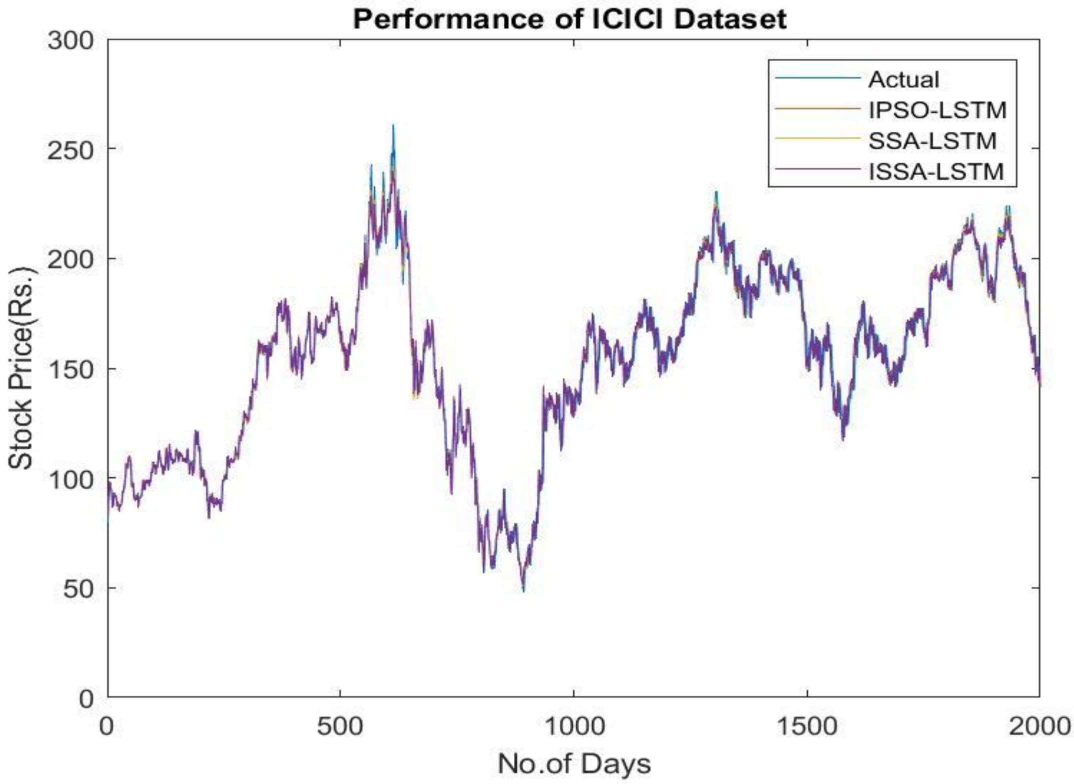

Fig. 6 : Performance result comparisons of ICICI datasets

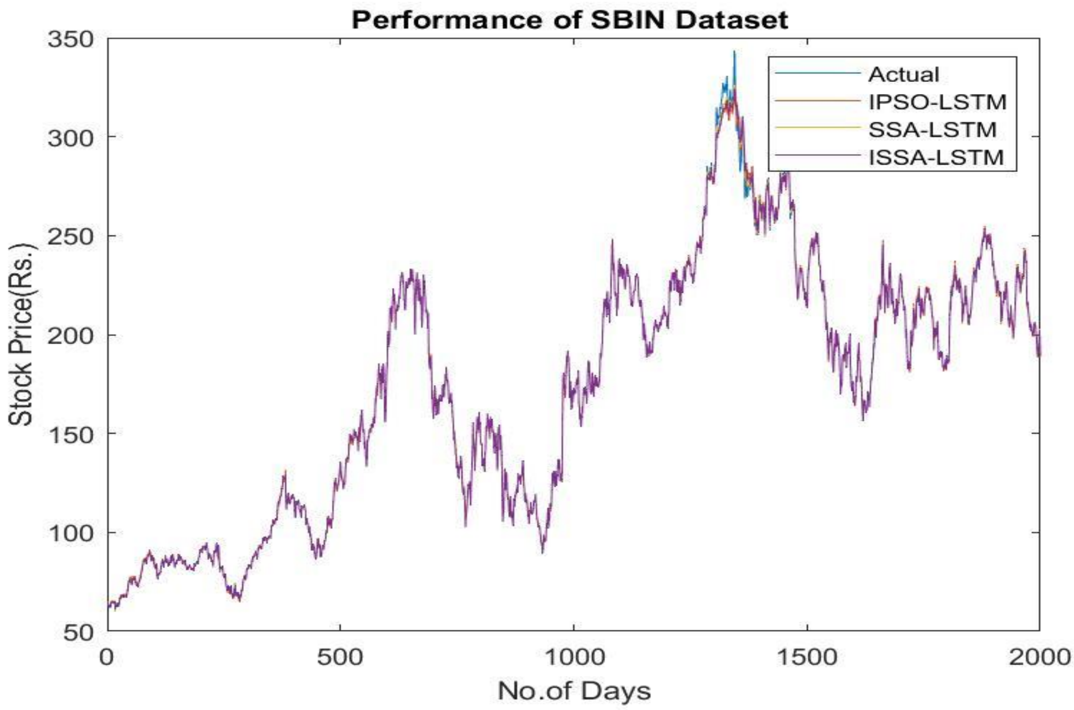

Fig. 7 : Performance result comparisons of SBIN datasets

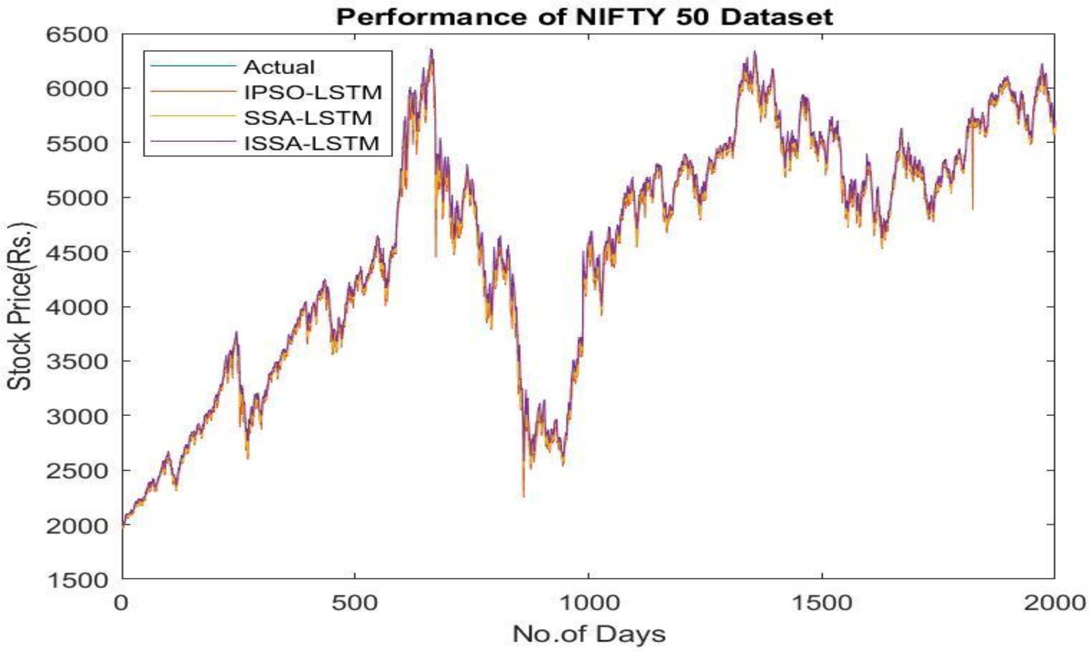

Fig. 8 : performance result comparisons of Nifty 50 datasets

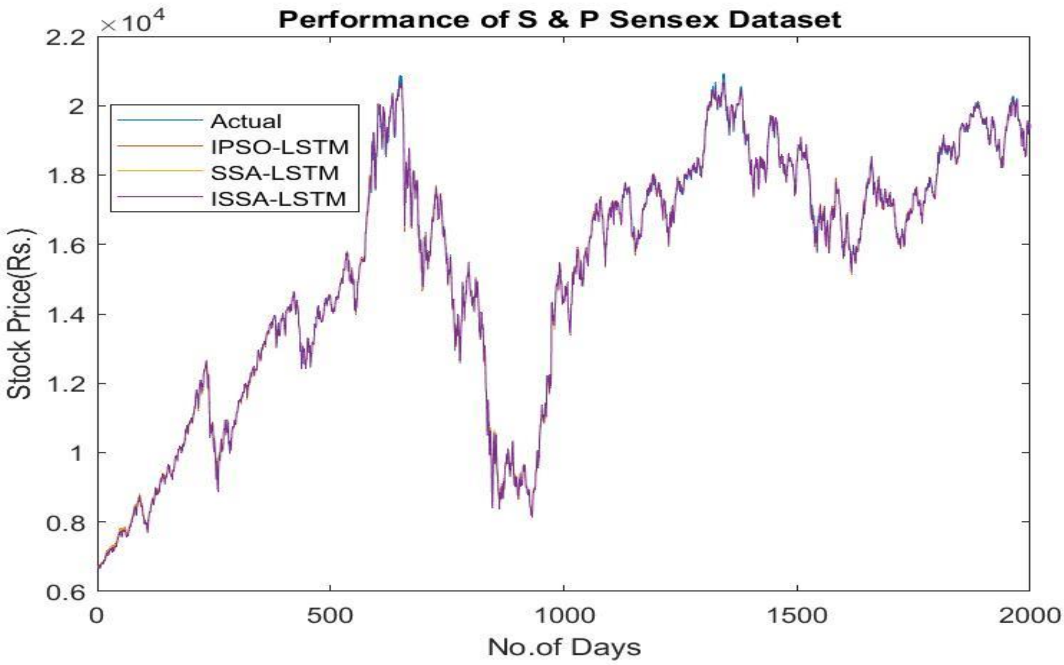

Fig. 9 : performance result comparisons of S & P Sensex datasets

In summary, the proposed strategy constantly attains the highest accuracy, the lowest time offset, and the closest predictive value using optimized LSTM. 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 and sustainable development of the economy with more timely and appropriate regulation, accurate and timely stock price prediction, and sound stock market guidance. The experimental findings reveal that for all four main indices, the predicting curve is extremely near to the real value, demonstrating the universality and resilience of the proposed ISSA-LSTM algorithm to various stock markets. In general, the