Developing a novel stock index trend predictor model by integrating multiple criteria decision-making with an optimized online sequential extreme learning machine

Sidharth Samal¹ · Rajashree Dash¹

Received: 29 April 2022 / Accepted: 19 July 2022 / Published online: 3 August 2022
© The Author(s), under exclusive licence to Springer Nature Switzerland AG 2022

Abstract
It has always been the goal of many researchers to gain a thorough understanding of the patterns in the stock market and forecast the trends it will follow. The use of an advanced forecasting model can assist with accurately forecasting the future price of stocks, their fluctuations in the markets, as well as make profits in trading. With this motivation, in this study, a novel stock index trend predictor model is designed by integrating Multiple Criteria Decision-Making (MCDM) with an optimized Online Sequential Extreme Learning Machine (OSELM). Forecasting the future stock index prices and analyzing the upward or downward trends of these price forecasts are the two objectives of the proposed model. As the performance of OSELM is heavily dependent on the activation functions used in it, suitable selection of the activation function for OSELM is addressed as a MCDM problem. According to this approach, the trend prediction performance of six popular activation functions is assessed based on five regression-based and five classification-based criteria. In this investigation, three MCDM approaches are used to assess the performance matrix and determine which activation function is the best for OSELM based on six alternative models and ten criteria. To further optimize OSELM’s performance, a hybrid crow search algorithm (hCSA) is incorporated in its training phase. By introducing the chaotic map and mutation operator in position update scheme and catfish behavior in the search process of original CSA, the proposed hCSA is able to achieve the right balance between exploration and exploitation improving the convergence. The proposed trend predictor model is empirically evaluated over historical data of three stock indices such as BSE SENSEX, S&P 500 and DJIA collected during pre-COVID and COVID time frame. In most of the test cases, the hCSA-OSELM model outperforms the state-of-the-art baseline models in terms of all evaluation criteria. When compared to the second-best baseline model, the suggested model is able to achieve the MSE improvements of 4–6%, 25–31%, and accuracy improvements of 0.4–0.8%, 0.9–1.3% over the pre-COVID and COVID time-frames, respectively. The statistical test also reveals the better performance of the proposed model. The robust and reliable MCDM-based model selection, superior prediction and classification outcomes clearly reveal that the proposed model can be used for financial time-series forecasting amid daily volatility as well as highly volatile markets.

Keywords Stock index trend analysis · ELM · OSELM · Crow search · MCDM

1 Introduction

Analysts and researchers have always been interested in analyzing and forecasting the patterns and insights of financial time-series data since it is at the forefront of economic risk analysis and decision-making. However, data derived from stock prices, gold prices, crude oil prices; bank interest rates and other sources frequently exhibit the following characteristics (Cavalcante et al. 2016; Tsay 2005): (1) the unpredictability and intricacy of the data; (2)
the data’s inherent non-linear correlations; and (3) the constant volatility. Hence, analyzing the patterns in the stock market and forecasting the trend the market might follow is exceptionally challenging. Furthermore, the market is excessively influenced by investor’s psychology and expectations, global economic conditions, fluctuations in other stock markets, government and bank policies, etc. (Park and Shin 2013). Nevertheless, due to the high rise in research related to financial time-series forecasting, this field of research has seen numerous new advancements, particularly in stock price prediction, movement and volatility prediction, and trend analysis (Dash and Dash 2015a, b; Tsay 2005). Numerous classical forecasting algorithms, such as Auto-Regressive Integrated Moving Average (ARIMA) (Ariyo et al. 2014), Generalized Auto Regressive Conditional Heteroskedasticity (GARCH) (Dana 2016), have been proposed by researchers. However, because these traditional forecasting methods are based on statistical approaches, they are insufficient to identify the inherent nonlinear compositions in time series data (Zheng and Zhong 2011). Subsequently, advanced machine-learning (ML) strategies such as Artificial Neural Network (ANN) and Functional Link Artificial Neural Network (FLANN) (Dash and Dash 2015b), Support Vector Machine (SVM) and Support Vector Regressor (SVR) (Kara et al. 2011; Vilela et al. 2019), Multilayer Perceptron (MLP) (Ravi et al. 2017), Linear Regression (LR), Radial Basis Function Network (RBFN) (Dash and Dash 2015a), Elman (Wang et al. 2021), etc. were able to address the nonlinearity in time series data.

Meanwhile, due to the gradient descent-based backpropagation training, complicated network architecture, frequent parameter adjustment, slower learning rate, and higher computing cost, these traditional ML algorithms are less suitable for forecasting (Fernández-Navarro et al. 2012). Huang et al. countered these limitations by introducing Extreme Learning Machine (ELM), a single hidden layer feed forward network (SLFN) architecture-based method (Huang et al. 2006, 2011). The input layer weights in a basic ELM network were assigned arbitrarily, whereas the output layer weights were calculated mathematically. Since its inception, ELM has seen a wide range of applications, including electrical load prediction (Zhang et al. 2013), financial market prediction (Das et al. 2020; Dash et al. 2014; Dash and Dash 2015b; Samal and Dash 2021a), sales market analysis (Sun et al. 2008) and many others (Huang et al. 2011). However, a preliminary survey of the literature reveals a number of significant challenges relating to ELM: (a) Single pass non-iterative training may not produce efficient results; (b) The incorporation of random, non-optimal input weights may have an influence on outputs. (c) The use of insufficient activation functions in the hidden layers has a significant effect on network performance. For solving the first issue, Huang et al. proposed Online Sequential ELM (OSELM), an ELM-based sequential batch learning algorithm that can learn training data in chunks of customizable length (Huang et al. 2005). OSELM has been used in a variety of research fields, including power quality event detection (Sahani et al. 2020), time series analysis (Das et al. 2019), and stream flow forecasting (Lima et al. 2017), and has outperformed basic ELM and other ML techniques as well. Several researchers have addressed the second issue by hybridizing the training of ELM using optimization techniques such as particle swarm optimization (PSO) (Pradeepkumar and Ravi 2017), harmony search (HS) (Dash et al. 2014), grey wolf optimization (GWO) (Liu et al. 2021), teaching-learning-based optimization (TLBO) (Das and Padhy 2018), crow search algorithm (CSA) (Dash et al. 2021), differential evolution (DE) (Abdual-Salam et al. 2010), and others. These models not only increase accuracy, but also enhance stability of the model. Third, the activation function employed in the hidden layer neurons has a direct impact on ELM performance. In the literature, researchers (Das et al. 2019; Yu et al. 2020) have only employed widespread activation functions such as Sigmoid, Gaussian, Tanh, and others to predict financial time series using ELM and its variations. According to a recent study (Samal and Dash 2021a), no single activation function can achieve excellent performance for diverse datasets, and linear unit-based activation functions have a higher overall generalization capability for stock market prediction.

