

Stock market prediction using Firefly algorithm with evolutionary framework optimized feature reduction for OSELM method

Smruti Rekha Das ^{a,*}, Debahuti Mishra ^a, Minakhi Rout ^b

^a Department of Computer Science and Engineering, Siksha 'O'Anusandhan Deemed to be University, Bhubaneswar, Odisha, India
^b School of Computer Engineering, KIIT Deemed to be University, Bhubaneswar, Odisha, India



ARTICLE INFO

Article history:

Received 8 April 2019

Revised 13 July 2019

Accepted 29 August 2019

Available online 30 August 2019

Keywords:

Stock market prediction
 Extreme Learning Machine
 Online Sequential Extreme Learning Machine
 Factor analysis
 Firefly Optimization
 Firefly with evolutionary framework

ABSTRACT

Forecasting future trends of the stock market using the historical data is the exigent demand in the field of academia as well as business. This work has explored the feature optimization capacity of firefly with an evolutionary framework considering the biochemical and social aspects of Firefly algorithm, along with the selection procedure of objective value in evolutionary notion. The performance of the proposed model is evaluated using four different stock market datasets, such as BSE Sensex, NSE Sensex, S&P 500 index and FTSE index. The datasets are regenerated using the proper mathematical formulation of the fundamental part belonging to technical analysis, such as technical indicators and statistical measures. The feature reduction through transformation is carried out on the enhanced dataset before employing the experimented dataset to the prediction models such as Extreme Learning Machine (ELM), Online Sequential Extreme Learning Machine (OSELM) and Recurrent Back Propagation Neural Network (RBPNN). For feature reduction, both statistical and optimized based feature reduction strategies are considered, where Principal Component Analysis (PCA) and Factor Analysis (FA) are examined for statistical based feature reduction and Firefly Optimization (FO), Genetic Algorithm (GA) and Firefly algorithm with evolutionary framework are well thought out for optimized feature reduction techniques. An empirical comparison is established among the experimented prediction models considering all the feature reduction techniques for the time horizon of 1 day, 3 days, 5 days, 7 days, 5 days and 30 days in advance, applying on all the datasets used in this study. From the simulation result, it can be clearly figured out that firefly with evolutionary framework optimized feature reduction applying to OSELM prediction model outperformed over the rest experimented models.

© 2019 The Authors. Published by Elsevier Ltd.
 This is an open access article under the CC BY-NC-ND license.
[\(<http://creativecommons.org/licenses/by-nc-nd/4.0/>\)](http://creativecommons.org/licenses/by-nc-nd/4.0/)

1. Introduction

Predicting the future price of the stock market is extremely important for the investors as knowing the appropriate movement of the stock price will reduce the risk for determining future trends so as to invest money. It's a challenging issue to know the future movements due to its highly fluctuating nature, hence, suitable computational methods are required to forecast the stock price movement. Regarding stock market predictability (Baek & Kim, 2018) many controversies have been rising from decades long. Initially, the movement of the stock price was characterized by the random walk theory (Cootner, 1964; Fama, 1965) and later the re-

search on price changes was carried out based on the Efficient Market Hypothesis (EMH) (Malkiel & Fama, 1970). According to their perspective, the future price movement does not depend on the present and past value and also from their point of view future stock price prediction is impossible. In another way, various studies have made effort to experimentally disprove the EMH and the observational evidences have delineated that the stock market can be predicted to some extent. Numerous traditional approaches have been developed by the researchers in the field of forecasting the stock price movements such as Autoregressive Moving Average (ARMA), Autoregressive Integrated Moving Average (ARIMA) etc., but (Abu-Mostafa & Atiya, 1996) these methods are having some limitations, they are not capable to handle the nonlinear relationship exist in time series data as they make the assumption of a linear form of the structure of the model. Furthermore, they are presumed to have constant variance, whereas, the financial time

* Corresponding author.

E-mail addresses: smruti.rekhadas@soa.ac.in (S.R. Das), debahutimishra@soa.ac.in (D. Mishra), minakhi.rout@cs.kiit.ac.in (M. Rout).

series data are very noisy by nature. Later the issues regarding the existence of non linearity in time series data has been solved by the approach of so many machine learning techniques such as an Artificial Neural Network (ANN), Functional Link Artificial Neural Network (FLANN), Support Vector Machine (SVM) etc. In the last decades, numerous researchers have explored the prediction ability of neural networks for financial market prediction.

This study emphasizes on increasing the ability of the prediction model through feature reduction techniques. From the immense literature study it has observed that various kinds of neural networks have been used for stock market prediction. Since the literature study hardly divulges the efficiency of Extreme Learning Machine (ELM), Online Sequential Extreme Learning Machine (OS-ELM) and Recurrent Back Propagation Neural Network (RBPNN) using both statistical and optimized feature reduction techniques in the domain of financial market prediction. In this study, a comparison has made on the outcome of the prediction model considering statistical based feature reduction along with swarm evolutionary based feature reduction technique. The proposed OSELM and firefly with an evolutionary framework strategy model has not used in the financial domain, specifically in stock market prediction and this work basically focused on prediction of the future value of the stock price, which is entirely application oriented and not reported or published yet in the financial domain. An empirical comparison is established among the performance of prediction models using both statistical and optimized feature reduction techniques. For statistical based feature reduction Principal Component Analysis (PCA) and Factor Analysis (FA) and for optimized feature reduction Firefly Optimization (FO), Genetic Algorithm (GA) and Firefly algorithm with evolutionary framework are considered for the experimental work. In the group of heuristic algorithm FO is considered as the most interesting and novel population based heuristic algorithm, which follows the behavior of fireflies. The beauty of firefly is that, each firefly supposed to be attracted (Kuo & Li, 2016; Nhu, Nitsuwat, & Sodani, 2013) by each other firefly irrespective of their sex, here the attraction between the fireflies is directly proportional to their brightness that means the lesser brighter is attracted towards the more brighter one and here in experimental point of view the brightness is decided by calculating the fitness function. On the other hand to generate new fireflies with better performance genetic algorithm is employed on FO, which is termed as Firefly algorithm with evolutionary framework in this study. The Firefly algorithm with evolutionary framework, which is the combination of FO and GA explores the optimal solutions more exactly than individual FO and GA. All these features reduction techniques are applied to the experimented prediction models such as ELM, OSELM and RBPNN. Empirically the model validation is realized over four stock market datasets such as BSE Sensex, NSE Sensex, S&P 500 index and the FTSE index within a given period of time. The dataset is regenerated using technical indicators and statistical measures, based on the available features such as open price, close price, maximum price, minimum price and change of price considering the window size seven. The proposed work provides better prediction ability than the other evaluated learning techniques and prediction models used in this experimental work.

The rest of the paper is organized as follows: Section 2 contains the related work which basically focuses on the promising techniques used by various researchers in financial domain to get best possible results. Section 3 covers the details of data samples and the dataset regeneration part. Section 4 introduces all the predictive models in details and Section 5 covers the particulars of statistical and optimized based feature reduction techniques. Section 6 provides the result analysis, which includes a schematic layout of the proposed prediction model, parameter setup, detailed steps of the proposed algorithm, the analysis of the experimental

outcome in the form of simulation graph and the training result in MSE along with the performance evaluation measures. Finally, Sections 7 and 8 present the overall work analysis and conclusions respectively.

2. Related work

In the development of various prediction models, ANN is realized as a promising technique from the significant contributions of various researchers, basically in the field of financial domain. Predicting future stock price through the application of various neural network technologies such as ANN, Back propagation Neural Network (BPNN), FLANN, ELM etc. has proven to be efficient, making the provision of the days ahead from 1 day to 1month to be predicted. Some researchers have explored ANN for the prediction of (Moghaddam, Moghaddam, & Esfandyari, 2016; Yetis, Kaplan, & Jamshidi, 2014) NASDAQ stock value considering the input data on NASDAQ stock index value. Though, ANN shows its potential for solving the existence of non linearity in financial data, but the random selection of weights may create some non optimal solutions. So, for this issue so many optimization techniques have been developed and applied to ANN by the researchers. Though, ANN is having very distinguished learning ability, but due to the availability of the noise data it may perform very inconsistently and unpredictably. Hence, Kim (2006) proposed GA to optimize the weights between the layers and selecting the task for the relevant instances. Zhang et al. modified bacterial chemotaxis optimization (BCO) (Zhang & Wu, 2009) and named it as Improved BCO and integrated to BPNN for getting the better performance and realized the validation over S&P 500. In addition to this FLANN is efficiently used in the financial domain, as it is less complex and simultaneously able to solve the dynamic and computational complexity problem. Majhi et al. proposed FLANN as a predictor and to train the weight of the prediction (Majhi, Panda & Sahoo, 2009) model least mean square and recursive least square algorithms is used and the efficiency of the model is evaluated by applying over DJIA and S&P 500. To iteratively adjust the parameters of the FLANN, PSO and GA and the gradient descent are used by Naik et al. and compared the (Naik, Nayak & Behera, 2015) result with the result of individuals like FLANN, GA adjusting the parameter of FLANN, FLANN adjusting the parameter by PSO. Tuning of parameter is one of the important issues in the working of NN, many researchers have introduced various learning techniques to tune the parameters. Bebartha et al. projected meta-heuristic Firefly algorithm to train (Bebartha & Venkatesh, 2016) FLANN and compared with FLANN trained with the learning rule of conventional BPNN. In another research (Shen, Guo, Wu, & Wu, 2011) article Artificial Fish Swarm Algorithm (AFSA) has used to learn the RBFNN and compared the result with other widely accepted optimization algorithm such as Particle Swarm Optimization (PSO) and GA. Arévalo et al. introduced a dynamic scheme (Arévalo, García, Guijarro, & Peris, 2017), where the author basically focus on the loss to be stopped and the profit to be taken, which is updated on the basis of quarterly. In addition to this the author also gives emphasis on technical analysis indicator, which is calculated both for 15 min and 1 day time frame. Similarly exploring on time convergence theorem, Xu et al. finally achieved success (Xu, Ke, Xie & Zhou, 2018) on fix time controlling on stock price fluctuation system. More over by analyzing over hybridization model, Jujie and Danfeng (2018) stated hybridized model is always having the best predictive performance compared to single algorithm. Fluctuation leads to rise or fall of the trading market. Hence, based on the concept of rise or fall, bullish or bearish market strategies arises, which define the thought of the trader about future. Tsinaslanidis focused on this concept of bullish and bearish (Tsinaslanidis, 2018), where the author analyzed bearish class predictions, which has

generated on average gives significantly maximum potential profits, whereas the performance of bullish does not give significant profit.

Though, traditional BPNN is having the potential to solve a complex problem, but it may suffer from trapping into local minima and slow convergence speed. Another batch learning algorithm called ELM has developed by Huang, Zhu, and Siew (2006) to train single layer feed forward neural network and the beauty of ELM is that the parameters of hidden layer need not to be tuned at each iteration and the output weights are generated analytically (Huang, Zhou, Ding, & Zhang, 2012; Bueno-Crespo, García-Laencina, & Sancho-Gómez, 2013). For many real applications ELM has shown good generalization performance. Apart from the advantages of ELM, some of the issues regarding random choosing of number of nodes in the hidden layer also arise in ELM. Comparing to other conventional neural network ELM needs more number of hidden layer nodes.

For predicting the improved outcomes feature reduction technique is addressed in so many research articles, as high dimensionality may cause high computational cost at the time of learning the network. Furthermore, most relevant features can also give more accurate results in predicting the outcome. Dimensionality reduction is process, where the number of attributes is reduced and a set of principal attributes is obtained, to get better performance result in the working of the network. Initially statistical based methods were used for the purpose of feature reduction, but now optimized based feature reduction is widely used in machine learning approach. PCA is a statistical technique for the reduction of dimension with the aim of minimizing the loss of variance in the (Jolliffe, 2011; Lam & Lee, 1999) original data. For extracting features, it's a domain of independent technique which can be applied to a wide variety of data. Through the calculation of eigenvectors of the covariance matrix belongs to the original inputs PCA transforms linearly the high dimensional input vector to the low dimensional vector with (Cao, Chua, Chong, Lee, & Gu, 2003) uncorrelated components. There is another statistical data reduction method called FA, which explains about the existence of correlations among the multiple outcomes as a result of one or more number of underlying factors. FA involve in data reduction technique representing a set of variables with a smaller number. Many optimized based feature reduction techniques have been developed by the researchers basically to improve the learning algorithm, feature selection and optimization of the topology. Kim et al. proposed ANN hybridized with (Kim & Han, 2000) GA for feature discretization and feature discretization is associated very closely (Liu & Motoda, 2012) to the dimensionality reduction. GAs is mostly appropriated for higher dimension and the (Painton & Campbell, 1995) stochastic problems with the available nonlinearities and discontinuities. Likewise firefly is one popular algorithm extensively used in numerous area of research. Xiong et al. used Firefly algorithm with (Xiong, Bao & Hu, 2014) support vector regression (SVR) in forecasting stock price index. In addition to this Kazem et al. proposed chaotic firefly (Kazem, Sharifi, Hussain, Saberi & Hussain, 2013) algorithm to optimize the hyperparameters of SVR and evaluated the performance using the stock market index. The result of predicted outcome is compared with the SVR optimized with firefly, where chaotic Firefly algorithms with SVR proved its efficiency over firefly with SVR. Firefly is a population based algorithm work on metheuristic approach, developed by Yang (2010a, 2009) inspired by the behavior of flashing characteristics of fireflies. The flashing light is developed in a way that it is related to the objective (dos Santos Coelho, de Andrade Bernert, & Mariani, 2011) function to be optimized. Simultaneously the concept of chaos theory is added to the optimization algorithm design, as in recent years, researchers have grown their interest on chaos theory and its features for better performance. This Firefly algorithm comes under stochastic algorithm,

which uses the process of randomization at the (Fister, Fister, Yang, & Brest, 2013) time of searching for a set of solutions.

3. Data acquisition

For empirical evaluation of ideas four stock indices such as BSE Sensex, NSE Sensex, S&P 500 index and FTSE index have considered in this study covering the range 24th February 2011 to 27th November 2017 for BSE, 27th October 2000 to 27th November 2017 for NSE, 2nd January 2008 to 16th March 2018 for S&P500 and 2nd January 2008 to 16th March 2018 for FTSE100. The total number of samples collected from the time duration of the respective years is derived through a process of windowing by providing 7 as the window size. The experimentation is carried out for a different prediction time horizon of 1 day, 3 days, 5 days, 7 days, 15 days and 30 days in advance, so accordingly the dataset is reset for the respective time horizon. Out of this total number of sample patterns 70% data are considered for training and 30% data are considered for testing. All the experimented datasets with its details are expressed in Table 1.

The experimental model has used various technical indicators and statistical measures, which transform the price data by adding some additional value and then use this price data as input to the model. The technical indicators and the statistical measures are chosen based on the usage of previous literature (Das, Mishra, & Rout, 2019; Nayak, Mishra, & Rath, 2015) and the combination of available data with technical indicator and statistical measures is able to capture the unsteady properties of financial market. A growing number of technical indicators and statistical measures are available to study, but a few numbers are considered in this study for experimentation are detailed on Table 2.

4. Prediction models

4.1. ELM

ELM works on the principle of Single Layer Feed-forward Neural Network (SLFN), with the basic difference is that unlike SLFN the hidden layer nodes and biases of ELM need not to be tuned at each iteration. In almost all practical learning algorithms of feed forward neural network, it has been marked that the input weights and hidden layer biases (Huang et al., 2006) are needed to be adjusted at each iteration. Comparing to SLFN, the learning speed of ELM is very faster with smallest training (Deo, Sahin, 2015; Huang, Zhu, & Siew, 2004; Sajjadi et al., 2016) error and smallest norm of the weights. For developing ELM, Moore-Penrose generalized inverse takes an important task.

For N number of (Huang et al., 2006) samples, let $x_j = [x_{j1}, x_{j2}, \dots, x_{jn}]^T \in R^n$ are the input to ELM and $t_j = [t_{j1}, t_{j2}, \dots, t_{jm}]^T \in R^m$ are the target output for the respective input. Let \tilde{H} and $f(x)$ is set as the number of hidden nodes and activation function respectively. The input weight vector $w_j = [w_{j1}, w_{j2}, \dots, w_{jn}]^T$ is calculated by connecting input nodes with the j th hidden nodes and the output weight vector $\beta_j = [\beta_{j1}, \beta_{j2}, \dots, \beta_{jm}]$ is generated analytically, which connects j th hidden nodes to the output nodes. Let b_j is the bias of j th hidden node. The equation of ELM can be written as

$$H_m \times \beta = T \quad (1)$$

where H_m is the hidden layer output matrix, β is the output weight matrix and T is the target vector and these three values are calculated by using the Eqs. (1)–(3)

Table 1
Description of data samples and data range.

Datasets	Time duration	Total no. of samples	Training samples	Testing samples
BSE	24/02/2011–27/11/2017	1675	1172	503
NSE	27/10/2000–27/11/2017	4254	2977	1277
S&P 500	02/01/2008–16/03/2018	2630	1841	789
FTSE 100	02/01/2008–16/03/2018	2580	1806	774

Table 2

List of selected technical indicators with statistical measures and their formulas.