above results demonstrate that the proposed ISSA-LSTM model performs significantly better than the other methods in most cases, and it is certainly a good learning model that is accurately what the depositors or buyers require in the real trading marketplace, taking both accuracies in prediction and forecast speed into account at the same time.

6 Conclusions

Accurate stock market price forecasting is crucial for capitalizing on the risk of financial gain or loss during an investment strategy. This requires assessing the most volatile data, which are highly adaptable and need to dynamically modify to recognize market behavior. As a result, several strategies have been put forth in the past to create an accurate model and forecast the value of financial time series. Additionally, the LSTM network has been used as a benchmark model in several publications because of its great performance and the essential benefit of using memory cells to retain input for sequential learning. Despite their effectiveness, the optimal learning architectural features of this network remain an unsolved task for scientists. In this study, we explored novel optimization approaches using improved swarm intelligence optimization algorithms to build and optimize the LSTM network. The model-building procedure, which involves fine-tuning the LSTM network's most sensitive hyperparameters, is the key to our success. Global searching, a crucial element of these methods, has removed the existing model's drawbacks based on several metrics, such as a quick rate of convergence, fewer iterations needed to find an optimal solution, a shorter training time due to fewer epochs, a very smooth validation loss, and—most importantly—better performance with increased accuracy. In further studies, feature selection and parameter optimization will be accomplished using alternative heuristics, meta-heuristics, or hybrid algorithms.

References

[1] L. Cao, "Ai in finance: challenges, techniques, and opportunities," ACM Computing Surveys (CSUR), vol. 55, no. 3, pp. 1-38, 2022.