According to a comprehensive review of the literature, ELM solves the limitations of conventional ML models in terms of computing time, parameter adjustment, and iterative training, among others. The sequential single-pass implementation of ELM, on the other hand, may produce sub-optimal results. This limitation of ELM is rectified by OSELM which outperforms ELM as well as other prominent online learning algorithms in terms of generalization capability while learning at a significantly faster rate (Guo et al. 2018a). Although OSELM outperforms basic ELM in terms of performance, the selection of an appropriate activation function for OSELM’s hidden layer has not been discussed in the literature, and only popular activation functions such as Sigmoid and Gaussian have been used for prediction of financial time-series data. Certain research (Samal and Dash 2021a), on the other hand, imply that activation functions of the Linear unit family (ELU, ReLU, LReLU, PReLU) can produce results comparable or better than popular activation functions. As a result, the determination of activation function necessitates additional analysis. Furthermore, while researchers have investigated the performance of many predictors or classifiers using a single criterion such as MSE or accuracy, assessing predictor performance using multiple criteria can result in a
robust and reliable output (Kou et al. 2012) and an MCDM-based analysis may provide robust outcome (Dash et al. 2018, 2019; Mehdiyev et al. 2016; Samal and Dash 2021a). Another shortcoming of ELM and OSELM is that their input weights are arbitrary, which might lead to inferior results. Therefore, choosing the most appropriate optimization algorithm among the wide variety for the task at hand remains a gap in the literature. Further, the significant proportion of analysis is involved with either predicting the continuous stock index price or classifying stock index movements. However, no other study, in our assessment, provides a forecasting model that can predict the price while also categorizing the anticipated price in terms of movement. This area also has to be improved as a single model should be capable of predicting and classifying at the same time.

Finally, despite the fact that the stock market is extremely volatile, it grew increasingly unstable during the global epidemic COVID-19 and triggered the circuit breaker mechanism. Is it possible for modern ML algorithms to predict this unforeseeable scenario? If so, how accurate is their forecast during the epidemic compared to typical trading days? As a result, a study is needed to develop a more robust and efficient prediction model, as well as to test how adaptable and adjustable the model is during normal trading days and during an unforeseen event.

The primary objective of this work is to design a time series prediction framework for predicting the upcoming price and classify its respective trends with utmost robustness and reliability, as well as the ability to adapt and extrapolate highly volatile market data during unforeseen scenarios such as COVID-19. Hence, in this study, a novel stock index trend analysis framework is proposed in which the upcoming stock index price is forecasted using an OSELM and the predicted price is further classified to identify whether the movement is upward or downward. Here the same model is used for prediction as well as classification. The proposed framework has two main contributions. First of all, the selection of the activation function for OSELM in its design phase is addressed as a Multiple Criteria Decision-Making (MCDM) problem. Further to enhance its performance, a hybrid crow search algorithm (hCSA) is incorporated in its training phase. According to this approach, initially the trend prediction performance of six distinct activation functions such as Sigmoid, Inverse Square Root Unit (ISRU), Rectified Linear Unit (ReLU), Parametric ReLU (PReLU), Leaky ReLU (LReLU) and Exponential Linear Unit (ELU) are assessed over OSELM based on five regression-based metrics such as MSE, RMSE, MAPE, $R^2$, and RMSLE and five classification-based criteria such as accuracy, precision, recall, f-measures, and g-mean. The overall performance of all the alternate OSELM models has been evaluated using TOPSIS (Hwang and Yoon 1981), PROMETHEE-II (Brans 1982), and VIKOR (Opricovic 1998)-based MCDM techniques to observe and derive the best-ranked activation function for OSELM.

In the next phase to further enhance the performance of OSELM, a hybrid crow search algorithm (hCSA) is incorporated in its training phase. By introducing the chaotic map and mutation operator of DE (Storn et al. 1997) in the position update scheme and catfish behavior in the search process of original CSA (Askarzadeh 2016), the proposed hCSA is able to achieve the right balance between exploration and exploitation improving the convergence. The implementation of hCSA over eight benchmark functions has clearly demonstrated the exceptional optimization capability of the proposed optimization technique compared to classic CSA, DE, PSO, GWO, and TLBO technique. Then the proposed trend predictor model is empirically evaluated over historical data of three stock indices such as BSE SENSEX, S&P 500 and DJIA collected during pre-COVID and COVID timeframes. The prediction models are trained on data from the pre-COVID timeframe and validated on data from both the pre-COVID and COVID timeframes. This analysis assists in identifying the resilience, reliability, and scalability of the prediction models to radical changes in the data. It tries to demonstrate if ML models can adapt to unrealized data and provide appropriate outputs when presented with uncertainty, such as a pandemic. In most of the test cases, the hCSA - OSELM model outperforms the basic ELM, OSELM, and a few cutting-edge baselines such as SVR with different kernels, RBFN, Elman, LR, and MLP-based predictors in terms of all evaluation criteria. In addition, the proposed hCSA-OSELM is compared with the hybrid OSELM models developed using canonical CSA and DE, as well as other benchmark optimization techniques such as PSO, GWO, TLBO, and HS.

The proposed research offers the solutions outlined below, which may address gaps in the literature related to stock index trend prediction and has an advantageous position than existing algorithms in the same area of research.

- An OSELM-based-time series prediction framework is proposed for predicting the upcoming price and to classify the respective trends from the predicted price.
- The activation function selection during design of OSELM is handled as an MCDM problem.
- The performance of six activation functions are evaluated based on five criteria related to stock index price prediction and five criteria related stock index movement classification.
- The best activation function of the predictor is finalized using the ranking obtained from three MCDM
approaches such as TOPSIS, PROMETHEE-II and VIKOR.

- To further optimize OSELM’s performance, a hybrid crow search algorithm (hCSA) is incorporated in its training phase.
- The hCSA is developed by introducing the chaotic map and mutation operator in position update scheme and catfish behavior in the search process of original CSA.
- The prediction and classification outcomes of the proposed hCSA-OSM model have been compared with state-of-the-art baseline models as well as various optimization techniques.
- The suggested model is benchmarked using daily data from three stock indices collected prior to COVID and during COVID to determine whether the proposed model can perform effectively during normal trading days and retain better forecasting in an unforeseen event such as the COVID-19 pandemic.

The remaining portion of this work is structured as follows: Section 2 highlights the relevant work available in the literature; Sect. 3 gives an overview of the related methodologies; Sect. 4 gives a thorough overview of the proposed work and the prediction workflow; Sect. 5 outlines the empirical development and operation of the study; Sect. 6 discusses the work’s observations; and Sect. 7 concludes this work.