Technical Indicators and Statistical Measures	Formula
SMA (Simple Moving Average)	$\frac{\text{Summation of } t \text{ days open price}}{t}$
Del C (Difference between open price)	$(ith + 1)\text{open price} - ith \text{ open price}$
EMA (Exponential Moving Average)	$(\text{Price}_{today} \times SF) + (\text{EMA}_{yesterday} \times (1 - SF))$ $SF \text{ (Smoothing Factor)} = \frac{2}{k+1}, k = \text{the length of the EMA}$
RSI (Relative Strength Index)	$RSI = 100 - \frac{100}{(1+RS^+)}$ where, $RS = \frac{\text{Average of } t \text{ days up opens}}{\text{Average of } t \text{ days down opens}}$ $RSI = \text{If}(Average \ loss = 0, 100, 100 - (\frac{100}{(1+RS^+)})$
ATR (Average True Range)	$ATR_{t-1} = \frac{X(n-1) + TR_t}{n}$ $TR = \text{Max}((Today's \ high - Today's \ low), (Today's \ high - Today's \ open), (Today's \ open - Today's \ low))$
%R (William %R)	$\frac{(Highest \ high - Open)}{(Highest \ high - Lowest \ low)} \times -100$
ROC	$\frac{Close - Close \ n \ periods \ ago}{Close \ n \ periods \ ago} \times 100$
Stochastic Oscillator	$\frac{Closing \ Price - Lowest \ Low}{Highest \ High - Lowest \ Low} \times 100$
ADX	First the directional movement (+DM and -DM) is calculated: $UM = today's \ high \ price - yesterday's \ high \ price$ $DM = yesterday's \ price - today's \ low \ price$ if $UM > DM$ and $UM > 0$, then $+DM = UM$, else $+DM = 0$ if $DM > UM$ and $DM > 0$, then $-DM = DM$, else $-DM = 0$ Number of periods is selected, +DI and -DI are: $+DI = 100$ periods the refined moving average of (+DM) divided by average true range $-DI = 100$ periods the refined moving average of (-DM) divided by average true range $ADX = 100$ times the refined moving average of the absolute value of (+DI - -DI) divided by (+DI + -DI)
Momentum	$\frac{Closing \ Price(P)}{Closing \ Price(P-n) \times 100}$
TRIX	Single – Smoothed EMA = 7 – period EMA of the closing price Double – Smoothed EMA = 7 – period EMA of Single – Smoothed EMA Triple – Smoothed EMA = 7 – period EMA of Double – Smoothed EMA TRIX = 1 – period percent change in Triple Smoothed EMA
Mean	$\frac{1}{N} \sum_{i=1}^N a_i$, The dataset containing the values are a_1, a_2, \dots, a_n .
Variance	$\frac{\sum (a_i - \bar{a})^2}{n-1}$, a_i is a number in the group, \bar{a} is the mean and n is the quantity of numbers in the group
Skewness	$\frac{\sum_{i=1}^N (a_i - \bar{a})^3 / N}{S^3}$, \bar{a} is the mean, s_i is the standard deviation and N is the number of data points
Kurtosis	$\frac{\sum_{i=1}^N (a_i - \bar{a})^4 / N}{S^4} - 3$, \bar{a} is the mean, s_i is the standard deviation and N is the number of data points
Range	Max _range–Min _range, where Max _range = MAX (open, high, low) for a range of t days and Min _range = MIN (open, high, low) for a range of t days.

$$H_m(w_1, \dots, w_{\tilde{H}}, b_1, \dots, b_{\tilde{H}}, x_1, \dots, x_{\tilde{H}}) \\ = \begin{bmatrix} f(w_1 \cdot x_1 + b_1) & \dots & f(w_{\tilde{H}} \cdot x_1 + b_{\tilde{H}}) \\ \vdots & \ddots & \vdots \\ f(w_1 \cdot x_N + b_1) & \dots & f(w_{\tilde{H}} \cdot x_N + b_{\tilde{H}}) \end{bmatrix} \quad (2)$$

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_{\tilde{H}}^T \end{bmatrix}_{\tilde{H} \times m} \quad \text{and} \quad T = \begin{bmatrix} t_1^T \\ \vdots \\ t_{\tilde{H}}^T \end{bmatrix}_{N \times \tilde{H}} \quad (3)$$

The choice of w_j and b_j is random, which is important for (Wang, Cao, & Yuan, 2011) accelerating the rate of learning. β is calculated by using the Moore-Penrose generalized inverse of

H_m . The beauty of ELM is that it avoid multiple iterations as the (Ding, Zhao, Zhang, Xu, & Nie, 2015) total number of training inputs is trained at a time and also it is having the better generalization ability with faster learning rate.

4.2. OSELM

From the batch learning approach of ELM, OSELM is generated (Huang & Siew, 2004; Huang et al., 2004; Liang, Huang, Saratchandran, & Sundararajan, 2006). OSELM can learn the training data in both the ways such as one-by-one and chunk by chunk (Liang et al., 2006; Rong, Huang, Sundararajan, & Saratchandran, 2009) and the size of the chunk can be fixed or can be varied. The training observations are discarded as soon as the learning procedure is

completed. Unlike ELM, at any time, only the newly arrived observations that may be single or chunk is sequentially learned in O-ELM. Here, the learning algorithm does not have any prior knowledge about the number of training observations. The major difference between batch ELM and OSELM is that in batch ELM all the training data available at the time of training, whereas in real application training data may arrive one by one or chunk by chunk, hence this problem can be solved through the online sequential process (Liang et al., 2006).

Let M_0 is set as the chunk of an initial set of total N training samples

$$M_0 = \left\{ (x_j, t_j) \right\}_{j=1}^{M_0} \quad (4)$$

where the number of hidden layer nodes H_n is less than M_0 . Considering the batch algorithm, it needs to minimize $H_0\beta - T_0$, where

$$H_0 = \begin{bmatrix} f(w_1 \cdot x_1 + b_1) & \dots & f(w_{H_n} \cdot x_1 + b_{H_n}) \\ \vdots & \ddots & \vdots \\ f(w_1 \cdot x_{M_0} + b_1) & \dots & f(w_{H_n} \cdot x_{M_0} + b_{H_n}) \end{bmatrix}_{M_0 \times H_n} \quad (5)$$

$$T_0 = \begin{bmatrix} t_1^T \\ \vdots \\ t_{M_0}^T \end{bmatrix}_{M_0 \times m} \quad (6)$$

For minimizing $H_0\beta - T_0$, the solution is given by

$$\beta^{(0)} = K_0^{-1} \times H_0^T \times T_0, \text{ where } K_0 = H_0^T \times H_0 \quad (7)$$

Let M_1 is another chunk of data of total N training samples

$$M_1 = \left\{ (x_j, t_j) \right\}_{j=M_0+1}^{M_0+M_1} \quad (8)$$

Again to minimize

$$\begin{bmatrix} H_0 \\ H_1 \end{bmatrix} \times \beta - \begin{bmatrix} T_0 \\ T_1 \end{bmatrix} \quad (9)$$

The output β is calculated by considering both the training datasets such as M_0 and M_1 ,

$$\beta^{(1)} = K_1^{-1} \times \begin{bmatrix} H_0 \\ H_1 \end{bmatrix}^T \times \begin{bmatrix} T_0 \\ T_1 \end{bmatrix} \quad (10)$$

where

$$K_1 = \begin{bmatrix} H_0 \\ H_1 \end{bmatrix}^T \times \begin{bmatrix} H_0 \\ H_1 \end{bmatrix} \quad (11)$$

By solving the Eq. (11), it can observed that

$$K_1 = K_0 + H_1^T \times H_1 \quad (12)$$

H_1 and T_1 is calculated as follows

$$H_1 = \begin{bmatrix} f(w_1 \cdot x_{M_0+1} + b_1) & \dots & f(w_{H_n} \cdot x_{M_0+1} + b_{H_n}) \\ \vdots & \ddots & \vdots \\ f(w_1 \cdot x_{M_0+M_1} + b_1) & \dots & f(w_{H_n} \cdot x_{M_0+M_1} + b_{H_n}) \end{bmatrix}_{M_1 \times H_n} \quad (13)$$

$$T_1 = \begin{bmatrix} t_{M_0+1}^T \\ \vdots \\ t_{M_0+M_1}^T \end{bmatrix}_{M_1 \times m} \quad (14)$$

Eq. (10) is derived and rewritten as

$$\beta^{(1)} = \beta^{(0)} + K_1^{-1} \times H_1^T \times (T_1 - H_1 \times \beta^{(0)}) \quad (15)$$

The equation can be generalized and represented in the form of $(K+1)$ th chunk data, Let it be M_{K+1} chunk of data and the β value is calculated

$$\beta^{(K+1)} = \beta^{(K)} + K_{K+1}^{-1} \times H_{K+1}^T \times (T_{K+1} - H_{K+1} \times \beta^{(K)}) \quad (16)$$

4.3. RBPNN

RBPNN is having an activation feedback with the short term memory, where the hidden (Sakim & Mustaffa, 2014) layer is updated by both external input and activation from the previous forward propagation. Using some set of weights the feedback is modified so as to enable the usual adaption through the network learning.

5. Feature reduction techniques

When the data of all the features are given to the prediction model, the prediction models, here in this study are neural network models, cannot be well trained properly. Hence, the feature reduction technique is introduced in this study to get the most relevant features of original datasets, for improving the performance of the model. Both statistical based and optimized based feature reduction methods are employed to the dataset. The details of which are described in this section.

5.1. Statistical based feature reduction

5.1.1. PCA

PCA used in multivariate data analysis as a standard tool for reducing the dimension. The basic aim is to consider the reduced number of data variables keeping the intact (Jolliffe, 2011) information available in the original data. As it requires only the eigen decomposition or singular decomposition, hence it is most popular in the perception of computation. To obtain the PCA solution, two alternative optimization approach exists (Xanthopoulos, Pardalos, & Trafalis, 2013) such as variance maximization and minimum error formulation. For feature extraction, PCA is a very prominent method, which calculate the eigenvectors of the covariance matrix belongs to the original inputs and transforms linearly the high (Cao et al., 2003) dimension input vector to low dimension with uncorrelated components. The step wise procedure of PCA is given below (Giri et al., 2013):

Step-1: The covariance matrix of order $M \times M$ is calculated for M dimensional signal samples.

$$\Sigma = \frac{1}{M} \left\{ (X_s - \bar{X}_s)(X_s - \bar{X}_s)^T \right\} \quad (17)$$

where X_s is the signal matrix of D points, each of which having M dimension and \bar{X}_s is denoted as the mean vector.

Step-2: Find out Eigen vectors and eigen values using the covariance matrix Σ by the given Eq. (18).

$$E_v^{-1} \sum E_v = D_E \quad (18)$$

where E_v matrix of Eigen vectors, and D_E is the matrix, whose diagonal elements are the eigen values of covariance matrix Σ

Step-3: Eigen vectors are sorted in the descending order of eigen values in D .

Step-4: The data is projected in the direction of Eigen vectors taking into account the dot product between the given data and the Eigen vectors.

Step-5: Through the restriction of a given percentage of variability depending on the given problem, first few components are selected.

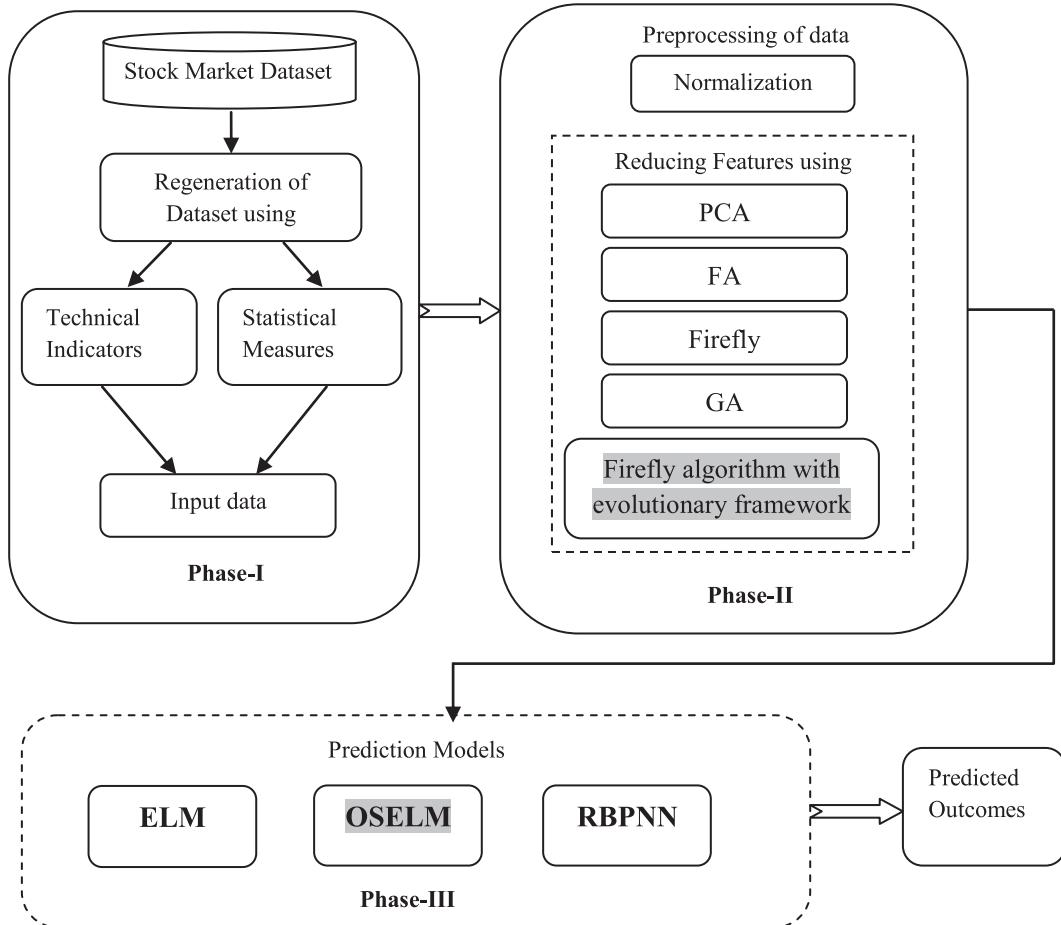


Fig. 1. Schematic layout of proposed prediction model.

5.1.2. FA

FA accepts the observed variables are the linear combination of unobservable factors and among these factors some are common to two or more variables and some expect unique to each variable. The factors which are unique do not contribute to the covariance between the variables or it (Kim & Mueller, 1978) can be said among the observed variables the common factors, which are smaller in number than the observed variable numbers contribute to the covariation. The basic difference of PCA from FA is that, in PCA, the principal components are having certain mathematical functions of the observed variables, while the combinations of observed variables is not able to express the common factors. Whereas, the main objective of FA is to determine the common factors which can produce the correlations among the observed variables satisfactorily.

5.2. Optimized based feature reduction

5.2.1. GA

GA is an evolutionary algorithm based on Darwinian rules of (Villegas, 2007) evolutionary dynamics and can be used efficiently to solve problems with many possible solutions. Initially, the random population is generated and then through the process of crossover, mutation and survival of the fittest, the individuals are selected to go to (Jarvis & Goodacre, 2004) the next generation until the particular stopping condition is satisfied. The following steps are the evolutionary stage of GA:

Step-1: From the parent population the fittest individuals are extracted.

Step-2: The selected off springs are recombined using the crossover operation.

Step-3: The populations after crossover are mutated.

Step-4: The newly generated off springs are passed to evaluate the fitness.

Step-5: The worst parents are replaced with the new off springs and reinserted into the population.

Step-6: The process is repeated until the stopping condition is satisfied.

5.2.2. FO technique

FO is a metaheuristic algorithm, developed by Xin-Shi Yang, based on the flashing characteristics of fireflies. The flashing characteristics can be idealized (Arora & Singh, 2013; Gandomi, Yang, Talatahari & Alavi, 2013; Yang, 2010b,c) by using these following rules:

- (1) All the fireflies are unisex and attracted towards each other despite of their sex.
- (2) The attraction between the two fireflies is directly proportional to their brightness and for any two flashing fireflies always the lesser bright is attracted towards the brighter one. If for a particular firefly their no longer brighter firefly is available then that particular firefly moves randomly in the space.
- (3) The brightness or we can say the light intensity is determined by the landscape of the objective function. Brightness of Firefly can also be defined in the way of fitness function of GA is defined.

Basically the light intensity variation and attractiveness formulation are the two important issues in Firefly algorithm. Simply we can say the attractiveness is determined by the brightness or the intensity of the light, which is successively associated with the objective function. The computational complexity of the (Lukasik & Zak, 2009) algorithm is considered to be $O(m^2)$ but the large population size may lead to substantial increase in the time complexity.

6. Result analysis

This section covers the details of experimental outcome through various steps. At first the schematic layout of the proposed model along with its step wise description is analyzed. Further the appropriate parameter value set for the techniques used in this experimental work is studied. Then the step wise description of the proposed algorithm is designed. The simulation result is delineated in the form of actual versus predicted graph along with the Mean Square Error (MSE) during the training phase. In addition to this the performance evaluation part is figured out, considering the test outcome of the prediction models through different performance measures.

6.1. The system framework

In the process of planning to take decision for investing money in financial markets, prediction is a powerful tool. Three prediction models such as ELM, OSELM and RBPNN are considered in this study and the efficiency of the prediction models are evaluated by four stock market datasets, BSE Sensex, NSE Sensex, S&P 500 index and FTSE index. The framework of the proposed prediction model is shown in Fig. 1. The experimental work has done phase wise; in the first phase the dataset has regenerated, where the regeneration occurs by adding some new features such as technical indicators and statistical measures to the existing stock market datasets. The dimension of the input has increased with the expectation of getting closer predicted value towards the actual value. The technical indicators and statistical measures are generated by using the mathematical formula taking into account 7 as the window size. In the second phase, data preprocessing occurs that involves normalization and feature reduction. For normalization, min-max normalization is well thought-out in this study, which scales the data between 0 and 1. In case of feature reduction two types of feature reduction occurs, one is statistical based feature reduction and another one is optimized based feature reduction. PCA and FA are used for statistical based feature reduction whereas GA, FO and Firefly algorithm with evolutionary framework are designed for optimized feature reduction. Now the reduced data are ready for input to ELM, OSELM and RBPNN prediction models.

In optimized based feature reduction GA, FO and Firefly algorithm with evolutionary framework are considered for reducing features before giving input to the models. FO is an interesting population based heuristic algorithm, where each brighter firefly attracts its partner in spite of their sex, which help to explore the search space efficiently. But to explore the search space more precisely for getting optimal solution, GA is added to Firefly, which is designed as Firefly algorithm with (Rahmani & MirHassani, 2014) evolutionary framework is designed. The reduced features obtained from of all the feature reduction techniques are individually set input to the prediction models. An empirical comparison is carried among all the prediction models for all types of feature reduction techniques using all the datasets. From the simulation result of all the experimental work, the proposed OSELM based on Firefly Algorithm with evolutionary framework optimized feature reduction model proves its efficiency over the rest of the combinations of feature reduction approach with prediction models.

6.2. Parameter setup

This section discussed the parameters of all the experimented methods one by one. ELM works on the principle of SLFN, hence it has one hidden layer. The number of nodes in the hidden layer is considered as 15 and the execution is carried out for 100 number of iterations. Apart from the parameters of ELM, OSELM is having one extra parameter such as the number of batches. This study considered 30 numbers of batches to get better predicted outcome. Similarly, for the execution of RBPNN, this experimental work set threshold value and learning rate to 0.1 and momentum coefficient to 0.05. Further the parameters of two optimized methods such as GA and FO is analyzed. GA is found to have two basic parameters, one is crossover rate and another is mutation rate. The parameter value 0.8 for crossover rate and 0.1 for mutation rate is most optimal according to the convergence time. In addition to this, the basic parameters of FO, which controls the algorithm are noticed. They are attraction coefficient and absorption coefficient. The appropriate values for these two parameters has to be chosen correctly to get optimal solutions. This study set attraction coefficient (β_0) to 1.0 and absorption coefficient (γ) to 0.8.