[2] D. Selvamuthu, V. Kumar, and A. Mishra, "Indian stock market prediction using artificial neural networks on tick data," Financial Innovation, vol. 5, no. 1, pp. 1-12, 2019.

[3] H. Yu and Q. Dai, "DWE-IL: a new incremental learning algorithm for non-stationary time series prediction via dynamically weighting ensemble learning," Applied Intelligence, vol. 52, no. 1, pp. 174-194, 2022.

[4] Q. Xu, L. Wang, N. Wang, X. Hei, and L. Zhao, "A review of opposition-based learning from 2005 to 2012," Engineering Applications of Artificial Intelligence, vol. 29, pp. 1-12, 2014.

[5] H. Ming, M. Wang, and X. Liang, "An improved genetic algorithm using opposition-based learning for flexible job-shop scheduling problem," in 2016 2nd International Conference on Cloud Computing and Internet of Things (CCIOT), 2016: IEEE, pp. 8-15.

[6] M. El-Abd, "Generalized opposition-based artificial bee colony algorithm," in 2012 IEEE congress on evolutionary computation, 2012: IEEE, pp. 1-4.

[7] M. Agarwal and G. M. S. Srivastava, "Opposition-based learning inspired particle swarm optimization (OPSO) scheme for task scheduling problem in cloud computing," Journal of Ambient Intelligence and Humanized Computing, vol. 12, no. 10, pp. 9855-9875, 2021.

[8] V. Prakash, V. Vinothina, K. Kalaiselvi, and K. Velusamy, "An improved bacterial colony optimization using opposition-based learning for data clustering," Cluster Computing, vol. 25, no. 6, pp. 4009-4025, 2022.

[9] D. Bairathi and D. Gopalani, "Opposition based salp swarm algorithm for numerical optimization," in Intelligent Systems Design and Applications: 18th International Conference on Intelligent Systems Design and Applications (ISDA 2018) held in Vellore, India, December 6-8, 2018, Volume 2, 2020: Springer, pp. 821-831.

[10] M. Tubishat, N. Idris, L. Shuib, M. A. Abushariah, and S. Mirjalili, "Improved Salp Swarm Algorithm based on opposition based learning and novel local search algorithm for feature selection," Expert Systems with Applications, vol. 145, p. 113122, 2020.

[11] K. Liu, J. Zhou, and D. Dong, "Improving stock price prediction using the long short-term memory model combined with online social networks," Journal of Behavioral and Experimental Finance, vol. 30, p. 100507, 2021.

[12] Y. Wang, L. Wang, F. Yang, W. Di, and Q. Chang, "Advantages of direct input-to-output connections in neural networks: The Elman network for stock index forecasting," Information Sciences, vol. 547, pp. 1066-1079, 2021.

[13] B. Yan and M. Aasma, "A novel deep learning framework: Prediction and analysis of financial time series using CEEMD and LSTM," Expert systems with applications, vol. 159, p. 113609, 2020.