2 Literature review

This section explores the literature on financial time-series prediction to have a better insight and awareness of the latest advancement. Although various traditional statistical method-based techniques were proposed in the early years of financial time-series data analysis, they became obsolete and failed to compete with pioneering techniques such as ANN, DT, SVM, and so on. Various ML architectures have provided strong advantages in financial time-series forecasting such as MLP (Ecer et al. 2020), SVM (Guo et al. 2018b), RBFN (Dash and Dash 2015a; Shen et al. 2011), etc. The prediction model advocated by Dash and Dash (2015a) used RBFN along with different basis function for predicting the stock index trends in BSE SENSEX and S&P 500 stock indices. Authors found that RBFN with Inverse multi-quadratics basis function has better performance than other competing models for stock trend prediction. The authors proposed a SVR-based stock price forecasting framework to predict five high-frequency data of Shanghai Stock Exchange in Guo et al. (2018b). It is demonstrated that the proposed adaptive SVR model outperforms regular SVR and backpropagation neural networks (BPNN). A combined hybrid model incorporating MLP and PSO is suggested by Ecer et al. (2020) to predict Borsa Istanbul 100 index price and found better accuracy with reduced processing time as compared to basic MLP and Genetic Algorithm-based MLP. In a recent work, Wang et al. (2021) have presented a stock index forecasting model using Stock index forecasting using Elman network. It is reported that Elman with direct input–output connection helps in improving accuracy, while reducing network complexity and computational cost. Recent work by researchers (Wu et al. 2021b) proposed a trading suggestion system based on Synergetic LSTM-GA. Their experimental outcomes over five Taiwan stocks revealed that LSTMLI-GA framework was able to achieve a higher profit margin as compared to other models. Day-ahead stock price prediction framework by authors (Wu et al. 2021a) combined CNN and LSTM to form a hybrid predictor framework. The authors reported that the proposed combined framework is able to achieve better performance and reduced loss of accuracy as compared to individual models when implemented upon ten stocks from USA and Taiwan markets. A PSO-based deep recurrent neural network-based stock index price predictor framework was reported to achieve 60% success rate for Dow Jones and Nikkei stock index datasets (Bas et al. 2022). Furthermore, the proposed LSTM-ANN outperformed Pi-Sigma ANN and simple recurrent neural network. Authors in another recent study employed Google trends keyword search data to investigate the association between search volume on Google trends and the Taiwan Weighted Stock Index (Fan et al. 2021). By comparing the outcomes of the ANN, SVM, and DT algorithms, it was observed that there was a correlation between using company names from the Taiwan 50 Index as search keywords and the rise and fall of the stock index price. Furthermore, ANN performed superior to baselines for the financial market prediction.

Although these ML techniques can produce accurate predictions, they suffer from drawbacks such as multiple parameters tuning, iterative learning schemes, complex network architecture, intricate implementation steps, and so on. All of these drawbacks were addressed in Huang et al. (2006, 2011) in which the authors proposed a faster learning algorithm known as ELM. The literature related to ELM and its variants over time-series as well as other areas of application suggests that ELM is less complex and has better generalization ability than pre-existing ML models (Huang et al. 2011). The gold price predictor framework based on ELM by Sivalingam et al. (2016) outperformed traditional feed forward neural network, RBFN and Elman networks. The hybrid model incorporating EMD, PSR and ELM outperformed Naive Random Walk, ARIMA, BPNN, and other EMD, PSR, and ELM configurations (Yang and Lin 2017). The hybrid ELM architecture developed by (Wang et al. 2018a) improves crude oil market forecasting.
Even though ELM has consistently provided better prediction since its conception, there still existed certain scope for improvement in it. Its sequential implementation and single-pass output weight calculation required a makeover. As a result, Huang et al. presented an OSELM, which is able to train the network with chunk-by-chunk training data while iteratively updating the output weights of the network using recursive least square algorithm (Huang et al. 2005). OSELM outperforms state-of-the-art algorithms in terms of generalization capability while learning at a considerably faster rate (Guo et al., 2018a).

A few critical aspects of ELM as well as OSELM, such as the initialization of optimal input weights and the selection of an appropriate activation function, have piqued the interest of researchers. Studies have suggested a large number of hybrid ELM frameworks that use various nature-inspired algorithms to tune the ELM input weights which can yield better performance than basic ELM models (Das et al. 2019; Dash et al. 2014; Yu et al. 2020). In another study, the authors (Das et al. 2019) claimed that by incorporating firefly algorithm with evolutionary framework with OSELM can yield desired result for stock index prediction. The hybrid model, integrating EWT, GPS-EO-ABC-ELM, and ARIMA advocated by Yu et al. (2020) achieved better performance as compared to other standalone as well as hybrid models. The authors (Nayak and Misra 2020) proposed an optimized ELM framework using chemical reaction optimization (CRO) for stock volatility prediction. Hybrid optimized models similar to the aforementioned models can be found in literature. The details of the reviewed papers of ML-based financial time-series prediction models are provided in Table 1.

Again, from the survey, it is observed that in different studies, different error-based or accuracy-based metrics are used by researchers for assessing the predictive performance of their advocated models. However, it is undefined whether one error metric outweighs another in terms of use cases and applicability. A systematic survey on recent trends in stock market prediction by Jiang (2021) revealed that researchers have used various classification as well as regression metrics to evaluate the performance of stock market prediction models. As suggested by (Kou et al. 2012; Samal and Dash 2021a), if the performance of a model is evaluated over multiple criteria, a robust performance evaluation system can be generated. To empirically authenticate their proposal, the authors (Kou et al. 2012) evaluated the performance of 11 different classification schemes over 10 performance evaluation criteria using MCDM techniques such as TOPSIS, ELECTRE, grey relational analysis, VIKOR, PROMETHEE. In another work, Mehdiyev et al. (2016) suggested PROMETHEE-based MCDM framework for evaluating classification algorithms. The authors (Dash et al. 2019) suggested a weighted voting ensemble of classifier-based stock index price movement prediction framework. The weights of individual base classifiers were assigned based on their respective TOPSIS ranking. More recently in Samal and Dash (2021a), 15 different activation function-based ELM frameworks have been evaluated based on 16 criteria for the prediction of stock index price movements. It is reported that MCDM-based evaluation of classifiers can provide a more robust and reliable outcome. Although MCDM techniques have not seen that much usability in ML model evaluation, their implications can be seen while solving real-world decision-making problems in various fields of research. Unlike previous relevant studies that cover more publications from the computer science and financial prediction disciplines, the authors (Amaral and Costa 2014) suggested a hospital resource management framework based on PROMETHEE-II. In their work seven different actions of hospital staffs were evaluated based on six criteria for managing hospital resources more efficiently. The maintenance supplier performance evaluation model proposed in Tong et al. (2020) uses a fuzzy PROMETHEE-II technique for evaluating four service providers based on nine different criteria. The authors (Hasnain et al. 2020) employed AHP-TOPSIS for evaluating different boilers using various factors to select most suitable industrial boiler for a production plant. In a more recent work, the authors (Hezer et al. 2021) evaluated 100 regions based on six health safety criteria for assessing the safety of individual regions during COVID-19 pandemic. Surveyed papers as well as their respective areas of implications related to MCDM are shown in Table 2. Most of the studies would focus on management, selection or evaluation-based models using MCDM techniques in various real-world domains. However, very less of the studies have focused on evaluating ML models in financial time-series prediction domain.