Apart from the individual algorithm specific parameters, population size and number of iterations are two common controlling parameters, which are kept fixed for all the experimentation of this work. Population size has a great role in determining the optimal solutions as a small population may miss the part of the solution space and the large populations may increase the computational time, hence a 100 number of populations is considered for a standard comparison. In addition to this, number of iterations is another parameter which effect the running time to get a good solution, hence number of iterations is also fixed to 100 for all the experimental work.

6.3. Detailed steps of stock exchange prediction using Firefly algorithm and Firefly algorithm with evolutionary framework applying on OSELM prediction model

Algorithm: Firefly algorithm.

Input: Training Data, Training Labels

Output: Transformation Matrix

Begin

1. Initialize population of firefly P_i for $i = 1, 2, \dots, n$
2. Find objective value O_i for each population P_i by function $f()$
3. Define absorption coefficient γ
4. Define attraction coefficient β_0
5. Find best population $Best_P$ and best cost $Best_O$ by finding minimum of O_i
6. While maximum iteration not reached do:
7. For each P_i do
8. For each P_j where $P_i \neq P_j$ do
9. If $O_j < O_i$ then do
10. $r_{ij} = (\frac{1}{d} \sum_{k=1}^d (P_j^k - P_i^k)^2)$
11. $\beta = \beta_0 \cdot e^{-\gamma \cdot r_{ij}}$
12. $New_P_i^k = P_i + \beta \cdot rand() \cdot (P_j^k - P_i^k)$ for $k = 1, 2, \dots, d$
13. $New_O_i = f(New_P_i)$
14. If $New_O_i < O_i$
- Replace P_i with New_P_i
16. If $New_O_i < Best_O$
- Replace $Best_P$ with New_P_i
18. End If
19. End If
20. End If
21. End For
22. End For
23. End While
24. Return $Best_P$

Description: The best population, which is obtained out of this FO technique is the transformation matrix. This transformation matrix is multiplied to the input data, by which the reduced fea-

tures are obtained. Instead of all the data only the data of those reduced features are set input to the OSELM prediction model.

Algorithm: Firefly algorithm with evolutionary framework.

Input: Training Data, Training Labels

Output: Transformation Matrix

Begin

1. Initialize population of firefly P_i for $i = 1, 2, \dots, n$
2. Find objective value O_i for each population P_i by function $f(\cdot)$
3. Define absorption coefficient γ
4. Define attraction coefficient β_0
5. Find best population $Best_P$ and best cost $Best_O$ by finding minimum of O_i
6. While maximum iteration not reached do:
7. For each P_i do
8. For each P_j where $P_i \neq P_j$ do
9. If $O_j < O_i$ then do
10. $r_{ij} = (\frac{1}{d} \sum_{k=1}^d (P_j^k - P_i^k)^2)$
11. $\beta = \beta_0 e^{-\gamma r_{ij}}$
12. $New_P_j^k = P_j + \beta \cdot rand() \cdot (P_j^k - P_i^k)$ for $k = 1, 2, \dots, d$
13. $New_O_j = f(New_P_j)$
14. If $New_O_j < O_i$
15. Replace P_i with New_P_j
16. If $New_O_j < Best_O$
17. Replace $Best_P$ with New_P_j
18. End If
19. End If
20. End If
21. End For
22. End For

23. Addition of evolutionary framework concept

24. Rank all the population and find the current best populations
25. Cross over mechanism is applied over two current best populations and two best feasible solutions is selected out of the parent and children according to their fitness value using the fitness function().
26. One population is randomly selected for mutation operation and then it is replaced
27. End While
28. Return $Best_P$

Description: Some improvements are added here to this FO technique (Rahmani & MirHassani, 2014) for transforming the current solutions to some enhanced solutions. The concept of GA is added to the updated solutions moving towards the best solutions, and for this the basic operation of GA such as selection, crossover and mutation is carried out. The basic reason behind this modification is ability to find good solutions in the reasonable time and avoiding of trapping in local minima. Hence, the two current best populations or in the concept of firefly, two populations of higher light intensity are selected through an attractiveness process. Crossover mechanism is applied over these two current best populations and two best solutions are selected based on their fitness value using the fitness function, eliminating the rest among the parent and children feasible solutions. In the next step, the mutation operation is performed by choosing one random population. If the new solution or new firefly is feasible and giving better solution through the functional value then the new one replaced the old solution. The process is continued till the termination condition is satisfied. Through, this improve method a best solution is generated which is a transformation matrix. The above transformation matrix is multiplied to the input data, through, which the reduced features are obtained. Now the data of these reduced features are considered as input to the OSELM prediction model.

6.4. Experimental outcome analysis

Four datasets such as BSE Sensex, NSE Sensex, S&P 500 index and FTSE index are used to explore the forecasting ability of the prediction models. Each dataset is divided into training and testing in the ratio of 7:3. The actual versus predicted graph of all the experimented datasets using Firefly algorithm with an evolutionary framework applying on ELM, OSELM and RBPNN for 1 day ahead closing price is depicted in Fig. 2, Fig. 3 and Fig. 4 respectively.

From these figures it can be clearly figured out that the performance of RBPNN in predicting the above mentioned stock value is poorer than ELM and OSELM.

The simulated graph of ELM, OSELM and RBPNN for BSE Sensex data considering six time horizon such as 1 day, 3 days, 5 days, 7 days, 15 days and 30 days ahead to be predicted is depicted in Figs. 5–7, where PCA feature reduction technique is set for reducing feature before giving stock data input to the prediction model. In the experimental outcome though all the simulated graph of actual versus predicted is achieved but due to lack of space, in some experimentation 1 day, 5 days and 15 days ahead is given, whereas in other experimentation 3 days, 7 days and 30 days ahead prediction result has shown but the information can be retrieved from the table through MSE result during training. Similarly, Fig. 8, Fig. 9 and Fig. 10 shows the actual versus predicted graph of the same data using FA feature reduction technique applying on ELM, OSELM and RBPNN respectively. Further the actual versus predicted graph of ELM, OSELM and RBPNN based on GA is pictured in Fig. 11, Fig. 12 and Fig. 13 respectively. From the figure it can be visualized that the predicted graph of GA is more closer to actual than the statistical based feature reduction method. The same result is reflected in the actual versus predicted graph presented in Fig. 14, Fig. 15 and Fig. 16, for firefly optimized feature reduction techniques applied on ELM, OSELM and RBPNN respectively. Moreover, Fig. 17, Fig. 18 and Fig. 19 represents the actual versus predicted graph of Firefly algorithm with evolutionary framework for the prediction models ELM, OSELM and RBPNN respectively, which shows the predicted graph of all the datasets considered here in this study with respect to the experimented time horizon is more closer to the actual graph in comparison to the rest of the feature reduction techniques. The experimental work covers the six different time horizon such as 1 day, 3 days, 5 days, 7 days, 15 days and 30 days of BSE, NSE, S&P 500 and FTSE dataset, but due to the limitation on space only the actual verses predicted graph of BSE for 1 day to 30 days time horizon is given in this study. The total result of all the prediction models reducing features with the statistical as well as optimized feature reduction techniques for BSE, NSE, S&P 500 and FTSE datasets is shown in the form of MSE during the training phase, which is given in the respective tables.

The experimental results obtained from the MSE during the training phase of PCA based ELM, OSELM and RBPNN for 1 day, 3 days, 5 days, 7 days, 15 days and 30 days ahead to be predicted using BSE, NSE, FTSE and S&P500 dataset are given in the Table 3, Table 4 and Table 5 respectively. The most notable result that we observed from the above tables is that, for 1 day ahead prediction, PCA-ELM is showing minimum error than the rest of days ahead to be predicted. As the number of days ahead prediction increases the performance of the model is significantly decreases. The above result helps to visualize that, through this model short term investment is more predictable than long term investment. Furthermore, the MSE result of PCA-OSELM exhibit that BSE and FTSE during 3 days, NSE during 5 days and S&P 500 during 30 days ahead to be predicted is outperformed over the rest time horizon for the respective datasets. In addition to this, the MSE result of PCA-RBPNN is analyzed, which make to know that the performance of RBPNN is worse for all the time horizon of all the experimented dataset.

Table 6–8 shows the prediction accuracy for the same dataset during the training period considering the time horizon of 1 day, 3 days, 5 days, 7 days, 15 days and 30 days ahead. For all the datasets it can be observed that like PCA-ELM, FA-ELM is showing better performance for 1 day ahead prediction than the rest of the days ahead considered here in this experimental work. In addition to this, the MSE result of FA-OSELM is analyzed and found that, for BSE prediction the above model is showing better performance for 1 day, for NSE the performance is better for 7 days whereas for FTSE and S&P500 the model is giving better prediction accuracy

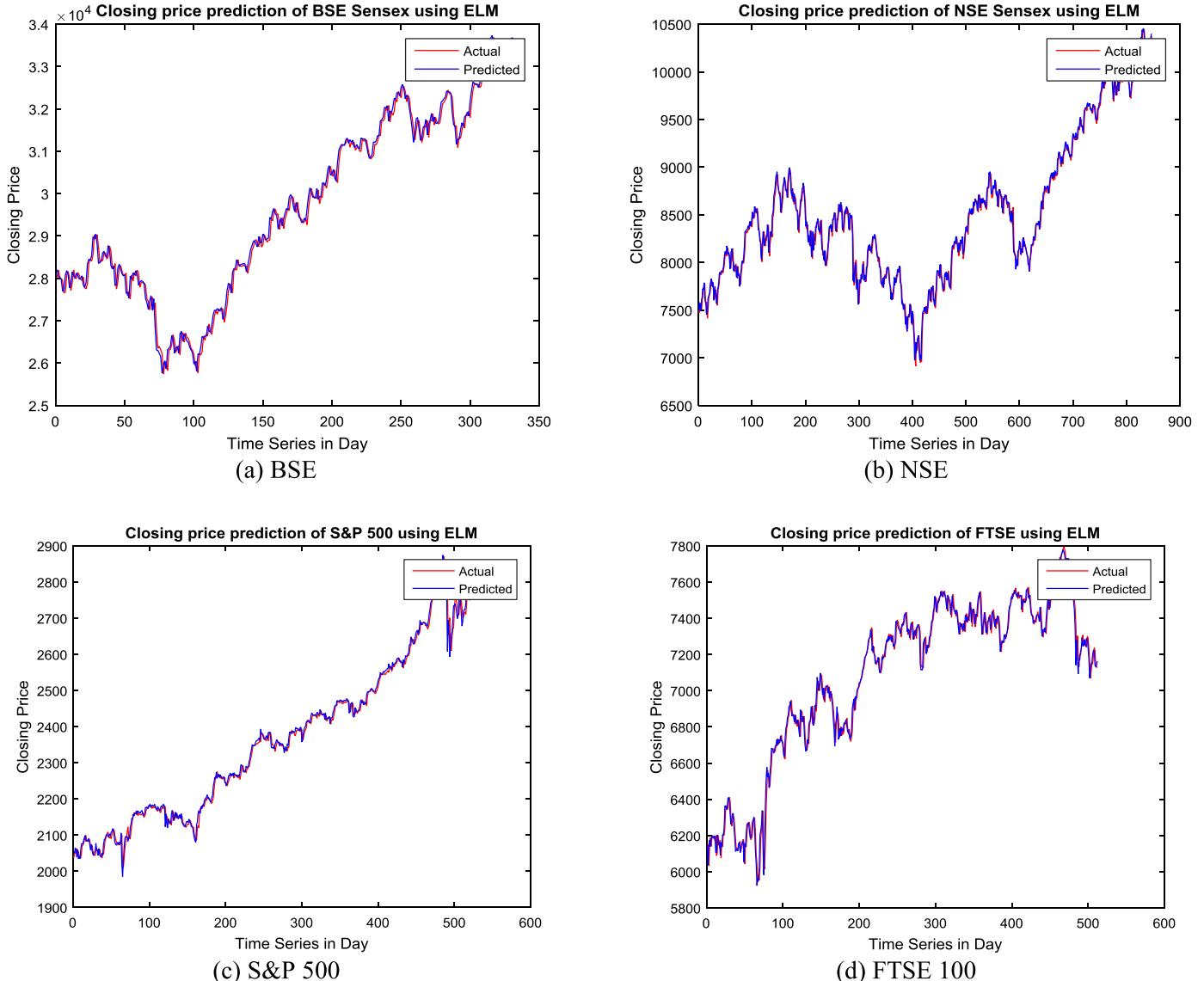


Fig. 2. Actual versus predicted graph of (a) BSE, (b) NSE, (c) S&P 500 index and (d) FTSE 100 index using Firefly algorithm with evolutionary framework applying on ELM for 1 day ahead closing price.

Table 3
Day(s) ahead prediction of BSE, NSE, FTSE and S&P Sensex data using PCA based ELM.

Datasets	1 day	3 days	5 days	7 days	15 days	30 days
BSE	0.018242	0.031889	0.060005	0.068082	0.16917	0.36839
NSE	0.03401	0.069061	0.083373	0.36479	1.7093	0.13337
FTSE	0.022097	0.10006	0.8234	0.31291	0.26579	1.1088
S&P	0.18793	0.40343	0.22553	0.74967	1.161	0.97085

on 3 days ahead prediction. Further, FA-RBPNN, which is used for a comparison, as it is a widely accepted model, is showing the worst performance for all the datasets irrespective of the time horizon.

The ability of the firefly optimized feature reduction for training the prediction models are shown in Tables 9–11. According to the MSE result of the above model, it can be seen that, firefly based feature reduction well performed in 1 day ahead prediction for ELM and OSELM and predicting better compared to the statistical based feature reduction method, in the regard of minimum MSE value. The MSE result of ELM and OSELM is comparatively less than the PCA and FA irrespective of the different experimented time horizon, applicable for all the datasets. The Firefly

algorithm is very efficient in generating optimized features, which implies that its success rate to find the global optima is more than the statistical based feature reduction techniques. Though RBPNN result is worse compared to ELM and OSELM, but for a comparison purpose, this study has included RBP neural network model, which is a bench mark model to the analysis of financial market forecasting.

The training phase result of GA in the term of MSE is illustrated in Tables 12–14. The performance of ELM and OSELM using GA as feature reduction is better for one day ahead prediction, same as FO, than the rest of time horizon ahead to be predicted. According to the result, if we consider the MSE of one day ahead, for BSE,

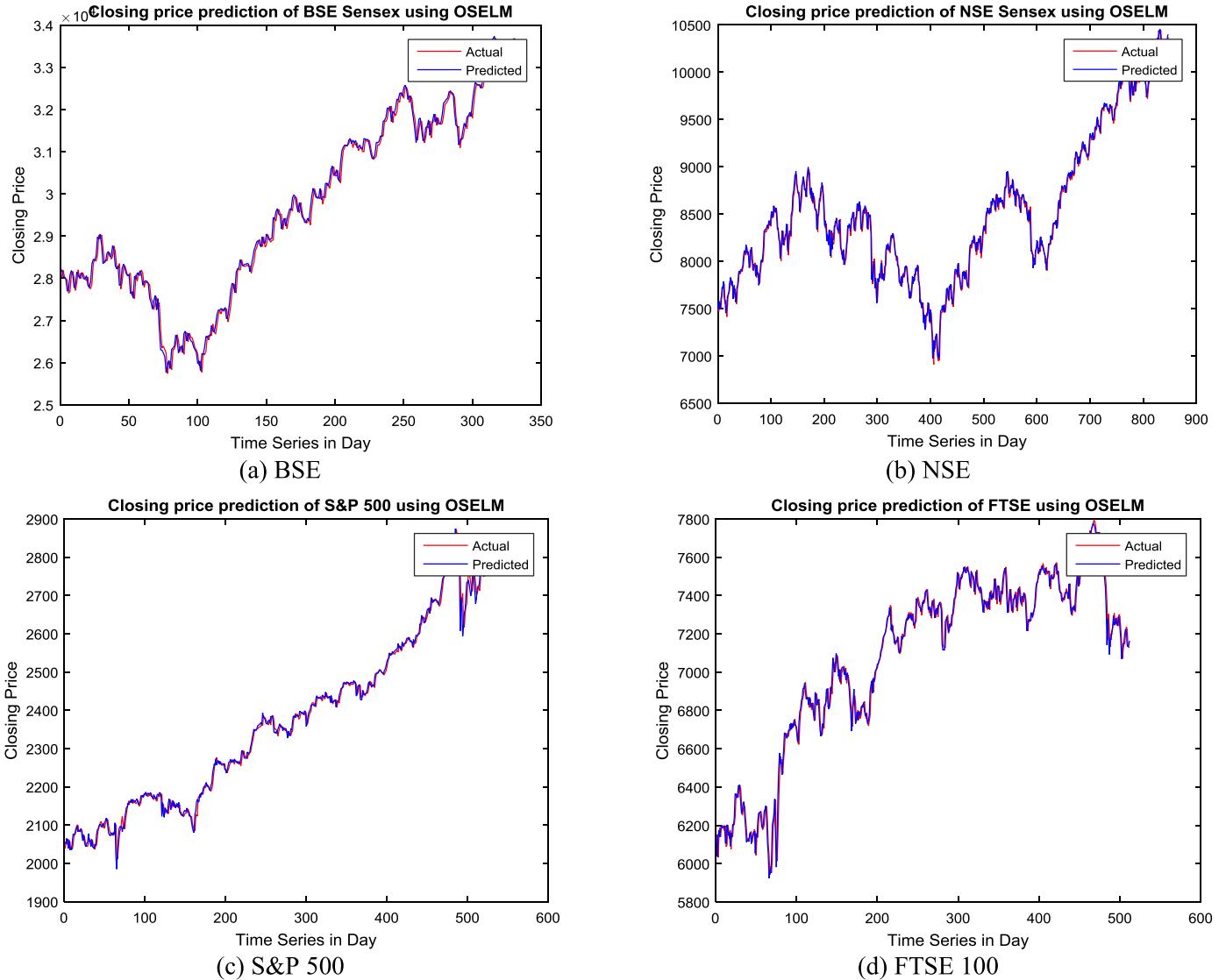


Fig. 3. Actual versus predicted graph of (a) BSE, (b) NSE, (c) S&P 500 and (d) FTSE 100 index using Firefly algorithm with evolutionary framework applying on OSELM for 1 day ahead closing price.