[14] A. Dezhkam, M. T. Manzuri, A. Aghapour, A. Karimi, A. Rabiee, and S. M. Shalmani, "A Bayesian-based classification framework for financial time series trend prediction," The Journal of supercomputing, vol. 79, no. 4, pp. 4622-4659, 2023.
[15] J. Zhao, D. Zeng, S. Liang, H. Kang, and Q. Liu, "Prediction model for stock price trend based on recurrent neural network," Journal of Ambient Intelligence and Humanized Computing, vol. 12, pp. 745-753, 2021.
[16] T. Gao and Y. Chai, "Improving stock closing price prediction using recurrent neural network and technical indicators," Neural computation, vol. 30, no. 10, pp. 2833-2854, 2018.
[17] L. Zheng and H. He, "Share price prediction of aerospace relevant companies with recurrent neural networks based on pca," Expert Systems with Applications, vol. 183, p. 115384, 2021.
[18] Z. Berradi and M. Lazaar, "Integration of principal component analysis and recurrent neural network to forecast the stock price of casablanca stock exchange," Procedia computer science, vol. 148, pp. 55-61, 2019.
[19] F. Yang, J. Chen, and Y. Liu, "Improved and optimized recurrent neural network based on PSO and its application in stock price prediction," Soft computing, vol. 27, no. 6, pp. 3461-3476, 2023.
[20] S. Luo and C. Tian, "Financial high-frequency time series forecasting based on sub-step grid search long short-term memory network," IEEE Access, vol. 8, pp. 203183-203189, 2020.
[21] J. Rasheed, A. Jamil, A. A. Hameed, M. Ilyas, A. Özyavaş, and N. Ajlouni, "Improving stock prediction accuracy using cnn and lstm," in 2020 International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy (ICDABI), 2020: IEEE, pp. 1-5.
[22] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural computation, vol. 9, no. 8, pp. 1735-1780, 1997.
[23] S. Mirjalili, A. H. Gandomi, S. Z. Mirjalili, S. Saremi, H. Faris, and S. M. Mirjalili, "Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems," Advances in engineering software, vol. 114, pp. 163-191, 2017.
[24] H. R. Tizhoosh, "Opposition-based learning: a new scheme for machine intelligence," in International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06), 2005, vol. 1: IEEE, pp. 695-701.
[25] S. Rahnamayan, H. R. Tizhoosh, and M. M. Salama, "Opposition-based differential evolution," IEEE Transactions on Evolutionary computation, vol. 12, no. 1, pp. 64-79, 2008.
[26] O. P. Verma, D. Aggarwal, and T. Patodi, "Opposition and dimensional based modified firefly algorithm," Expert Systems with Applications, vol. 44, pp. 168-176, 2016.
[27] G. Xiong, J. Zhang, D. Shi, and Y. He, "Oppositional Brain Storm Optimization for Fault Section Location in Distribution Networks," in Brain Storm Optimization Algorithms: Springer, 2019, pp. 61-77.
[28] A. A. Ewees, M. Abd Elaziz, and D. Oliva, "A new multi-objective optimization algorithm combined with opposition-based learning," Expert Systems with Applications, vol. 165, p. 113844, 2021.
[29] H. Muthusamy, S. Ravindran, S. Yaacob, and K. Polat, "An improved elephant herding optimization using sine–cosine mechanism and opposition based learning for global optimization problems," Expert Systems with Applications, vol. 172, p. 114607, 2021.
[30] Z. Zhang, Z. Xu, S. Luan, and X. Li, "A Hybrid Max–Min Ant System by Levy Flight and Opposition-Based Learning," International Journal of Pattern Recognition and Artificial Intelligence, p. 2151013, 2021.
[31] K. Kalaiselvi, K. Velusamy, and C. Gomathi, "Financial prediction using back propagation neural networks with opposition based learning," in Journal of Physics: Conference Series, 2018, vol. 1142, no. 1: IOP Publishing, p. 012008.
[32] D. Bairathi and D. Gopalani, "Opposition-based sine cosine algorithm (OSCA) for training feed-forward neural networks," in 2017 13th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS), 2017: IEEE, pp. 438-444.
[33] Z. Hu, Y. Bao, and T. Xiong, "Partial opposition-based adaptive differential evolution algorithms: Evaluation on the CEC 2014 benchmark set for real-parameter optimization," in 2014 IEEE Congress on Evolutionary Computation (CEC), 2014: IEEE, pp. 2259-2265.
[34] T. Fischer and C. Krauss, "Deep learning with long short-term memory networks for financial market predictions," European journal of operational research, vol. 270, no. 2, pp. 654-669, 2018.
[35] H. Chung and K.-s. Shin, "Genetic algorithm-optimized long short-term memory network for stock market prediction," Sustainability, vol. 10, no. 10, p. 3765, 2018.
[36] B. Shao, M. Li, Y. Zhao, and G. Bian, "Nickel price forecast based on the LSTM neural network optimized by the improved PSO algorithm," Mathematical Problems in Engineering, vol. 2019, 2019.
[37] L. Jovanovic et al., "Multi-step crude oil price prediction based on lstm approach tuned by salp swarm algorithm with disputation operator," Sustainability, vol. 14, no. 21, p. 14616, 2022.https://finance.yahoo.com/.