3 Methodology

3.1 Online sequential extreme learning machine

Huang et al. (2011) developed ELM as a computational intelligence technique for implementing the SLFN depicted in Fig. 1. Its fundamental superiority over other existing traditional artificial intelligence techniques is demonstrated by the faster learning speed, lower computing complexity, and better overall performance on multiclass classification problems.

Consider \( N \) distinct data samples \((X_i, Y_i)\) with \( X_i \) inputs and \( Y_i \) outputs. A basic SLFN with \( L \) hidden neurons and \( f(X) \) being activation function can estimate these \( N \)
Table 1 Literature summary related to financial time-series prediction

| Authors (year)          | Objective of the study                                                                 | Dataset               | Methods/techniques                                                                 | Performance measure | Relevant findings                                                                 |
|-------------------------|----------------------------------------------------------------------------------------|-----------------------|-----------------------------------------------------------------------------------|---------------------|----------------------------------------------------------------------------------|
| Park and Shin (2013)    | Stock prediction using a semi-supervised learning (SSL) algorithm                     | KOSPI200 listed companies | SSL, SVM, ANN, Buy-and-Hold                                                       | AUC, ROI            | SSL is beneficial for forecasting non-stationary stock market data based on volatility of the stock prices of other corporations and the economic index |
| Guo et al. (2018b)     | High-Frequency Stock Price Forecasting using SVR                                        | 5 high frequency SSE data | BPNN, SVR, AD_SVR                                                                 | RMSE, MAPE, MAD     | AD_SVR outperforms classical SVR and BPNN in terms of adaptability and prediction results |
| Vilela et al. (2019)   | Financial series forecasting using SVR                                                  | IBrXIndexado equity fund | K-Means, Hard fuzzy C-Means, Soft fuzzy C-Means, HM, SVM MLP, CRB                  | MAPE, RMSE, Theil’s U | Clustering-based SVRs are more accurate than benchmark model                      |
| Ecer et al. (2020)     | Stock Price Index Prediction novel ML model                                             | BIST 100 index         | MLP, MLP-GA, MLP-PSO                                                               | RMSE, MAPE, Correlation coefficient | MLP-PSO is the most suitable model among MLP and MLP-GA considering both processing time and accuracy |
| Ravi et al. (2017)     | Hybrid MLP-based financial time-series predictor                                       | JPY, GBP, EUR to USD and Gold price | Various models using Chaos, MLP, NSGA-II, PSO, MOPSO                              | MSE Dstat Theil’s U | The Chaos + MLP + NSGA-II hybrid outperformed the other hybrid models, demonstrating its superior ability as a financial time series predictor |
| Dash and Dash (2015a)  | Stock trend prediction using RBFN                                                      | BSE SENSEX and S&P 500 | RBFN with 7 different basis functions                                               | Accuracy and F-measure | RBFN with Inverse multi quadrics basis function has better performance than competing models |
| Shen et al. (2011)     | Stock indices forecasting using optimized RBFN                                         | Shanghai Composite Indices | RBF optimized with AFSA, GA, PSO; ARIMA, BP, SVM                                   | forecast error       | RBF optimized by AFSA is simple prediction model producing better and robust result |
| Wang et al. (2021)     | Stock index forecasting using ELMAN network                                            | SSE, KOSPI, Nikkei225, SPX | 8 ELMAN networks with different configurations, MLP                              | RMSE, MAE, MAPE     | ELMAN with direct input–output connection helps in improving accuracy, while reducing network complexity and computational cost |
| Bas et al. (2022)      | Stock index price forecasting                                                         | Dow Jones, Nikkei Index | LSTM-ANN, Pi-Sigma ANN, simple recurrent neural network                             | RMSE                 | PSO-based deep LSTM-ANN was able to achieve success rate of 60% in both datasets |
| Fan et al. (2021)      | Deep learning-based financial market prediction                                       | Taiwan 50 Index        | ANN, SVM, DT (J48, CART)                                                           | Pearson correlation test, profit analysis | Proposed framework showed promising outcomes and ANN-based predictor outperformed other models |
| Xue et al. (2018)      | l2,1-norm with Random Fourier Mapping ELM-based financial predictor                    | TianChi and BCS        | SVM, ANN, EI-ELM, SaE-ELM, OP-ELM, PCA-ELM                                        | NMSE                | The l2.1 RF-ELM-based predictor outperforms regular neural networks and other ELMs in terms of capital return and liquidity risk management |
| Yang and Lin (2017)    | Exchange Rates Forecasting using hybrid models                                         | USD, EUD, GBP, and AUD to TWD | Naive Random Walk, ARIMA, BPNN, ELM, variants of ELM, EMD and PSR                  | MAPE, MAE, RMSE    | EMD + PSR + ELM hybrid models improve prediction by efficiently capturing the irregular movements in the data |
| Wang et al. (2018b)    | Development of hybrid models for predicting the price volatility of crude oil’s futures market | Crude oil price        | BEMD + ELM, ELM-Fil and ELM-Rec, AR, NN, SV                                      | RMSE, MAE, MAPE, RMSE | The suggested hybrid ELM framework improves forecasting of the crude oil market by utilizing data from numerous internet sources |
samples with error close to zero if there exists \( W_i, b_i \) and \( \beta_i \) then;

\[
\sum_{i=1}^{L} f(W_i \cdot X_j + b_i) \beta_i = Y_j; \quad j = 1, 2, 3, ..., N,
\]

if \( h_{ij} = W_i \cdot X_j \), then

\[
\sum_{i=1}^{L} f(h_{ij} + b_i) \beta_i = Y_j,
\]

where \( W_i \) input weight connecting input neurons and \( i \)th hidden neuron, \( b_i \) bias of \( i \)th hidden neuron and \( \beta_i \) output weights connecting \( i \)th hidden neuron and output neurons.

The compact form of Eq. (1) can be represented as,

\[
H \beta = Y,
\]

where \( H \) hidden layer outputs. Also, from Eq. (2) \( \beta \) can be calculated as follows;

\[
\tilde{\beta} = H^\dagger Y = (H^T H)^{-1} H^T Y,
\]

where \( H^\dagger \) Moore–Penrose pseudo inverse of \( H \).