Table 4
Day(s) ahead prediction of BSE, NSE, FTSE and S&P Sensex data using PCA based OSELM.

Datasets	1 day	3 days	5 days	7 days	15 days	30 days
BSE	0.073453	0.0348	0.53266	0.16099	2.2145	0.47495
NSE	0.078005	0.06153	0.061084	0.077821	0.16177	0.2826
FTSE	0.82785	0.095044	0.53815	0.16421	1.6874	1.2882
S&P	0.52407	0.15101	0.28387	1.2051	0.25584	0.10085

Table 5
Day(s)n ahead prediction of BSE, NSE, FTSE and S&P Sensex data using PCA based RBPNN.

Datasets	1 day	3 days	5 days	7 days	15 days	30 days
BSE	491,394,731.4709	491,831,971.0308	492,029,873.7347	492,215,198.5763	493,363,724.1234	495,578,973.7223
NSE	15,197,731.4568	15,217,297.2463	15,225,618.3892	15,245,545.8347	15,290,324.9568	15,387,721.0519
FTSE	1.1351	32,197,152.1171	32,195,425.1182	32,195,451.7851	32,199,817.6513	32,205,589.4047
S&P	1,754,970.9799	1,756,108.4377	1,756,499.8391	1,756,949.4891	1,760,258.6556	1,766,095.359

Table 6
Day(s) ahead prediction of BSE, NSE, FTSE and S&P Sensex data using FA based ELM.

Datasets	1 day	3 days	5 days	7 days	15 days	30 days
BSE	0.034941	0.093144	0.072128	0.17093	0.36457	0.36776
NSE	0.04129	0.087231	0.19352	0.39221	2.07512	0.31934
FTSE	0.03291	0.21003	0.89314	0.41983	0.78152	2.92611
S&P	0.19412	0.56219	0.39812	0.78229	2.62131	1.39812

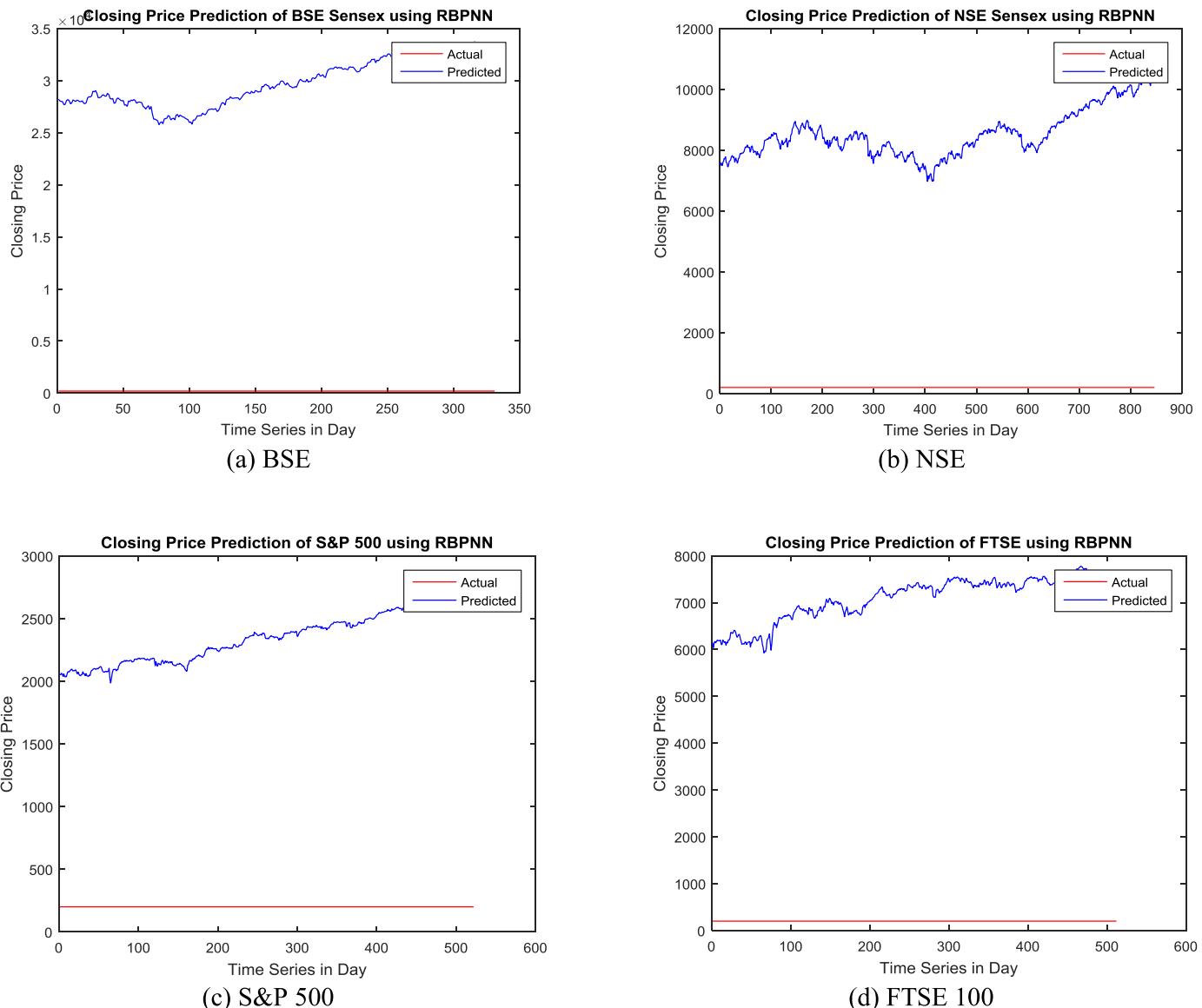


Fig. 4. Actual versus predicted graph of (a) BSE, (b) NSE, (c) S&P 500 and (d) FTSE 100 index using Firefly algorithm with evolutionary framework applying on RBPNN for 1 day ahead closing price.

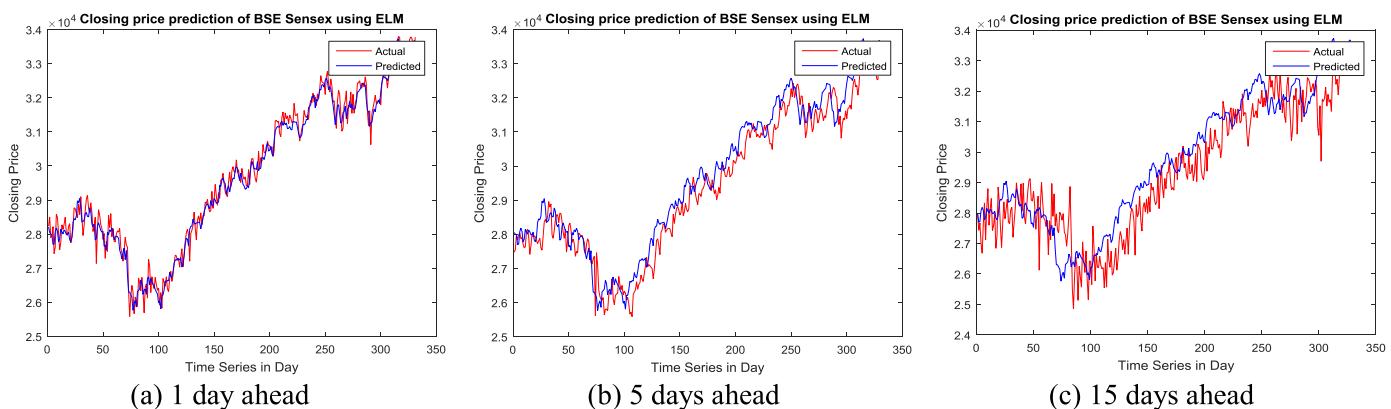


Fig. 5. Closing price prediction of BSE Sensex using PCA based on ELM for 1 day, 5 days and 15 days ahead.

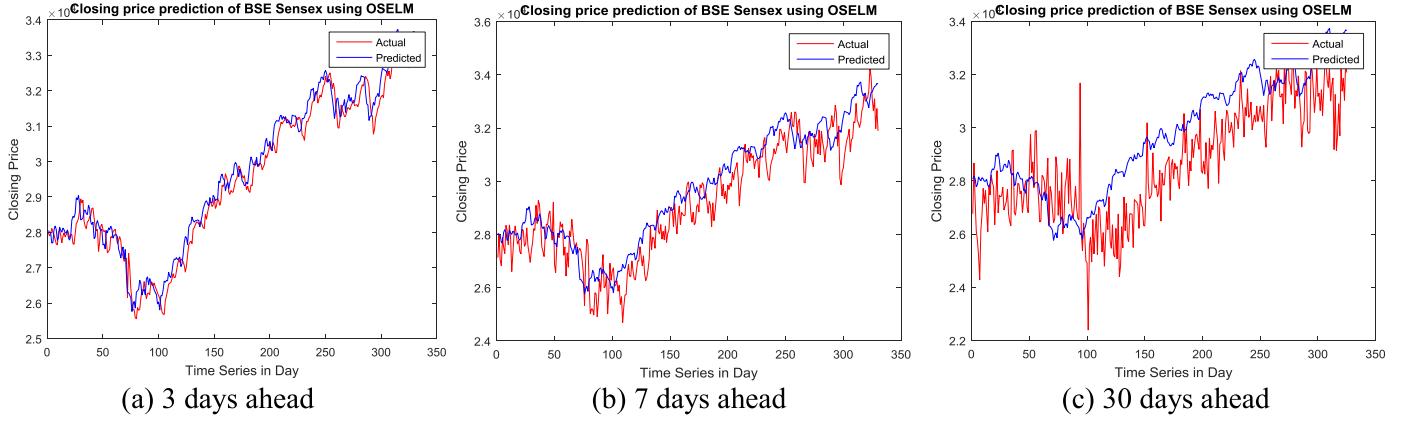


Fig. 6. Closing price prediction of BSE Sensex using PCA based on OSELM for 3 days, 7 days and 30 days ahead.

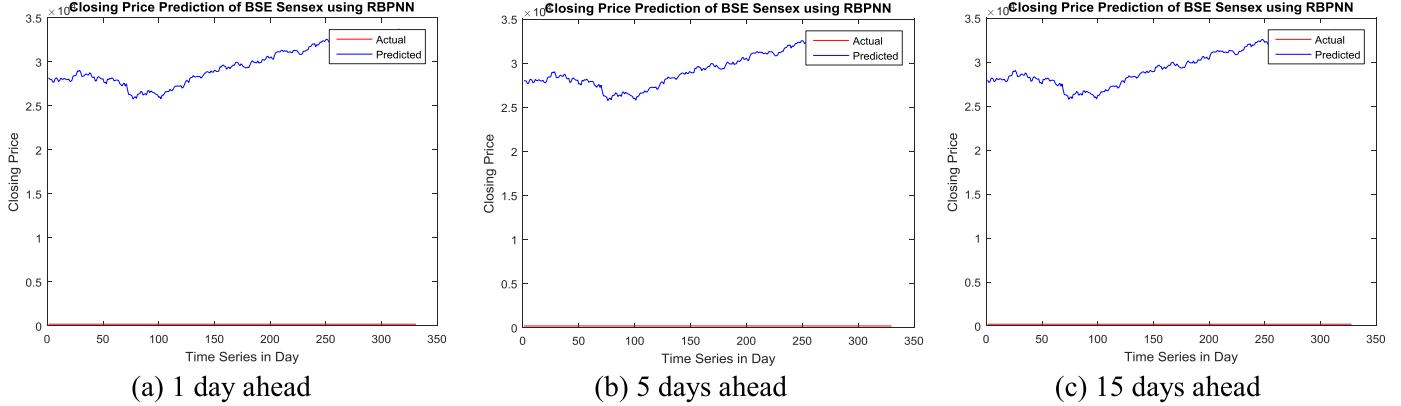


Fig. 7. Closing price prediction of BSE Sensex using PCA based on RBPNN for 1 day, 5 days and 15 days ahead.

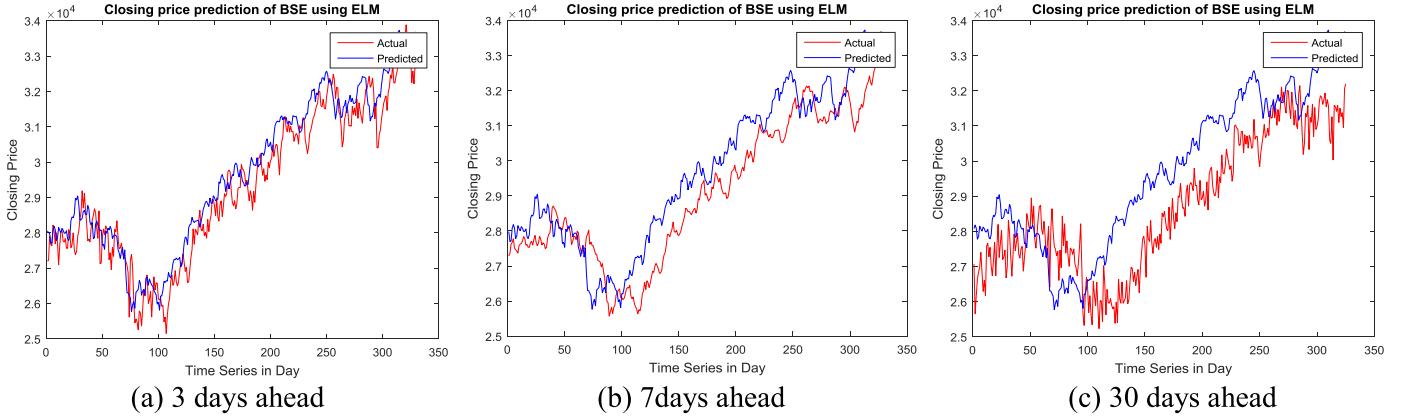


Fig. 8. Closing price prediction of BSE Sensex using FA based on ELM for 3 days, 7 days and 30 days ahead.

Table 7
Day(s) ahead prediction of BSE, NSE, FTSE and S&P Sensex data using FA based OSELM.

Datasets	1 day	3 days	5 days	7 days	15 days	30 days
BSE	0.033134	0.41221	0.17489	0.17232	0.37286	0.78972
NSE	0.082611	0.09271	0.17824	0.09245	0.62871	1.54287
FTSE	0.88254	0.18725	0.591733	0.42871	2.78453	1.09153
S&P	0.43271	0.12987	0.49812	1.87231	0.65211	1.89243

Table 8
Day(s) ahead prediction of BSE, NSE, FTSE and S&P Sensex data using FA based RBPNN.

Datasets	1 day	3 days	5 days	7 days	15 days	30 days
BSE	491,831,970.9217	492,029,874.0208	492,215,198.9369	493,363,724.0737	495,578,973.5507	495,578,973.2484
NSE	19,895,292.4845	17,345,176.6342	187,453,215.8945	18,734,519.987	187,345,287.987	187,453,219.876
FTSE	1.2543	341,265,341.71	335,412,654.219	32,653,298.5431	352,874,563.6784	328,773,654.205
S&P	18,736,253.979	1,632,871.895	19,835,423.983	1,874,532.987	19,934,217.76	16,983,421.731

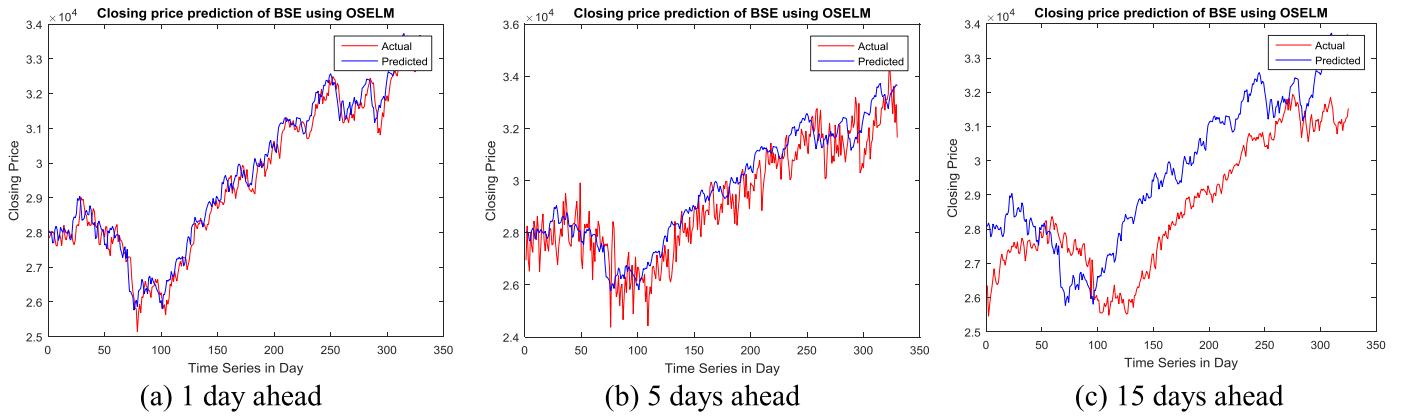


Fig. 9. Closing price prediction of BSE Sensex using FA based on OSELM for 1 day, 5 days and 15 days ahead.

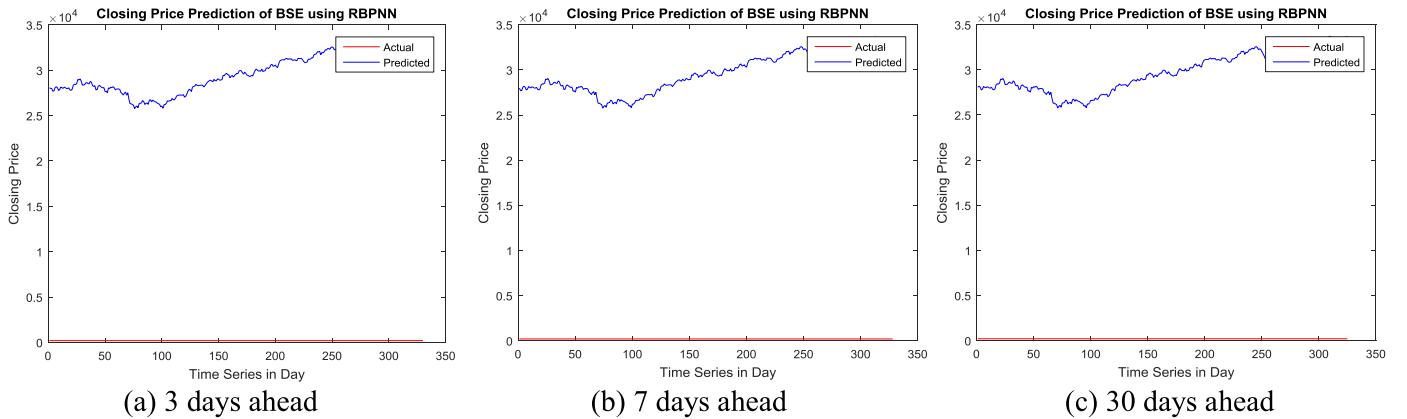


Fig. 10. Closing price prediction of BSE Sensex using FA based on RBPNN for 3 days, 7 days and 30 days ahead.