In contrast to single-iteration batch learning of ELM, Huang et al. (2005) introduced sequential learning-based ELM, namely online sequential ELM (OSELM) where the training data are fed to the network sequentially or chunk by chunk. Unlike ELM’s single batch learning, OSELM has two phases; initialization and sequential learning. A \( \text{rank}(H_0) = L \) is required during the initialization phase of OSELM, where \( H_0 \) hidden layer output matrix during initialization phase. Also, \( N_0 \) being the training data sample during the initialization phase must follow \( N_0 \geq L \).

**Initialization phase:** In this phase a portion of the training data \((X_i, Y_i)_{i=1}^{N_0}\) is fed to the network such that \( N_0 \geq L \).

### Table 1 continued

| Authors (year) | Objective of the study | Dataset | Methods/techniques | Performance measure | Relevant findings |
|---------------|------------------------|---------|--------------------|---------------------|------------------|
| Sivalingam et al. (2016) | Gold price prediction using ELM | Gold price | ELM, Feed forward networks without feedback, Feed forward back propagation networks, RBFN, ELMAN | Accuracy | When compared to other networks, the ELM algorithm achieves a 3% boost in training and testing efficiency. ELM achieves a training success rate of 97.5% and a testing success rate of 93.82% |
| Das et al. (2020) | Currency exchange prediction using hybridized ELM | USD to INR and EURO | Hybrid models of ELM, NN, FLANN | MSE, MAPE, MAE, Theil’s U, ARV | ELM trained using the Jaya optimization algorithm outperforms other prediction strategies |
| Yu et al. (2020) | Hybrid ELM-based Financial Time Series Forecasting | Treasury bill rates, S&P 500 index, Closing index | Hybrid models of AR, LSTM, ANN integrated with ABC | RMSE, MAPE, MAE, | The hybrid model incorporating EWT, GPS-EO-ABC ELM, and ARIMA provides the best predicting performance |
| Das et al. (2019) | Stock market prediction using optimized OSELM | BSE Sensex,NSE, S&P 500 and FTSE index | ELM, OSELM, RBPNN, PCA, FA | RMSE, MAPE, MAE, Theil’s U, ARV | ELM, OSELM, and RBPNN are significantly outperformed by the hybrid firefly algorithm with evolutionary framework-based OSELM forecasting |
| Nayak and Misra (2020) | Stock volatility prediction using optimized ELM | BSE indices | ELM models integrated with CRO, GD, PSO, GA | MAPE, ARV | ELM optimized using CRO is effective for training an SLFN for stock volatility forecasting and outperforms other optimized ELMs |
| Das et al. (2021) | Optimized ELM-based stock market price forecasting | DJI, HIS,IXIC, N 100, NSEI, RUT, GDAXI | PGCSA-ELM, CSA-ELM, ELM, GARCH-DAN2, DAN2, GARCH-MLP, MLP | MAE, MSE, MAAPE, CoV, CORR, Theil’s U | PGCSA-ELM model is suitable tool to predict next day closing price having better performance than other models |
The initialization phase can be summarized in the following steps.

a. Assign initial parameters of the network such as $X_i, W_i,$ and $b_i$.

b. Calculate $H_0$ for initial phase.

c. Calculate output connection weights $\beta^{(0)}$.


d. Update the number of chunks of data $k = 0$.

\[
H_0 = \sum_{i=1}^{L} f(W_i \cdot X_j + b_i); \quad i = 1, 2, ..., L; \quad j = 1, 2, 3, ..., N_0
\]
Sequential learning phase: The sequential learning phase begins for any further chunks of training data and this phase is summarized below:

a. Calculate $H_{k+1}$ for $N_1^{th}$ chunk where $N_1$ begins from $N_0 + 1$.
$$H_{k+1} = \sum_{i=1}^{L} f(W_i \cdot X_{N_0+j} + b_i); \quad i = 1, 2, ..., L; \quad j = 1, 2, 3, ..., N_1$$
(6)

b. Calculate $\beta^{(k+1)}$.
$$\beta^{(k+1)} = \beta^{(k)} + M_{k+1}(Y_i^T - H_{k+1}^T \beta_k)$$
(7)
where $M_{k+1} = M_k - \frac{M_{k+1} \beta_k^T \beta_k}{1 + \beta_k^T M_k \beta_k}$

c. Update the number of chunks of data $k = 1$.

4 Crow search

In 2016, Askarzadeh suggested a nature-inspired algorithm that mimics crows’ food concealing technique (Askarzadeh 2016). One of the most impressive characteristics of crows is their ability to efficiently hide and recall the location of their food. The operation of CSA is based on four basic principles: Crows live in communities, memorize the location of stashed food, pursue other member of their species, and finally, protect their stockpiles from being raided at random. As suggested by a comprehensive review in Hussien et al. (2020), CSA has been utilized to tackle a variety of problems relating to feature selection, image segmentation and electromagnetic optimization because of its simplicity in design and efficiency in problem solving. The working idea of CSA is based on crows’ abilities to hide and retrieve food which is represented below.

1. The first step involves initializing the objective function, decision variables, boundary constraints (Lower bound and upper bound; LB and UB) and adjustable parameters such as crow flock size, crow flight length (fl), awareness probability (AP) and maximum number of iterations (MaxIt).

2. A matrix of size $(n \times d)$ is constructed where $n$ and $d$ represent the number of crows and the number of decision variables, respectively. Since, in the beginning the crows do not have any knowledge regarding the food source, it is assumed that the food sources are distributed randomly. Hence, the memory for each crow (mem) is set to an arbitrary location in the solution space.

3. The value of the fitness function is then computed.

4. Following that, the position of each crow is modified using the equation,
$$pos_{itr+1}^{itr} = \begin{cases} pos_{itr}^{itr} + \text{rand1} \times fl \times (\text{mem}_{itr}^{itr} - pos_{itr}^{itr}); & \text{if rand2} > \text{AP}_{itr} \\ \text{random location}; & \text{otherwise} \end{cases}$$
(8)

5. The objective function for the new position of crow is computed and the memory of the crow is modified with the help of following equation:
$$\text{mem}_{itr}^{itr+1} = \begin{cases} \text{pos}_{itr}^{itr+1}; & \text{if } f(\text{pos}_{itr}^{itr+1}) \text{ is superior to } f(\text{pos}_{itr}^{itr}) \\ \text{mem}_{itr}^{itr}; & \text{otherwise} \end{cases}$$
(9)

6. Steps 4 and 5 are continued until the stopping criteria satisfied and the final memory of the crow that holds the specific location of food is deemed the best solution for the problem.

4.1 Proposed hCSA

Any optimization algorithm must drive equity between exploitation and exploration. The two milestones are in conflict with each other so promoting one would degrade the other. The sheer exploration of the search space does not allow an algorithm to achieve an exact global optimum. Likewise, mere exploitation creates local optima that stagnate and yields approximated optimum with low quality. To achieve the right balance between exploration and exploitation, in this study a new variant of CSA is suggested by introducing the chaotic map and mutation operator of DE (Storn et al. 1997) in the position update scheme and catfish behavior in the search process of original CSA.
4.1.1 Improved position-updating method

The main disadvantage of the CSA algorithm is that the search agents frequently do not pursue the best answer obtained thus far. As stated in Eq. (8), each solution in canonical CSA updates its position based on the location of the best food source acquired so far or to a randomly chosen point in the search space. However, randomly distributing locations based on AP alone drastically reduces CSA performance. Therefore, instead of random position update, a mutation-based position update scheme is adopted from DE algorithm. After analyzing various mutation operators from (Das et al. 2016; Das and Suganthan 2011), the DE/best/1 mutation operator (Das et al. 2016) has been integrated with CSA in proposed hCSA. The proposed improved position update equation for hCSA is as follows.

\[
pos_i^{ir+1} = \begin{cases} 
pos_i^{ir} + f_i^{ir} \times c_2 \times ((c_1 \times \text{mem}_i^{ir}) - ((1 - c_1) \times \pos_i^{ir})); & \text{if rand2} > \Ap_j^{ir} \\ Mpos_i^{ir} = c_1 \times (\pos_{\text{best}}^{ir} + F \times (\pos_{\text{rand3}}^{ir} - \pos_{\text{rand4}}^{ir})) \times (1 - c_2); & \text{otherwise} \end{cases} 
\]

where \(Mpos_i^{ir}\) mutated position vector of crow, \(\text{rand1}, \text{rand2} = \) arbitrary numbers \(\{\text{rand1}, \text{rand2}| 0 < \text{rand1}, \text{rand2} < 1\}\), \(\pos_{\text{rand3}}^{ir}, \pos_{\text{rand4}}^{ir}\) are arbitrarily selected source vectors and \(\text{rand3} \neq \text{rand4} \neq i\).

These processes ensure that all search agents adhere to the previously discovered best solutions and that no sub-optimal arbitrary solutions are generated.

4.1.2 Improved convergence

In the aforementioned improvement Eq. (10), the search agents adjust their position based on the best solution obtained so far using either the canonical CSA or the mutation operator. Despite having a good convergence rate, the proposed modification may not always perform well in identifying global optima since using simply the best solution to pursue may leave a large area of solution space unexplored and may also lead to a local minima problem (Tian et al. 2019). As a result, chaos is added in the hCSA algorithm to overcome this disadvantage and enhance its efficacy. Chaos, in general, is a stochastic, unregulated technique found in nonlinear, non-converging, and constrained non-linear dynamical systems. To introduce chaos into optimization methods, many chaotic maps with various mathematical equations are used. Since chaotic maps behave dynamically, they help the optimization algorithms to explore the search space more effectively and extensively (He et al. 2001). Based on a survey of the literature (Gandomi and Yang 2014; Kohli and Arora 2018), 10 of the most relevant uni-dimensional chaotic maps were applied in the suggested hCSA, and the Logistic chaos map was the most appropriate and best performing map among them, and it has been adopted here. Because chaotic maps induce uncertainty in the feasible region, they improve the CSA algorithm’s convergence rate. As a result, chaos is incorporated into Eq. (10), and the improved position update equation is as follows:

where \(c_1, c_2\) are chaos coefficients.

4.1.3 Renewed search

The underlying concept for incorporating the catfish effect in CSA was inspired by the reaction of sardines when a small quantity of catfish was introduced to them (Chuang et al. 2008). The authors (Chuang et al. 2011, 2008) exploited catfish effect by adding it into PSO, to strengthen the tendency of moving out of the local minima and found effective outcome. The catfish, in competition with the sardines, causes the sardines to move more. Similarly, the introduction of catfish crows causes the other “sardine” crows in hCSA to recommence their search. In other words, catfish crows can steer classic crows confined in local minima into new locations of solution space, potentially leading to an improved location. If the range among flock-best and the neighboring crows is short, each crow will advance a relatively short step in the following iteration. This kind of premature convergence can be avoided by injecting catfish crows and substituting certain percentage of the original crows which have lowest fitness values in the flock.

Further, the detailed flowchart of the hCSA mechanism is illustrated in Fig. 2.
5 Proposed hCSA-OSELM prediction mode

Figure 3 depicts the suggested stock index trend analysis framework. The subsections below depict the workflow of the MCDM-based activation function selection and the hCSA-OSELM-based stock index trend analysis.

5.1 Datasets

The literature on financial time-series prediction shows that recent ML models have surpassed traditional ML strategies in terms of getting accurate predictions. However, these prediction models were implemented over a standard trading setup where the margin of price movement is within a specific price band. So, in a day, the stock price may rise and fall in the given price band, say 5%, 10%, or 20%. However, during an unforeseen event such as a pandemic, the stock market does not follow conventional rules and may fall significantly lower than the predefined price band. So, the question arises whether recent ML models can observe and adapt to these drastic changes in stock price and provide good predictions. Therefore, in this experiment, we have trained and tested the model’s generalization performance using three standard stock indices datasets, namely the BSE SENSEX, the S&P 500, and the DJIA, from January 2014 to January 2021. As the corona
virus (COVID-19) outbreak was declared a global pandemic during this time period, the raw data were initially divided into two parts: data prior to COVID-19, i.e., pre-COVID, and data during the pandemic, i.e., during COVID. Table 3 provides a comprehensive overview of the data. Initially, the pre-COVID data were pre-processed and divided into training and testing sets. The suggested model’s testing performance assessment is carried out in two phases: the first phase includes pre-COVID data samples and the second phase comprises of during COVID data samples. Although the World Health Organization officially designated COVID-19 a worldwide pandemic in March 2020, for the purposes of evaluating the robustness and adaptability to uncertainty in data of our proposed prediction model, in this experiment we selected January 2020 to January 2021 to be the COVID period. In March 2020, the US stock market tripped the circuit breaker four times in 10 days. Since its establishment in 1987, the breaker has only been activated once, in 1997. Along with the US stock market fall, stock markets in Europe and Asia have also plummeted (BBC News 2020). Therefore, in this experiment, we used two major indices from the US market, the S&P 500 and DJIA, as well as one index from the Asian market, the BSE SENSEX, as our benchmark data to determine whether the ML-based predictive models are capable of predicting upcoming market trends. If this is the case, the question is whether they will be able to maintain the same level of performance in the event of an unforeseen event, such as the COVID-19 pandemic. The datasets analyzed and experimented during the current study are available at https://www.wsj.com/, https://finance.yahoo.com/, openly.

### 5.2 Input preparation

Six popular technical indicators were used in this experiment, including the simple moving average (SMA), moving average convergence and divergence (MACD), Stochastic $K$ and $D$, relative strength index (RSI), and William’s $R$ percent from (Dash et al. 2019). The previous
day’s closing prices with the six technical indicators are used as input parameters for the OSELM framework. The Min–Max normalization technique is used on these seven input variables to transform their values between 0 and 1.