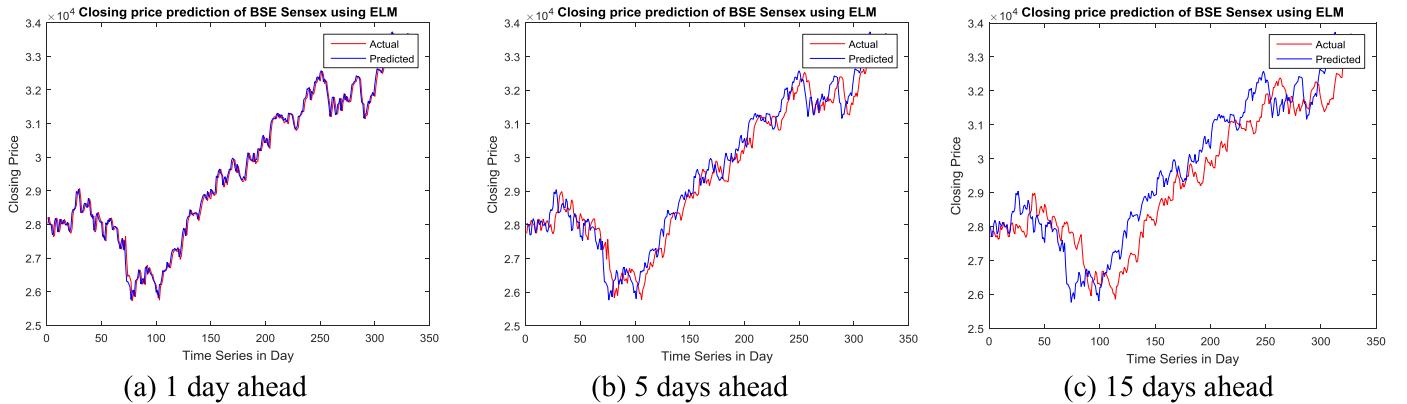


Fig. 11. Closing price prediction of BSE Sensex using GA based on ELM for 1 day, 5 days and 15 days ahead.

Table 9

Day(s) ahead prediction of BSE, NSE, FTSE and S&P Sensex data using Firefly based ELM.

Datasets	1 day	3 days	5 days	7 days	15 days	30 days
BSE	0.0090134	0.025003	0.042767	0.062172	0.11656	0.26138
NSE	0.010138	0.028573	0.048367	0.063843	0.12682	0.13163
FTSE	0.013809	0.038334	0.056995	0.079367	0.18344	0.31857
S&P	0.0064297	0.014781	0.024182	0.033257	0.05935	0.099269

Table 10

Day(s) ahead prediction of BSE, NSE, FTSE and S&P Sensex data using Firefly based OSELM.

Datasets	1 day	3 days	5 days	7 days	15 days	30 days
BSE	0.0089993	0.025008	0.042765	0.062191	0.11655	0.26143
NSE	0.009926	0.028537	0.048713	0.063832	0.12685	0.13154
FTSE	0.013806	0.03833	0.05702	0.079261	0.18382	0.31798
S&P	0.0064446	0.014862	0.024254	0.033571	0.059268	0.098802

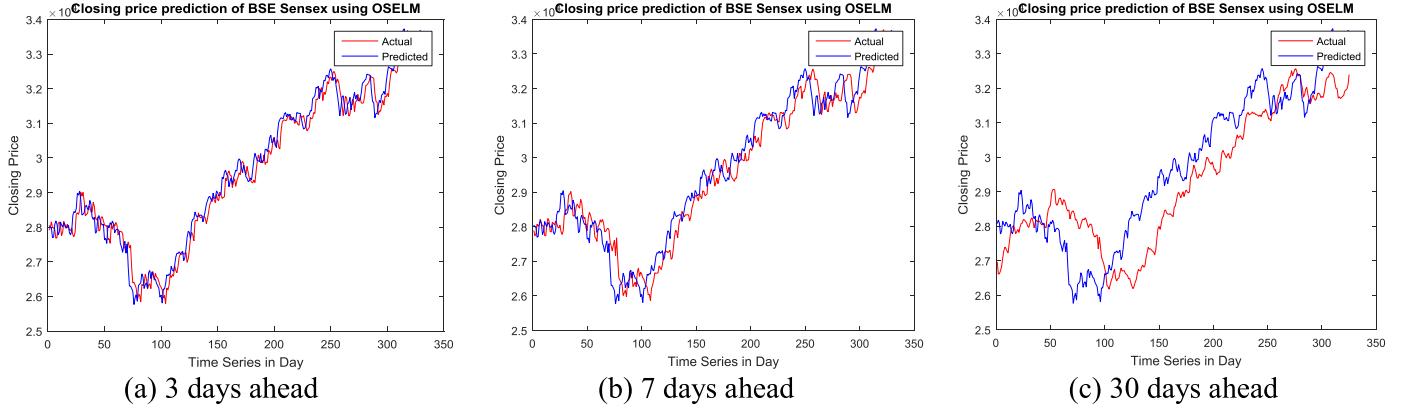


Fig. 12. Closing price prediction of BSE Sensex using GA based on OSELM for 3 days, 7 days and 30 days ahead.

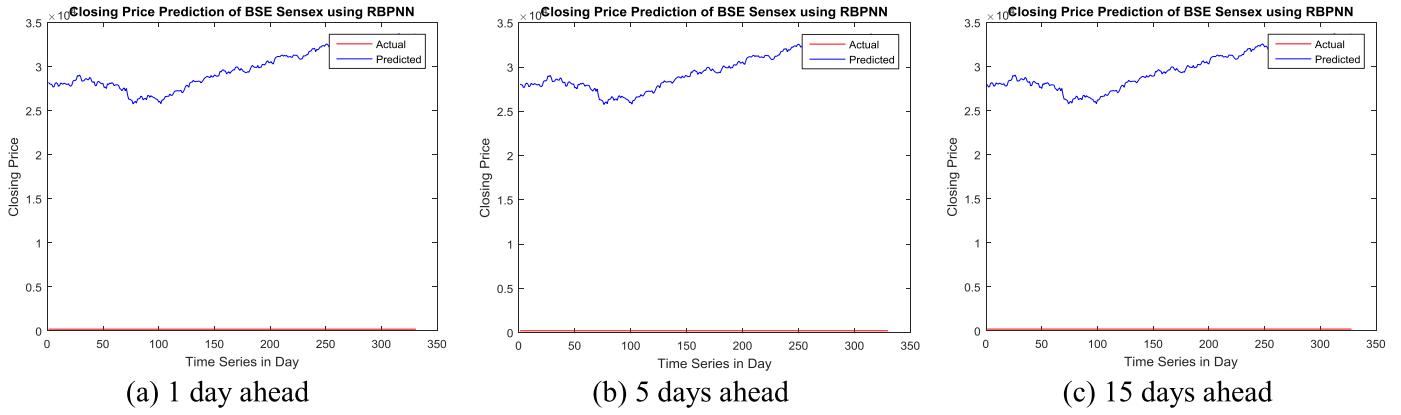


Fig. 13. Closing price prediction of BSE Sensex using GA based on RBPNN for 1 day, 5 days and 15 days ahead.

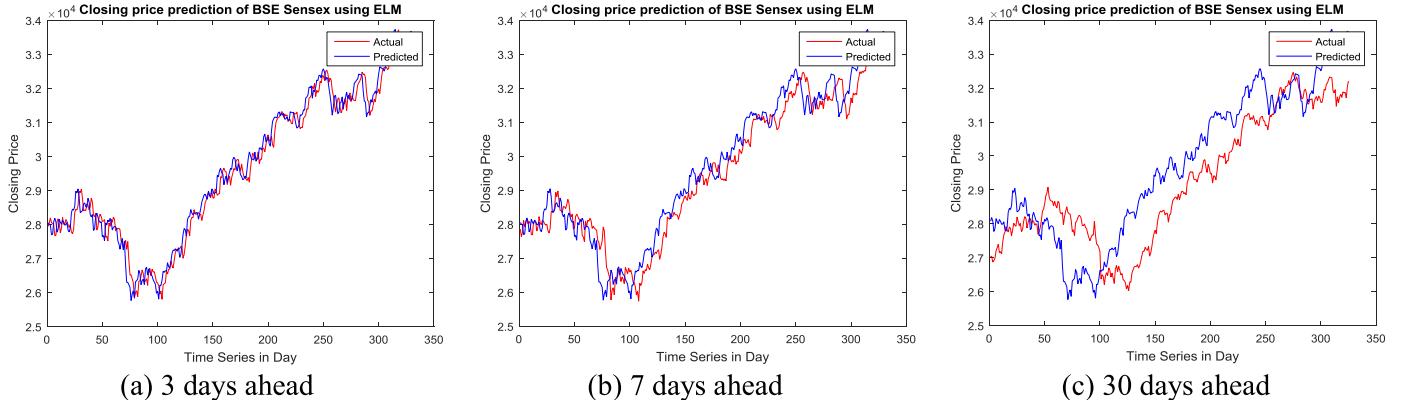


Fig. 14. Closing price prediction of BSE Sensex using Firefly based on ELM for 3 days, 7 days and 30 days ahead.

Table 11
Day(s) ahead prediction of BSE, NSE, FTSE and S&P Sensex data using Firefly based RBPNN.

Datasets	1 day	3 days	5 days	7 days	15 days	30 days
BSE	491,394,728.6185	491,831,967.7814	492,029,869.8954	492,215,197.6812	493,363,721.1107	495,578,969.9148
NSE	15,197,732.0066	15,217,297.064	15,225,619.0984	15,245,544.1953	15,290,323.4466	15,387,719.6822
FTSE	32,195,433.4962	32,197,150.6072	32,195,423.6611	32,195,450.3301	32,199,817.0054	32,205,588.7204
S&P	1,754,970.3258	1,756,106.5801	1,756,498.0106	1,756,946.9937	1,760,259.6852	1,783,452.6487

Table 12
Day(s) ahead prediction of BSE, NSE, FTSE and S&P Sensex data using GA based ELM.

Datasets	1 day	3 days	5 days	7 days	15 days	30 days
BSE	0.0084435	0.025796	0.042979	0.055997	0.12618	0.23868
NSE	0.010165	0.028648	0.048383	0.064304	0.12694	0.12947
FTSE	0.012952	0.042152	0.058261	0.074889	0.18138	0.31339
S&P	0.0061816	0.015256	0.024486	0.030817	0.060276	0.094053

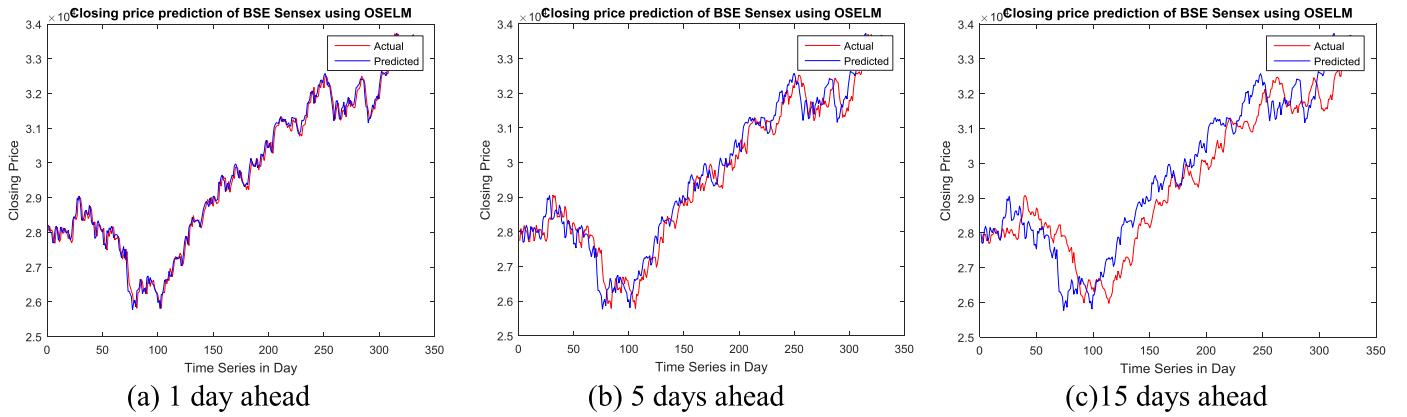


Fig. 15. Closing price prediction of BSE Sensex using Firefly based on OSELML for 1 day, 5 days and 15 days ahead.

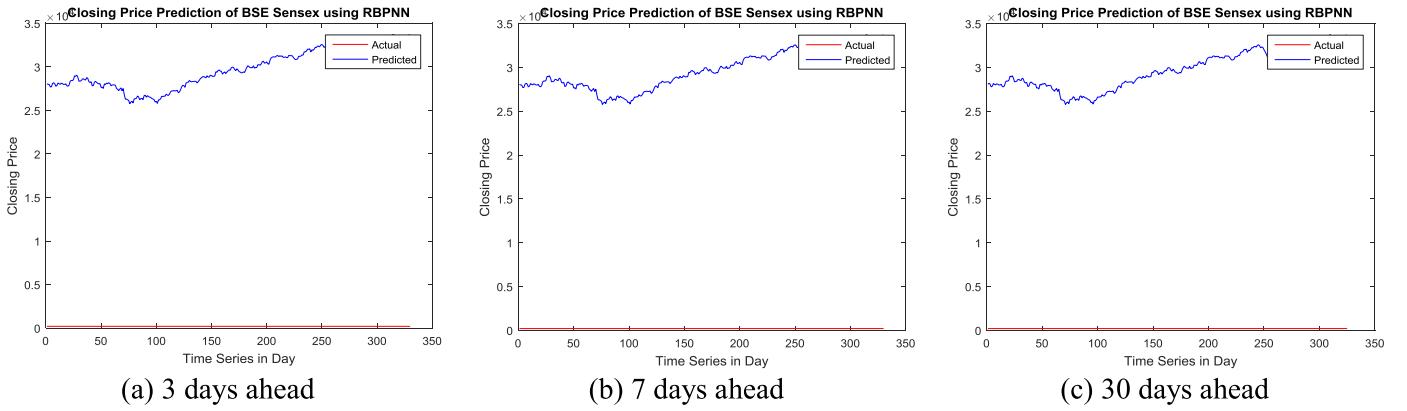


Fig. 16. Closing price prediction of BSE Sensex using Firefly based on RBPNN for 3 days, 7 days and 30 days ahead.

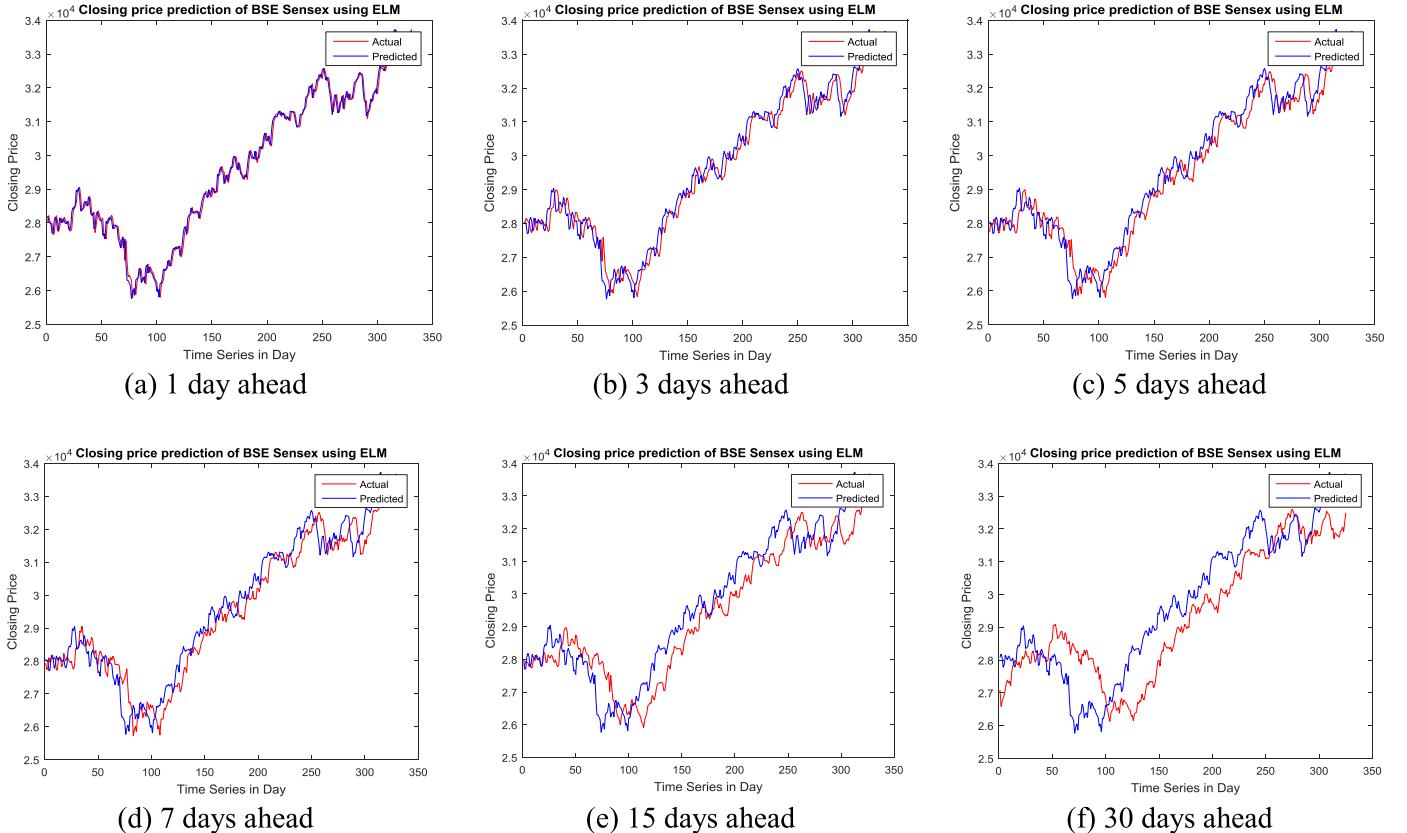


Fig. 17. Closing price prediction of BSE Sensex using Firefly algorithm with evolutionary framework based on ELM for 1 day, 3 days, 5 days, 7 days, 15 days and 30 days ahead.