### 5.3 Output preparation

The output of the prediction model is the closing price of the stock index that has continuous values. Furthermore, a moving average-based closing price is calculated using the procedure given in Fig. 4 over both predicted and actual close prices. These new close prices are further used to calculate the stock index movement by employing the procedure in Fig. 4. If the $d$th day close price is more than that of $(d-1)$th day, the movement is considered as upward and if the $d$th day close price is less than that of $(d-1)$th day, the movement is considered as downward. The observed upward and downward movements are assigned the values 1 and 0 respectively, resulting in a binary classification problem. In Fig. 4, CP = close price, MAC = moving average close price, ws = window size, n = total number of samples, Movement = upward or downward movement.

### 5.4 MCDM-based activation function selection for OSELM

The activation function used on the hidden layer input is the most significant component in an OSELM network which must be properly considered. It is, however, impossible to guarantee that a single activation function is ideal for any and all datasets. Initially, the closing price is predicted in Phase II using Sigmoid, ISRU, ReLU, PReLU, LReLU, and ELU activation functions as described in Table 4 with OSELM and the movement is determined using the process described in Sect. 4.3. Later, the performance of OSELMs with distinct activation functions were evaluated for both stock index price prediction and stock index movement classification utilizing prediction and classification measures given in Table 5. Since this experiment evaluates the performance of six different activation function-based OSELMs based on ten criteria, it is addressed as a MCDM problem. Three popular MCDM approaches such as TOPSIS, PROMETHEE-II, and VIKOR are used to assess this MCDM problem. The MCDM approach can assess and propose the best activation function among all of these models, as well as assign appropriate ranks to each activation function. These

### Table 4 Activation functions

| Activation function | Abbreviation | Formula |
|---------------------|--------------|---------|
| Sigmoid             | AF1          | $f(x) = \frac{1}{1+e^{-x}}$ |
| ISRU                | AF2          | $f(x) = \frac{x}{\sqrt{1+x^2}}$ |
| ReLU                | AF3          | $f(x) = \begin{cases} 0, & \text{for } x < 0 \\ x, & \text{otherwise} \end{cases}$ |
| PReLU               | AF4          | $f(x) = \begin{cases} x, & \text{for } x < 0 \\ x, & \text{otherwise} \end{cases}$ |
| LReLU               | AF5          | $f(x) = \begin{cases} 0.01x, & \text{for } x < 0 \\ x, & \text{otherwise} \end{cases}$ |
| ELU                 | AF6          | $f(x) = \begin{cases} a(e^x - 1), & \text{for } x < 0 \\ x, & \text{otherwise} \end{cases}$ |
rankings will assist in determining which activation functions are superior and which are inferior. The considered MCDM approaches are adopted from (Hwang and Yoon 1981) for TOPSIS, (Brans 1982) for PROMETHEE-II, and Opricovic (1998) for VIKOR. The ranking obtained after employing these MCDM techniques over the predictors provides an easy-to-interpret system through which the most optimal activation function for stock index trend analysis can be selected.

In literature, the prediction or classification performance of any given framework has been evaluated based on any single error-based or accuracy-based metrics. However, employing multiple performance evaluation metrics to evaluate these frameworks can produce a more robust and reliable framework. Therefore, assessing the performance of multiple activation functions based on multiple criteria can be viewed as an MCDM problem. So, in this experiment, the performance evaluation and ranking of given activation functions are carried out using three MCDM approaches; TOPSIS, PROMETHEE-II, and VIKOR. The primary objective for modeling the proposed framework as an MCDM problem is that any given activation function may or may not give the best outcome regarding all performance measures. For example, one activation function may give the best MSE, while another may give the best accuracy. Hence, it is difficult to assess these two activation functions solely on a single criterion. By employing MCDM, we can assess these two activation functions based on MSE and accuracy and find a ranking-based system that gives us the best among these two.

### 5.5 OSELM weight update using hCSA

Since the fundamental problem with OSELM is the randomly initialized input layer weights, researchers in the literature have used various optimization strategies such as DE, PSO, CSA, and so on. This paper offers a novel hybrid CSA algorithm that incorporates the mutation scheme of DE (Das et al. 2016; Storn et al. 1997) as well as chaos factor (Gandomi and Yang 2014; He et al. 2001) and cat-fish behavior (Chuang et al. 2008) into the CSA (Askarzadeh 2016) position update. The proposed hCSA was used to tune the OSELM input layer weights for stock index trend prediction. The hCSA seeks to optimize the input weights W in Eq. (1) of OSELM during training, which has then been used for testing to evaluate the generalization performance of suggested hCSA-OSELM for stock index trend prediction.

### 5.6 Predicted price to movement conversion

The forecasted closing price continuous values are transformed into discrete values of 1 or 0 using a sliding window-based stock index movement estimate mechanism depicted in Sect. 4.3. The transformed values represent the stock index’s movements, with 1 signifying upward movement and 0 denoting downward movement, respectively. This projected movement is then used to classify stock index movements by comparing it to the actual movement.

### 6 Experimental design

We do a series of experiments on three stock index benchmark datasets: the BSE SENSEX, the S&P 500, and the DJIA. The datasets are discussed in detail in Sect. 4.1. The empirical authentication for the proposed hCSA, as well as the baselines in use for comparison and the experimental methodology, are discussed in the following subsections. The proposed hybrid model is developed on a machine with an Intel (R) Core (TM) i7-8750H CPU @ 2.20 GHz, 16.00 GB RAM, and the WIN11 64-bit operating system using MATLAB R2020a.

| Table 5 Performance evaluation metrics |  |
|-----------------|-----------------|
| **Prediction metrics** | **Classification metrics** |
| $\text{MSE} = \frac{1}{n} \sum_{i=1}^{n} (v_i - \hat{v}_i)^2$ | $\text{PM1 \quad \text{Accuracy} = \frac{TP}{TP+FP}$ |
| $\text{MAPE} = \frac{100}{n} \sum_{i=1}^{n} \left| \frac{v_i - \hat{v}_i}{v_i} \right|$ | $\text{AM1}$ |
| $R^2 = 1 - \frac{\sum_{i=1}^{n} (v_i - \hat{v}_i)^2}{\sum_{i=1}^{n} (v_i - \bar{v})^2}$ | $\text{PM2 \quad \text{Precision} = \frac{TP}{TP+FP}$ |
| $\text{MAE} = \frac{1}{n} \sum_{i=1}^{n} |v_i - \hat{v}_i|$ | $\text{AM2}$ |
| $\text{RMSLE} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (\log(v_i) - \log(\hat{v}_i))^2}$ | $\text{PM3 \quad \text{Recall} = \frac{TP}{TP+FN}$ |
| $\nu$ actual value, $\hat{v}$ predicted value, $\bar{v}$ mean of all values, $n$ total number of samples | $\text{AM3}$ |
| $\nu$ actual value, $\hat{v}$ predicted value, $\bar{v}$ mean of all values, $n$ total number of samples | $\text{PM4 \quad F - measures = \frac{PM3}{PM4}$ |
| $\nu$ actual value, $\hat{v}$ predicted value, $\bar{v}$ mean of all values, $n$ total number of samples | $\text{AM4}$ |
| $\nu$ actual value, $\hat{v}$ predicted value, $\bar{v}$ mean of all values, $n$ total number of samples | $\text{PM5 \quad G - mean = \sqrt{\text{Specificity} \times \text{Sensitivity}$ |
| $\nu$ actual value, $\hat{v}$ predicted value, $\bar{v}$ mean of all values, $n$ total number of samples | $\text{AM5}$ |
6.1 Empirical analysis of proposed hCSA