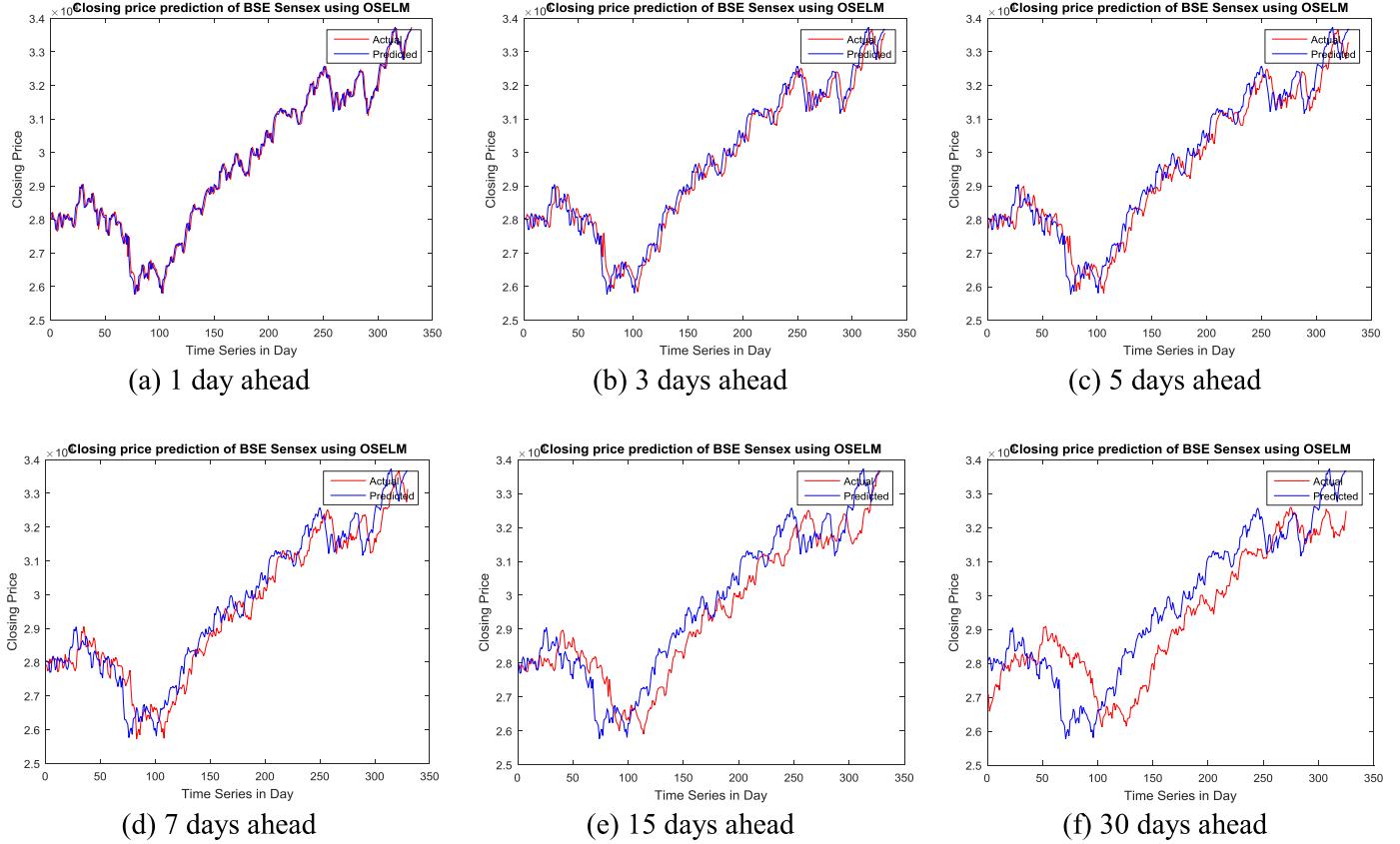


Fig. 18. Closing price prediction of BSE Sensex using Firefly algorithm with evolutionary framework based on OSELM for 1 day, 3 days, 5 days, 7 days, 15 days and 30 days ahead.

Table 13
Day(s) ahead prediction of BSE, NSE, FTSE and S&P Sensex data using GA based OSELM.

Datasets	1 day	3 days	5 days	7 days	15 days	30 days
BSE	0.0084808	0.025797	0.043261	0.056035	0.12624	0.24089
NSE	0.0099718	0.028423	0.048381	0.064277	0.12671	0.12945
FTSE	0.012953	0.042174	0.057773	0.075114	0.17927	0.31259
S&P	0.0061442	0.015252	0.024712	0.031359	0.063171	0.094842

Table 14
Day(s) ahead prediction of BSE, NSE, FTSE and S&P Sensex data using GA based RBPNN.

Datasets	1 day	3 days	5 days	7 days	15 days	30 days
BSE	491,394,729.3744	491,831,967.7814	492,029,870.166	492,215,207.7977	493,363,720.6026	495,578,969.9148
NSE	15,197,731.0221	15,217,296.5716	15,225,616.8814	15,245,544.0896	15,290,324.0815	15,387,719.8804
FTSE	32,195,435.1247	32,197,150.1185	32,195,423.1719	32,195,449.5139	32,199,815.6956	32,205,587.403
S&P	1,754,969.048	1,756,106.58	1,756,497.6907	1,756,947.2224	1,760,256.2438	1,766,096.3268

FTSE and S&P dataset, GA is presenting lesser error than FO in ELM prediction model, whereas for NSE dataset FO is performing better. Similarly, by establishing the comparison of FO and GA applying on OSELM, in case of NSE dataset, FO is found to be performing better than GA but for the rest of the dataset GA is performing better. The Table 14 result shows it worse performance compared to ELM and OSELM regardless of all the feature reduction technique.

The performance of Firefly algorithm with evolutionary framework, which is the combination of FO and GA is better than the individual performance of FO and GA for all the prediction models for the same datasets, which is briefly illustrated in Tables 15–17. The aspect of combining FO with GA is to explore the search space more particularly than the individual explore of search space by

FO. FO can search more efficiently but, whenever the concept of GA is added, which is described briefly in the algorithm part, it is able to explore more precisely. The result shows that Firefly algorithm with evolutionary framework, proficiently predict 1 day ahead data than the rest of the days ahead prediction, which is seen in both ELM and OSELM. Comparing all the prediction outcome of ELM and OSELM with respect to both statistical and optimized based feature reduction techniques, it is observed that not only for 1 day but also for 3 days, 5days, 7 days, 15 days and 30 days the performance of Firefly algorithm with evolutionary framework is better than PCA, FA, FO and GA. Though the result of RBPNN is worse than ELM and OSELM but one thing can be noticed that the result of Firefly algorithm with evolutionary framework based on RBPNN is found

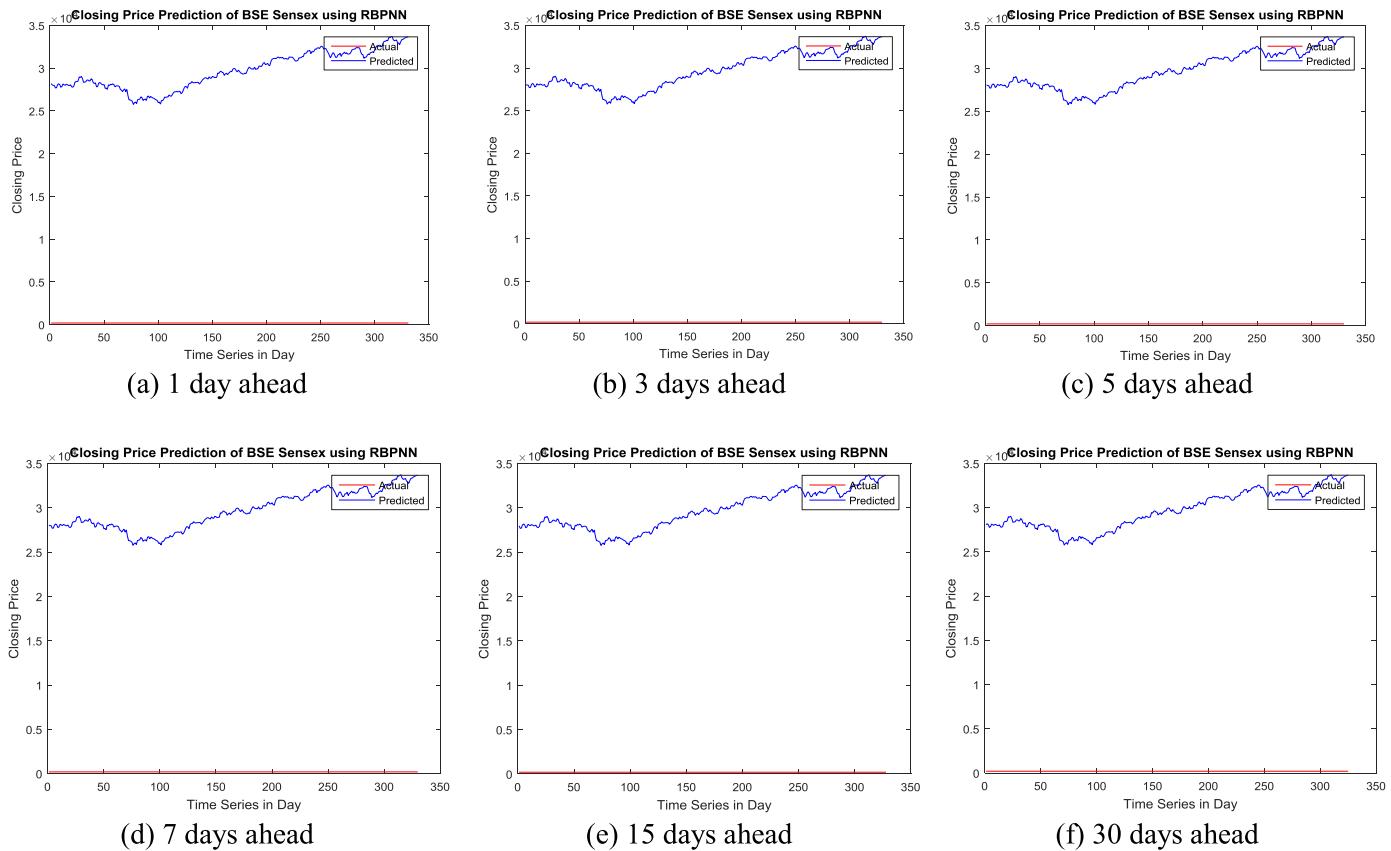


Fig. 19. Closing price prediction of BSE Sensex using Firefly algorithm with evolutionary framework based on RBPNN for 1 day, 3 days, 5 days, 7 days, 15 days and 30 days ahead.

Table 15
Day(s) ahead prediction of BSE, NSE, FTSE and S&P Sensex data using Firefly algorithm with evolutionary framework based ELM.

Datasets	1 day	3 days	5 days	7 days	15 days	30 days
BSE	0.0078098	0.023482	0.041917	0.053478	0.11514	0.23741
NSE	0.0091273	0.024076	0.040227	0.062937	0.12798	0.11878
FTSE	0.011265	0.030216	0.054644	0.076389	0.17015	0.31865
S&P	0.0057362	0.01249	0.024542	0.032532	0.063061	0.088898

Table 16
Day(s) ahead prediction of BSE, NSE, FTSE and S&P Sensex data using Firefly algorithm with evolutionary framework based OSELM.

Datasets	1 day	3 days	5 days	7 days	15 days	30 days
BSE	0.0073082	0.022482	0.042917	0.053454	0.11499	0.23738
NSE	0.0091131	0.026067	0.040271	0.062949	0.12831	0.13608
FTSE	0.0111103	0.033261	0.056459	0.071424	0.17055	0.31145
S&P	0.0054486	0.013434	0.024547	0.031588	0.05145	0.090106

to be performing significantly better than the rest of the feature reduction methods based on RBPNN.

6.5. Performance evaluation criterion

The performance of the predicted models is evaluated by calculating the deviation of predicted stock price and the actual stock price through the calculation of Root Mean Square Error (RMSE), Mean Absolute Percentage Error (MAPE), Mean Absolute Error (MAE), Theil's U and Average Relative Variance (ARV). The above measures are used to assess the performance (Das et al., 2019; Das, Mishra, & Rout, 2017; Ferreira, Vasconcelos, & Adeodato, 2008; Majhi, Rout, & Baghel, 2014) of the prediction model during

the testing phase, whose mathematical formulation is described in Eqs. (17)–(21).

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (A_i - P_i)^2} \quad (17)$$

$$\text{MAPE} = \frac{\sum_{i=1}^N \left| \frac{A_i - P_i}{A_i} \right|}{N} \times 100 \quad (18)$$

$$\text{MAE} = \frac{1}{N} \sum_{i=1}^N |A_i - \bar{P}_i| \quad (19)$$

Table 17

Day(s) ahead prediction of BSE, NSE, FTSE and S&P Sensex data using Firefly algorithm with evolutionary framework based RBPNN.

Datasets	1 day	3 days	5 days	7 days	15 days	30 days
BSE	481,394,727.7547	481,831,967.2771	482,029,870.5267	482,215,194.791	483,363,721.1107	485,578,972.4789
NSE	15,117,730.0377	15,211,297.064	15,205,618.113	15,215,544.5331	15,250,324.0815	15,317,720.6733
FTSE	32,105,431.8679	32,117,150.6072	32,190,422.6127	32,125,449.5139	32,144,816.1868	3,221,588.7205
S&P	1,714,970.3256	1,706,106.5801	1,656,499.2904	1,726,947.2224	1,710,258.0791	1,722,093.904

Table 18

Performance evaluation measures for all the models during testing.

Methods	RMSE	MAPE	MAE	Theil's U	ARV
PCA-ELM	352.1926	3.1366	225.485	0.0026472	0.6748
PCA-OSELM	314.6581	0.66891	764.6662	0.00095769	0.019815
PCA-RBPNN	563,875.67	42.7534	652,314.675	0.017625	131.83564
FA-ELM	421.6123	3.8714	324.8734	0.0054128	0.81654
FA-OSELM	449.6183	1.1273	778.9123	0.00128453	0.071654
FA-RBPNN	27,654.872	341.8735	37,612.983	34.872435	1.72634
Firefly-ELM	195.975	0.50162	151.4629	0.0009365	0.0085354
Firefly-OSELM	196.0351	0.50103	151.4611	0.00098385	0.0085334
Firefly-RBPNN	32,765.0453	23.7612	56,432.986	0.0238716	1.72843
GA-ELM	197.9491	0.50543	145.4301	0.0014054	0.0090632
GA-OSELM	197.9382	0.51682	145.398	0.0013734	0.009054
GA-RBPNN	27,658.6534	12.7634	65,128.873	34.8721	12.98543
Firefly algorithm with evolutionary framework -ELM	159.397	0.32217	132.011	0.0002625	0.0081708
Firefly algorithm with evolutionary framework -OSELM	143.1104	0.30896	121.8011	0.0002543	0.0080989
Firefly algorithm with evolutionary framework -RBPNN	23,845.9823	9.672345	28,761.476	1.872352	1.872433

$$\text{Theil's U} = \frac{\sqrt{\frac{1}{N} \sum_{i=1}^N (A_i - P_i)^2}}{\sqrt{\frac{1}{N} \sum_{i=1}^N A_i^2} + \sqrt{\frac{1}{N} \sum_{i=1}^N P_i^2}} \quad (20)$$

$$\text{ARV} = \frac{\sum_{i=1}^N (P_i - A_i)^2}{\sum_{i=1}^N (P_i - \bar{A})^2} \quad (21)$$

Here, A_i and P_i are the i th day actual and predicted stock price and N is the number of total experimented samples. Lesser the value of RMSE, MAPE, MAE, Theil's U and ARV leads to a better prediction model.

Table 18 shows the summary of the verification result obtained from different performance measures during the testing phase. The performance of different feature reduction techniques with respect to the prediction models is analyzed briefly. In case of PCA feature reduction technique, PCA-OSELM is showing better results in RMSE, MAPE, Theil's U and ARV, except MAE where PCA-ELM is having minimum value. Similarly, in FA, the performance of FA-OSELM is better in terms of minimum value of MAPE, Theil's U and ARV. The rest two measures such as RMSE and MAE is minimum for FA-ELM. In addition to this the performance of firefly and GA with respect to all the prediction model is examined, where it is found that Firefly-OSELM is showing minimum value in terms of MAPE, Theil's U, MAE and ARV error, except RMSE. Similarly GA-OSELM is giving a minimum of RMSE, Theil's U, MAE and ARV error except MAPE, GA-ELM is giving minimum value. Further the efficiency of Firefly algorithm with evolutionary framework is viewed through the result of performance measures presented in Table 18, where it is clearly depicted that the proposed Firefly algorithm with evolutionary framework-OSELM model is outperformed over all the experimented model with the minimum value of RMSE, MAPE, MAE, Theil's U and ARV. All the above performance measures are calculated for 1 day ahead to be predicted for BSE Sensex dataset.

The convergence graph of Firefly, GA and Firefly algorithm with evolutionary framework using MSE for 1 day, 3 days, 5 days, 7 days, 15 days and 30 days ahead for BSE Sensex data is depicted in Fig. 20. The figure itself clearly outlined that with the increase of

days ahead, the convergence speed decreases, it means 1 day and 3 days ahead is converging faster than the rest of days considered here in this study. Fig. 21 addresses the convergence speed of BSE, NSE, S&P 500 and FTSE 100 using MSE for Firefly algorithm with evolutionary framework. The figure shows that BSE Sensex dataset is converging at about 6, NSE is at 10, S&P 500 is at 25 and FTSE is at 51 number of iterations. The convergence speed of BSE, NSE, S&P 500 and FTSE 100 using MSE for GA, Firefly and Firefly algorithm with evolutionary framework is produced in Fig. 22, where the convergence speed of different optimized feature reduction techniques can be marked. From the convergence graph it can be clearly figured that the Firefly algorithm with evolutionary framework, which is the combination of Firefly and GA, is converging faster than the individual converging rate of Firefly and GA optimization algorithm.

7. Overall work analysis

Analyzing the stock market is one of the important task, where many methods are proposed by the researchers to predict stock price with the expectations of getting predicted results closer to the actual result. The empirical results in predicting four different stock values using the feature reduction methods for prediction models are shown in this experimental work.

- (1) BSE Sensex, NSE Sensex, S&P 500 index and FTSE 100 index are four stock market datasets are selected to evaluate the prediction model. BSE Sensex is one of the main stock market indices, which includes 30 blue chip company. Similarly, NSE popularly known as nifty fifty, which covers 50 specialized company and these companies is focused on index as a core product. Apart from these two Indian stock exchange, this study has considered two non Indian stock exchange such as S&P 500 and FTSE 100, which is belongs to American and UK exchange respectively. S&P 500 based on 500 market capitalizations whereas FTSE 100 based on share index of 100 companies with highest market capitalization. For a better evaluation of the experimented models these four different stock exchange value is considered for experimental purpose.

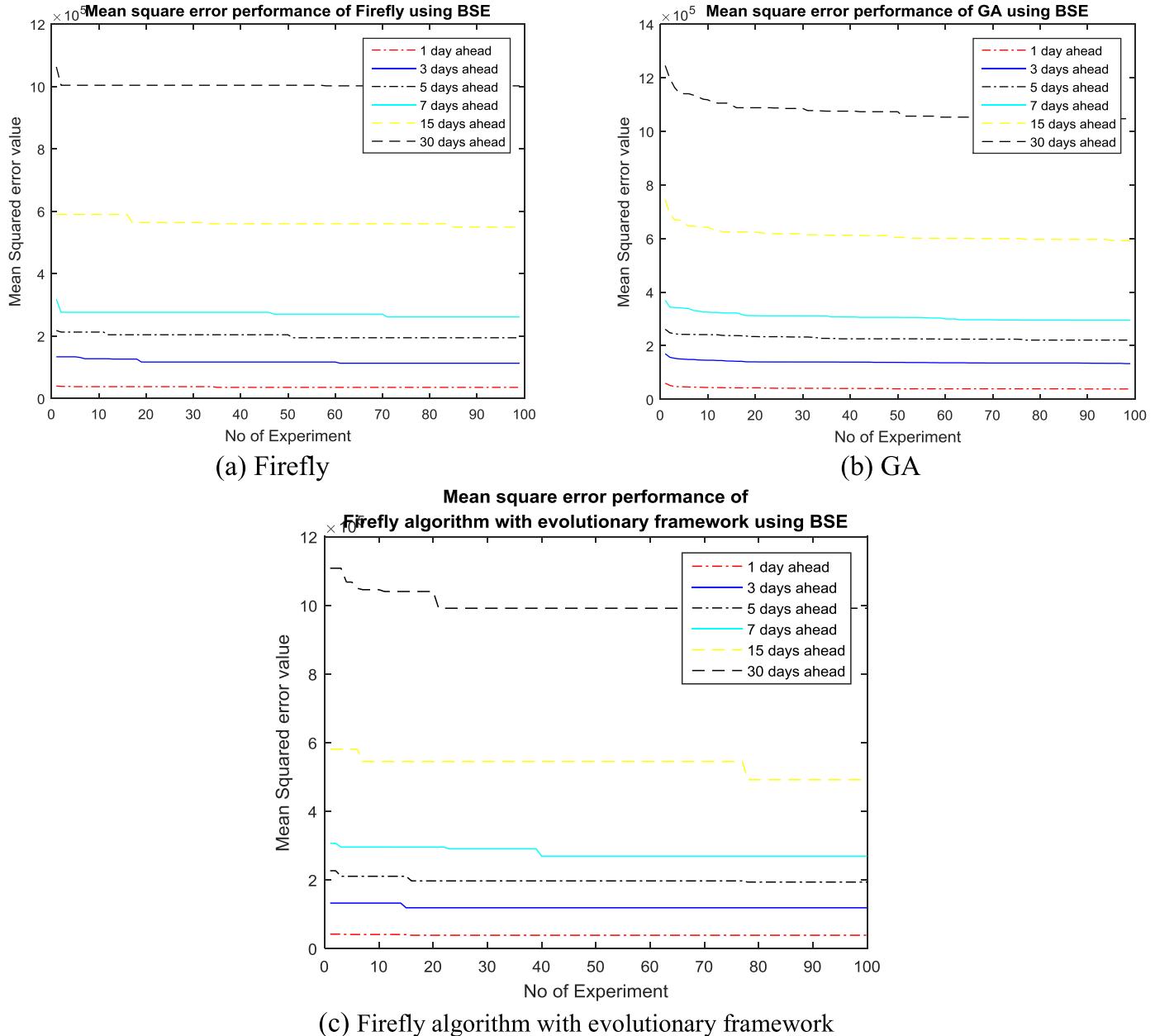


Fig. 20. Convergence speed of (a) Firefly, (b) GA, (c) Firefly algorithm with evolutionary framework using MSE for 1 day, 3 days, 5 days, 7 days, 15 days and 30 days ahead for BSE Sensex data.

- (2) The dataset is regenerated by using technical indicators and statistical measures considering 7 as window size as in our previous work of exchange rate prediction, the window size has explored, where 7 window size is found to be giving better result (Das et al., 2019). The prediction result of six different time horizon such as 1 day, 3 days, 5 days, 7 days, 15 days and 30 days has observed. The enhanced datasets for different time horizon are the input to the prediction models.
- (3) This study compares the prediction result of ELM, OSELM and RBPNN. All these three models are neural network based model and in many of financial market research analysis, they are found to be giving better prediction outcome compared to the statistical based model.
- (4) For statistical based feature reduction, both PCA and FA employing on ELM, OSELM and RBPNN is giving nearly equal to result for all the dataset with the different experimented

time horizon. In both the cases ELM with the reduced features obtained from PCA and FA is performing better for one day ahead prediction than the rest of the time horizon days ahead to be predicted. Similarly the result of optimized based feature reduction is analyzed. GA, firefly and Firefly algorithm with the evolutionary framework optimized feature reduction methods are well thought out for this study. GA is a stochastic algorithm, where there is no possibility of trap into local optima rather it jumps out of locality. Similarly FO is a heuristic population based algorithm, which explore the search space very precisely. Combining the concept of Firefly algorithm with GA, Firefly algorithm with evolutionary framework is designed, where GA updates the current best solutions generated by Firefly Optimization to move towards the best solutions. The motivation behind this integration of firefly with GA is to find good solutions and simultaneously avoiding for trapping into local minima.

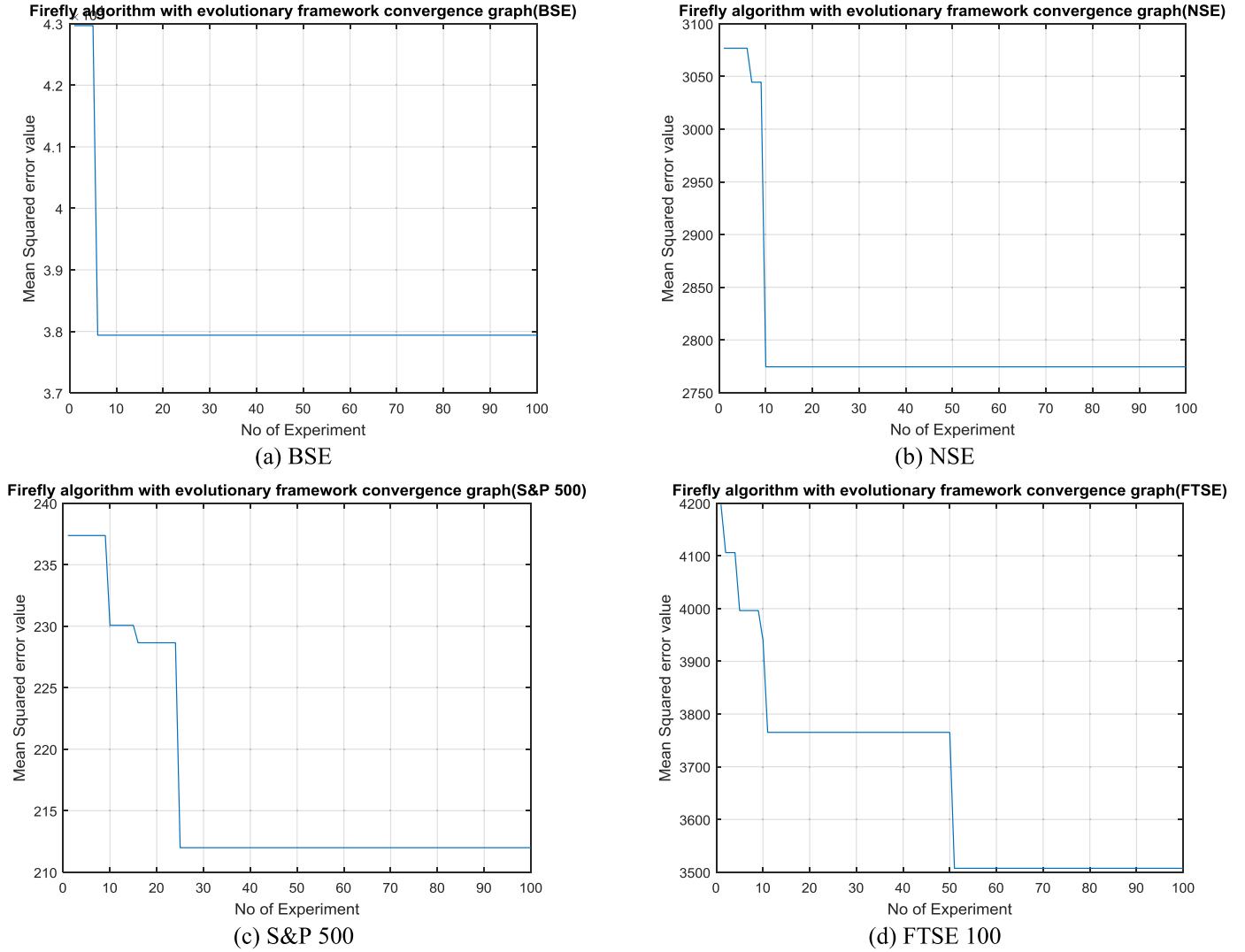


Fig. 21. Convergence speed of (a) BSE, (b) NSE, (c) S&P 500 and (d) FTSE 100 using MSE for Firefly algorithm with evolutionary framework.

For all the optimized feature reduction techniques the common controlling parameters such as number of populations and number of iterations is set to 100 to make the comparison standardize. Apart from this common controlling parameters each algorithm is having some specific parameters. Hence, for a particular algorithm proper parameter value set is based on the architecture of the algorithm, as tuning of this algorithm specific parameter has a vital role in the performance of the model. For GA the two basic parameters, crossover rate and mutation rate, where the crossover rate value at 0.8 and the mutation rate at 0.1 is giving more optimal result according to the time of convergence. Similarly the attraction coefficient and absorption coefficient are two specific parameters of Firefly algorithm, where the attraction coefficient with 1 and absorption coefficient with 0.8 configurations achieved the best result in the course of executing test runs. The simulation result of GA and firefly with respect to the number days in terms of MSE is nearly equal for all the experimented stock market data. In both these optimized feature reduction based method ELM and OSELM are giving better results than RBPNN. In comparison to firefly and GA, Firefly algorithm with the evolutionary framework showing better performance irrespective of the prediction models and datasets. The features reduced by Firefly algorithm with the evolutionary framework for OSELM outperforms over all

the combinations of both statistical and optimized feature reduction methods for all the experimented prediction models, whose relative improvements result for BSE Sensex is shown in Tables 19–23. The relative improved results of the proposed Firefly algorithm with evolutionary framework for OSELM in predicting BSE Sensex follow-up days ahead by,

- More than 59%, 29%, 28%, 21%, 32%, 35% (1, 3, 5, 7, 15, 30 days ahead respectively) compared to PCA-ELM,
- More than 90%, 35%, 91%, 66%, 94%, 50% (1, 3, 5, 7, 15, 30 days ahead respectively) compared to PCA-OSELM,
- More than 99% for 5, 7, 15, 30 days ahead respectively, and almost 100% improved for 1day and 3 days ahead compared to PCA-RBPNN, FA-RBPNN, Firefly-RBPNN, GA-RBPNN and Firefly algorithm with evolutionary framework-RBPNN
- Furthermore the relative improvements of Firefly algorithm with evolutionary framework-OSELM is compared with FA based prediction models, where it is found that, improvements is more than 79%, 75%, 40%, 68%, 36%, 36% (1, 3, 5, 7, 15, 30 days ahead respectively) compared to GA-ELM,
- More than 77%, 95%, 75%, 68%, 69%, 69% (1, 3, 5, 7, 15, 30 days ahead respectively) compared to FA-OSELM,

Table 19

Comparison of overall prediction accuracy of Firefly algorithm with evolutionary framework-OSELM with PCA-ELM, PCA-OSELM and PCA-RBPNN network for stock price prediction base on the MSE for BSE Sensex data.

Day(s) ahead	Comparison level (%)			Comparison level (%)			Comparison level (%)		
	Firefly algorithm with evolutionary framework-OSELM	PCA-ELM	Improvement in (%)	Firefly algorithm with evolutionary framework-OSELM	PCA-OSELM	Improvement in (%)	Firefly algorithm with evolutionary framework-OSELM	PCA-RBPNN	Improvement in (%)
1 day	0.0073082	0.018242	59.93	0.0073082	0.073453	90.05	0.0073082	491,394,731.4709	1
3 days	0.022482	0.031889	29.49	0.022482	0.0348	35.39	0.022482	491,831,971.0308	1
5 days	0.042917	0.060005	28.47	0.042917	0.53266	91.94	0.042917	492,029,873.7347	99.99
7 days	0.053454	0.068082	21.48	0.053454	0.16099	66.79	0.053454	492,215,198.5763	99.99
15 days	0.11499	0.16917	32.02	0.11499	2.2145	94.8	0.11499	493,363,724.1234	99.99
30 days	0.23738	0.36839	35.56	0.23738	0.47495	50.02	0.23738	495,578,973.7223	99.99

Table 20

Comparison of overall prediction accuracy of Firefly algorithm with evolutionary framework-OSELM with FA-ELM, FA-OSELM and FA-RBPNN network for stock price prediction base on the MSE for BSE Sensex data.

Day(s) ahead	Comparison level (%)			Comparison level (%)			Comparison level (%)		
	Firefly algorithm with evolutionary framework-OSELM	FA-ELM	Improvement in (%)	Firefly algorithm with evolutionary framework-OSELM	FA-OSELM	Improvement in (%)	Firefly algorithm with evolutionary framework-OSELM	FA-RBPNN	Improvement in (%)
1 day	0.0073082	0.034941	79.08	0.0073082	0.033134	77.94	0.0073082	491,831,970.9217	1
3 days	0.022482	0.093144	75.86	0.022482	0.41221	95.45	0.022482	492,029,874.0208	1
5 days	0.042917	0.072128	40.49	0.042917	0.17489	75.46	0.042917	492,215,198.9369	99.99
7 days	0.053454	0.17093	68.72	0.053454	0.17232	68.97	0.053454	493,363,724.0737	99.99
15 days	0.11499	0.36457	68.45	0.11499	0.37286	69.16	0.11499	495,578,973.5507	99.99
30 days	0.23738	0.36776	35.45	0.23738	0.78972	69.94	0.23738	495,578,973.2484	99.99

Table 21

Comparison of overall prediction accuracy of Firefly algorithm with evolutionary framework-OSELM with GA-ELM, GA-OSELM and GA-RBPNN network for stock price prediction base on the MSE for BSE Sensex data.

Day(s) ahead	Comparison level (%)			Comparison level (%)			Comparison level (%)		
	Firefly algorithm with evolutionary framework-OSELM	GA-ELM	Improvement in (%)	Firefly algorithm with evolutionary framework-OSELM	GA-OSELM	Improvement in (%)	Firefly algorithm with evolutionary framework-OSELM	GA-RBPNN	Improvement in (%)
1 day	0.0073082	0.0084435	13.44	0.0073082	0.0084808	13.82	0.0073082	491,394,729.3744	1
3 days	0.022482	0.025796	12.84	0.022482	0.025797	12.85	0.022482	491,831,967.7814	1
5 days	0.042917	0.042979	0.14	0.042917	0.043261	0.79	0.042917	492,029,870.166	99.99
7 days	0.053454	0.055997	4.54	0.053454	0.056035	4.6	0.053454	492,215,207.7977	99.99
15 days	0.11499	0.12618	8.86	0.11499	0.12624	8.91	0.11499	493,363,720.6026	99.99
30 days	0.23738	0.23868	0.54	0.23738	0.24089	1.45	0.23738	495,578,969.9148	99.99

Table 22

Comparison of overall prediction accuracy of Firefly algorithm with evolutionary framework-OSELM with Firefly-ELM, Firefly-OSELM and Firefly-RBPNN network for stock price prediction base on the MSE for BSE Sensex data.

Day(s) ahead	Comparison level (%)			Comparison level (%)			Comparison level (%)		
	Firefly algorithm with evolutionary framework-OSELM	Firefly-ELM	Improvement in (%)	Firefly algorithm with evolutionary framework-OSELM	Firefly-OSELM	Improvement in (%)	Firefly algorithm with evolutionary framework-OSELM	Firefly-RBPNN	Improvement in (%)
1 day	0.0073082	0.0090134	18.91	0.0073082	0.0089993	18.79	0.0073082	491,394,728.6185	1
3 days	0.022482	0.025003	10.08	0.022482	0.025008	10.1	0.022482	491,831,967.7814	1
5 days	0.042917	0.042767	No improvement	0.042917	0.042765	No improvement	0.042917	492,029,869.8954	99.99
7 days	0.053454	0.062172	14.02	0.053454	0.062191	14.04	0.053454	492,215,197.6812	99.99
15 days	0.11499	0.11656	1.34	0.11499	0.11655	1.33	0.11499	493,363,721.1107	99.99
30 days	0.23738	0.26138	9.18	0.23738	0.26143	9.19	0.23738	495,578,969.9148	99.99

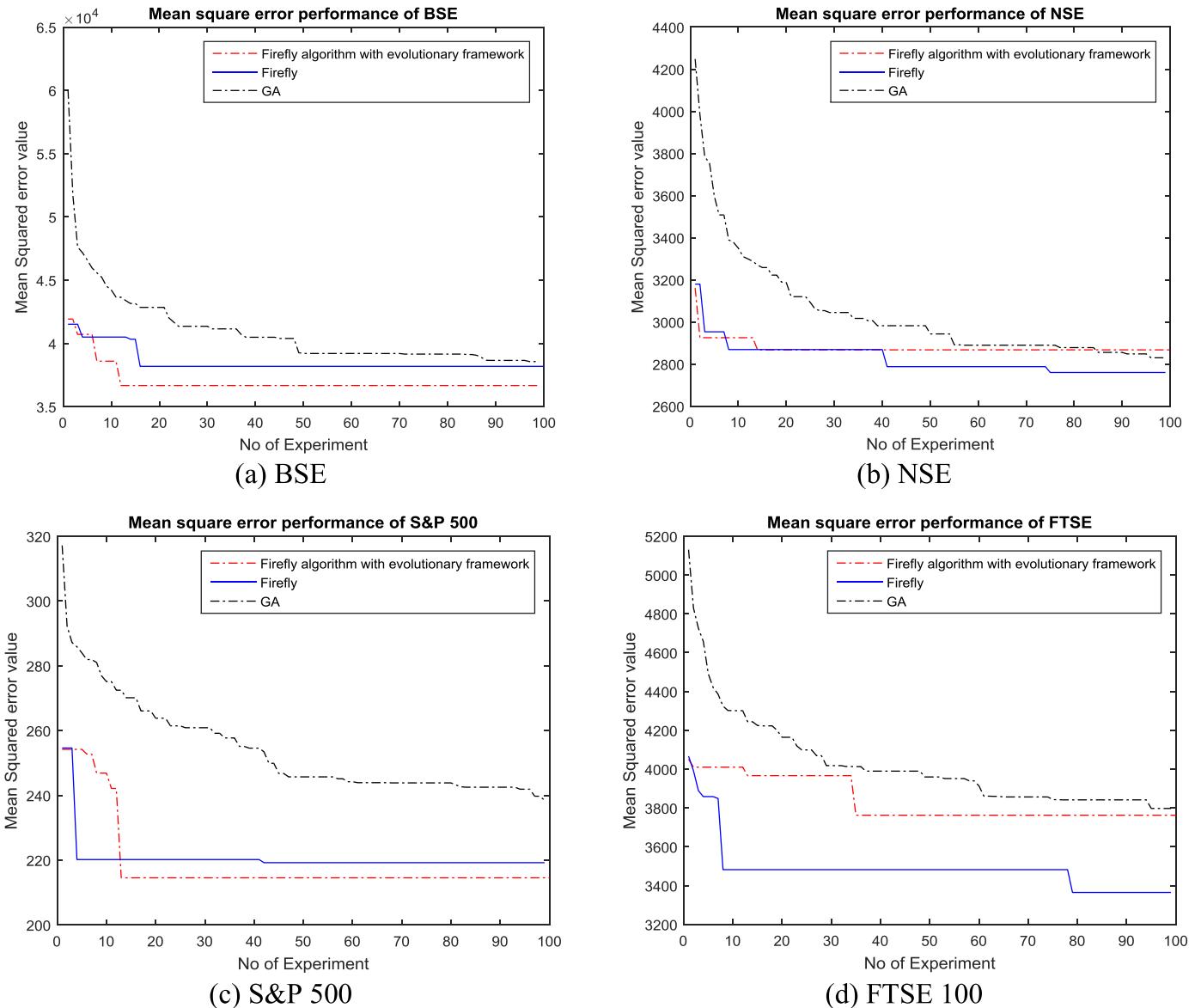


Fig. 22. Convergence speed of (a) BSE, (b) NSE, (c) S&P 500 and (d) FTSE 100 using MSE for GA, Firefly and Firefly algorithm with evolutionary framework.

Table 23

Comparison of overall prediction accuracy of Firefly algorithm with evolutionary framework-OSELM with Firefly algorithm with evolutionary framework-ELM, Firefly algorithm with evolutionary framework-RBPNN network for stock price prediction base on the MSE for BSE Sensex data.

Day(s) ahead	Comparison level (%)			Comparison level (%)		
	Firefly algorithm with evolutionary framework-OSELM	Firefly algorithm with evolutionary framework-ELM	Improvement in (%)	Firefly algorithm with evolutionary framework-OSELM	Firefly algorithm with evolutionary framework-RBPNN	Improvement in (%)
1 day	0.0073082	0.0078098	6.42	0.0073082	481,394,727.7547	1
3 days	0.022482	0.023482	4.25	0.022482	481,831,967.2771	1
5 days	0.042917	0.041917	No improvement	0.042917	482,029,870.5267	99.99
7 days	0.053454	0.053478	0.04	0.053454	482,215,194.791	99.99
15 days	0.11499	0.11514	0.13	0.11499	483,363,721.1107	99.99
30 days	0.23738	0.23741	0.01	0.23738	485,578,972.4789	99.99

- (f) More than 13%, 12%, 0.14%, 4.54%, 8.86%, 0.54% (1, 3, 5, 7, 15, 30 days ahead respectively) compared to GA-ELM,
- (g) More than 13%, 12%, 0.79%, 4.6%, 8.91%, 1.45% (1, 3, 5, 7, 15, 30 days ahead respectively) compared to GA-OSELM,
- (h) More than 18%, 10%, 14%, 1.34%, 9.18% (1, 3, 7, 15, 30 days ahead respectively) compared to Firefly-ELM, but there is no improvements for 5 days ahead to be predicted,

- (i) More than 18%, 10%, 14%, 1.33%, 9.19% (1, 3, 7, 15, 30 days ahead respectively) compared to Firefly-OSELM, here also no improvements occur for 5 days ahead prediction.
- (j) More than 6%, 4%, 0.04%, 0.13%, 0.01% (1, 3, 7, 15, 30 days ahead respectively) compared to Firefly algorithm with evolutionary framework-ELM and there are no improvements for 5 days ahead prediction.

The aforementioned remarks about the performance of Firefly algorithm with evolutionary framework for OSELM should be accompanied by a conclusion that the proposed model is significantly improved and better than other feature reduction through transformation method applied on the prediction models used for the experimentation by this study.

8. Conclusions

This study presents the Firefly algorithm with evolutionary framework based feature reduction through transformation model for OSELM to predict the future stock price. The proposed model's performance is evaluated using four stock market dataset such as BSE Sensex, NSE Sensex, S&P 500 index and FTSE index. The objective on feature reduction focused on two factors: statistical based feature reduction and optimized feature reduction. PCA and FA are considered for statistical based feature reduction, whereas Firefly, GA and Firefly algorithm with evolutionary framework are used to reduce the features using the respective optimization technique. After reducing the features the datasets are employed to three prediction models such as ELM, OSELM and RBPNN, exploring the predicted outcome of six time horizon. Though, all the feature reduction methods have the ability to reduce features with the expectation of getting better result, but the potential of the proposed Firefly algorithm with evolutionary framework is very high and also delineating outstanding performance for OSELM prediction model. In future, this work can be extended to other financial aspects with analyzing their influencing factors.

Declaration of Competing Interest

There is no conflict of interest.

CRediT authorship contribution statement

Smruti Rekha Das: Conceptualization, Data curation, Formal analysis, Methodology. **Debahuti Mishra:** Supervision, Validation. **Minakhi Rout:** Writing - review & editing.

References

- Abu-Mostafa, Y. S., & Atiya, A. F. (1996). Introduction to financial forecasting. *Applied Intelligence*, 6(3), 205–213.
- Arévalo, R., García, J., Guijarro, F., & Peris, A. (2017). A dynamic trading rule based on filtered flag pattern recognition for stock market price forecasting. *Expert Systems with Applications*, 81, 177–192.
- Arora, S., & Singh, S. (2013). The firefly optimization algorithm: Convergence analysis and parameter selection. *International Journal of Computer Applications*, 69(3), 48–52.
- Baek, Y., & Kim, H. Y. (2018). ModAugNet: A new forecasting framework for stock market index value with an overfitting prevention LSTM module and a prediction LSTM module. *Expert Systems with Applications*, 113, 457–480.
- Bebartha, D. K., & Venkatesh, G. (2016). A low complexity FLANN architecture for forecasting stock time series data training with meta-heuristic firefly algorithm. In *Computational intelligence in data mining: vol. 1* (pp. 377–385). New Delhi: Springer.
- Bueno-Crespo, A., García-Laencina, P. J., & Sancho-Gómez, J. L. (2013). Neural architecture design based on an extreme learning machine. *Neural Networks*, 48, 19–24.
- Cao, L. J., Chua, K. S., Chong, W. K., Lee, H. P., & Gu, Q. M. (2003). A comparison of PCA, KPCA and ICA for dimensionality reduction in support vector machine. *Neurocomputing*, 55(1–2), 321–336.
- Cootner, P. H. (1964). *The random character of stock market prices*.
- Das, S. R., Mishra, D., & Rout, M. (2017). A hybridized ELM-Jaya forecasting model for currency exchange prediction. *Journal of King Saud University-Computer and Information Sciences*.
- Das, S. R., Mishra, D., & Rout, M. (2019). An optimized feature reduction based currency forecasting model exploring the online sequential extreme learning machine and krill herd strategies. *Physica A: Statistical Mechanics and its Applications*, 513, 339–370.
- Deo, R. C., & Shahin, M. (2015). Application of the extreme learning machine algorithm for the prediction of monthly Effective Drought Index in eastern Australia. *Atmospheric Research*, 153, 512–525.
- Ding, S., Zhao, H., Zhang, Y., Xu, X., & Nie, R. (2015). Extreme learning machine: Algorithm, theory and applications. *Artificial Intelligence Review*, 44(1), 103–115.
- dos Santos Coelho, L., de Andrade Bernert, D. L., & Mariani, V. C. (2011). A chaotic firefly algorithm applied to reliability-redundancy optimization. In *2011 IEEE congress of evolutionary computation (CEC), June* (pp. 517–521). IEEE.
- Fama, E. F. (1965). The behavior of stock-market prices. *The Journal of Business*, 38(1), 34–105.
- Ferreira, T. A., Vasconcelos, G. C., & Adeodato, P. J. (2008). A new intelligent system methodology for time series forecasting with artificial neural networks. *Neural Processing Letters*, 28(2), 113–129.
- Fister, I., Fister, I., Jr, Yang, X. S., & Brest, J. (2013). A comprehensive review of firefly algorithms. *Swarm and Evolutionary Computation*, 13, 34–46.
- Gandomi, A. H., Yang, X. S., Talatahari, S., & Alavi, A. H. (2013). Firefly algorithm with chaos. *Communications in Nonlinear Science and Numerical Simulation*, 18(1), 89–98.
- Giri, D., Acharya, U. R., Martis, R. J., Sree, S. V., Lim, T. C., Thajudin Ahamed, et al. (2013). Automated diagnosis of coronary artery disease affected patients using LDA, PCA, ICA and discrete wavelet transform. *Knowledge-Based Systems*, 37, 274–282.
- Huang, G. B., & Siew, C. K. (2004). Extreme learning machine: RBF network case. In *ICARCV 2004 8th control, automation, robotics and vision conference, 2004, December*: 2 (pp. 1029–1036). IEEE.
- Huang, G. B., Zhou, H., Ding, X., & Zhang, R. (2012). Extreme learning machine for regression and multiclass classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 42(2), 513–529.
- Huang, G. B., Zhu, Q. Y., & Siew, C. K. (2004). Extreme learning machine: A new learning scheme of feedforward neural networks. *Neural Networks*, 2, 985–990.
- Huang, G. B., Zhu, Q. Y., & Siew, C. K. (2006). Extreme learning machine: Theory and applications. *Neurocomputing*, 70(1–3), 489–501.
- Jarvis, R. M., & Goodacre, R. (2004). Genetic algorithm optimization for pre-processing and variable selection of spectroscopic data. *Bioinformatics*, 21(7), 860–868.
- Jolliffe, I. (2011). *Principal component analysis* (pp. 1094–1096). Berlin Heidelberg: Springer.
- Jujie, W., & Danfeng, Q. (2018). An experimental investigation of two hybrid frameworks for stock index prediction using neural network and support vector regression. *Economic Computation & Economic Cybernetics Studies & Research*, 52(4), 193–210.
- Kazem, A., Sharifi, E., Hussain, F. K., Saberi, M., & Hussain, O. K. (2013). Support vector regression with chaos-based firefly algorithm for stock market price forecasting. *Applied Soft Computing*, 13(2), 947–958.
- Kim, J. O., & Mueller, C. W. (1978). *Factor analysis: Statistical methods and practical issues*. Sage (No. 14).
- Kim, K. J. (2006). Artificial neural networks with evolutionary instance selection for financial forecasting. *Expert Systems with Applications*, 30(3), 519–526.
- Kim, K. J., & Han, I. (2000). Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index. *Expert systems with Applications*, 19(2), 125–132.
- Kuo, R. J., & Li, P. S. (2016). Taiwanese export trade forecasting using firefly algorithm based K-means algorithm and SVR with wavelet transform. *Computers & Industrial Engineering*, 99, 153–161.
- Lam, S. L., & Lee, D. L. (1999). Feature reduction for neural network based text categorization. In *Proceedings. 6th international conference on advanced systems for advanced applications* (pp. 195–202). IEEE.
- Liang, N. Y., Huang, G. B., Saratchandran, P., & Sundararajan, N. (2006). A fast and accurate online sequential learning algorithm for feedforward networks. *IEEE Transactions on Neural Networks*, 17(6), 1411–1423.
- Liu, H., & Motoda, H. (2012). *Feature selection for knowledge discovery and data mining*: vol. 454. Springer Science & Business Media.
- Lukasik, S., & Źak, S. (2009). Firefly algorithm for continuous constrained optimization tasks. In *International conference on computational collective intelligence, October* (pp. 97–106). Berlin, Heidelberg: Springer.
- Majhi, B., Rout, M., & Baghel, V. (2014). On the development and performance evaluation of a multiobjective GA-based RBF adaptive model for the prediction of stock indices. *Journal of King Saud University-Computer and Information Sciences*, 26(3), 319–331.
- Majhi, R., Panda, G., & Sahoo, G. (2009). Development and performance evaluation of FLANN based model for forecasting of stock markets. *Expert Systems with Applications*, 36(3), 6800–6808.
- Malkiel, B. G., & Fama, E. F. (1970). Efficient capital markets: A review of theory and empirical work. *The Journal of Finance*, 25(2), 383–417.
- Moghaddam, A. H., Moghaddam, M. H., & Esfandyari, M. (2016). Stock market index prediction using artificial neural network. *Journal of Economics, Finance and Administrative Science*, 21(41), 89–93.
- Naik, B., Nayak, J., & Behera, H. S. (2015). A novel FLANN with a hybrid PSO and GA based gradient descent learning for classification. In *Proceedings of the 3rd international conference on frontiers of intelligent computing: Theory and applications (FICTA) 2014* (pp. 745–754). Cham: Springer.
- Nayak, R. K., Mishra, D., & Rath, A. K. (2015). A Naïve SVM-KNN based stock market trend reversal analysis for Indian benchmark indices. *Applied Soft Computing*, 35, 670–680.
- Nhu, H. N., Nitsuwat, S., & Sodanil, M. (2013). Prediction of stock price using an adaptive neuro-fuzzy inference system trained by firefly algorithm. In *2013 international computer science and engineering conference (ICSEC), September* (pp. 302–307). IEEE.
- Painton, L., & Campbell, J. (1995). Genetic algorithms in optimization of system reliability. *IEEE Transactions on Reliability*, 44(2), 172–178.

- Rahmani, A., & MirHassani, S. A. (2014). A hybrid firefly-genetic algorithm for the capacitated facility location problem. *Information Sciences*, 283, 70–78.
- Rong, H. J., Huang, G. B., Sundararajan, N., & Saratchandran, P. (2009). Online sequential fuzzy extreme learning machine for function approximation and classification problems. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 39(4), 1067–1072.
- Sajjadi, S., Shamshirband, S., Alizamir, M., Yee, L., Mansor, Z., & Manaf, A. A. (2016). Extreme learning machine for prediction of heat load in district heating systems. *Energy and Buildings*, 122, 222–227.
- Sakim, H. A. M., & Mustaffa, M. T. (2014). *The 8th international conference on robotic, vision, signal processing & power applications*.
- Shen, W., Guo, X., Wu, C., & Wu, D. (2011). Forecasting stock indices using radial basis function neural networks optimized by artificial fish swarm algorithm. *Knowledge-Based Systems*, 24(3), 378–385.
- Tsinaslanidis, P. E. (2018). Subsequence dynamic time warping for charting: Bullish and bearish class predictions for NYSE stocks. *Expert Systems with Applications*, 94, 193–204.
- Villegas, F. J. (2007). Parallel genetic-algorithm optimization of shaped beam coverage areas using planar 2-D phased arrays. *IEEE Transactions on Antennas and Propagation*, 55(6), 1745–1753.
- Wang, Y., Cao, F., & Yuan, Y. (2011). A study on effectiveness of extreme learning machine. *Neurocomputing*, 74(16), 2483–2490.
- Xanthopoulos, P., Pardalos, P. M., & Trafalis, T. B. (2013). Principal component analysis. In *Robust data mining* (pp. 21–26). New York, NY: Springer.
- Xiong, T., Bao, Y., & Hu, Z. (2014). Multiple-output support vector regression with a firefly algorithm for interval-valued stock price index forecasting. *Knowledge-Based Systems*, 55, 87–100.
- Xu, Y., Ke, Z., Xie, C., & Zhou, W. (2018). Dynamic evolution analysis of stock price fluctuation and its control. *Complexity*, 2018, 1–9.
- Yang, X. S. (2009). Firefly algorithms for multimodal optimization. In *International symposium on stochastic algorithms, October* (pp. 169–178). Berlin, Heidelberg: Springer.
- Yang, X. S. (2010a). *Nature-inspired metaheuristic algorithms*. Luniver press.
- Yang, X. S. (2010b). Firefly algorithm, stochastic test functions and design optimisation. arXiv preprint arXiv:1003.1409.
- Yang, X. S. (2010c). Firefly algorithm, Levy flights and global optimization. In *Research and development in intelligent systems XXVI* (pp. 209–218). London: Springer.
- Yetis, Y., Kaplan, H., & Jamshidi, M. (2014). Stock market prediction by using artificial neural network. In *2014 world automation congress (WAC), August* (pp. 718–722). IEEE.
- Zhang, Y., & Wu, L. (2009). Stock market prediction of S&P 500 via combination of improved BCO approach and BP neural network. *Expert Systems with Applications*, 36(5), 8849–8854.