The suggested hCSA’s performance is compared to that of classic CSA, DE, PSO, GWO, and TLBO using the eight benchmark functions listed in Table 6. The following are the fundamental parameters of these algorithms: the population size has been set to 100, with a maximum iteration of 1000. Table 7 displays the mean, standard deviation (σ), and variance (Var) of solutions acquired across twenty independent runs of individual algorithms. In the case of all eight benchmark functions, the suggested hCSA demonstrated exceptional optimization capability, as shown in Table 7. However, in terms of $F_{10}$, it lags beneath TLBO. Nonetheless, in all eight test functions, hCSA outperforms both traditional CSA and DE. Take $F_1$ as an example; when compared to the original CSA, the average solution of twenty trial runs is reduced by 1.07. Aside from that, the standard deviation is reduced by 1.057 and the variance of the solutions is reduced by 1.061649 for CSA, indicating that the suggested algorithm can yield more accurate solutions consistently and reliably than conventional CSA. The suggested algorithm’s average performance is the best among all other algorithms, indicating hCSA’s ability to approximate to the most ideal solutions. The improved position updating scheme and catfish-based renewed search introduced in hCSA assist crows in avoiding local optima, improving their positions, improving convergence towards the optimal food source, and renewing the search if necessary. As a result, the hCSA offers impressive global search capabilities. Furthermore, the hCSA can approach ideal solutions with greater stability and constancy, demonstrating the algorithm’s superiority. As a result, the hCSA could be an excellent solution to improve both local and global search capabilities.

6.2 Parameter setup and experimentation

As mentioned in earlier sections, certain parameters influence the performance of ELM as well as OSELM: hidden neurons and activation function. The results obtained from 20 individual runs on ELM and OSELM, the best set of hidden neurons is found to be 9 for all three benchmark datasets for pre-COVID and COVID timeframe. For convenience and better comparison, the evaluation results of pre-COVID and COVID timeframe are averaged and tried for MCDM-based ranking. The average results of the 20 runs are depicted in Tables 8 and 9 for ELM and OSELM, respectively. In Tables 8 and 9, the MCDM rankings obtained from TOPSIS, VIKOR and PROMETHEE-II are denoted as R1, R2 and R3, respectively, and these rankings obtained from TOPSIS, VIKOR and PROMETHEE-II are denoted as R1, R2 and R3, respectively, and these rankings show that AF6 i.e., ELU is the best performing activation function for both ELM and OSELM which is highlighted in bold. Moreover, OSELM has an upper-hand over ELM in stock index trend prediction which validates our initial proposal of using OSELM instead of ELM for stock index trend prediction.

### Table 6 Description of benchmark functions

| Name | Formula | Range | $F_{\min}$ |
|------|---------|-------|------------|
| Ackley | $F_1(x) = -20 \exp \left( -0.2 \sqrt{\frac{1}{n} \sum_{i=1}^{n} x_i^2} \right) - \exp \left( \frac{1}{n} \sum_{i=1}^{n} \cos(2\pi x_i) \right) + 20 + e$ | $-32, 32$ | 0 |
| Bohachevsky | $F_2(x) = x_1^2 + 2x_2^2 - 0.3 \cos(3\pi x_1) - 0.4 \cos(4\pi x_2) + 0.7$ | $-100, 100$ | 0 |
| Griewank | $F_3(x) = \sum_{i=1}^{n} x_i^2 - \frac{1}{4000} \sum_{i=1}^{n} \cos \left( \frac{x_i}{\sqrt{n}} \right) + 1$ | $-600, 600$ | 0 |
| Periodiceval/Price’s | $F_4(x) = 1 + (\sin(x_1))^2 + (\sin(x_2))^2 - 0.1 \exp(-x_1^2 - x_2^2)$ | $-10, 10$ | 0.9 |
| Rastrigin | $F_5(x) = 10n + \sum_{i=1}^{n} \left[ x_i^2 - 10 \cos(2\pi x_i) \right]$ | $-5.12, 5.12$ | 0 |
| Rosenbrock | $F_6(x) = \sum_{i=1}^{n-1} \left[ 100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2 \right]$ | $-5, 10$ | 0 |
| Sphere | $F_7(x) = \sum_{i=1}^{n} x_i^2$ | $-5.12, 5.12$ | 0 |
| Zakharov | $F_8(x) = \sum_{i=1}^{n} x_i^2 + \left( \sum_{i=1}^{n} 0.5ix_i \right)^2 + \left( \sum_{i=1}^{n} 0.5ix_i \right)^4$ | $-5, 10$ | 0 |

6.3 Baselines

The predictive performance of the proposed hCSA- OSELM is compared to those of state-of-the-art baseline prediction models. To discover the best combinations,
several tuning parameters of individual baselines, as presented in Table 10, are evaluated. A brief overview of each baseline model is given below.

a. **MLP**: It is perhaps the most prevalent and adaptable feedforward network architecture, with numerous applications in a variety of fields to handle classification and regression problems (Ecer et al. 2020; Ravi et al. 2017). It usually consists of three layers: an input layer, a hidden layer, and an output layer. It is trained using the well-known back-propagation algorithm, which iteratively updates the weights linking the layers. Once trained, the network can anticipate new and previously unknown data.

b. **SVR**: It is a supervised learning technique that predicts continuous values rather than discrete values, based on the same concepts as SVM for classification (Guo et al. 2018b; Vilela et al. 2019). The primary principle of SVR is to identify the optimum fit line or hyperplane with the most points. Unlike other regression models, which aim at reducing the difference between the real and predicted values, the SVR attempts to fit the best line within a given tolerance margin (epsilon).

c. **RBFN**: A RBFN has the same architecture as a simple feed forward network with three layers. The neurons of hidden layer are also referred to as radial centers and are employed to calculate the output of hidden layer nodes. Following that, each node in the hidden layer generates an activation based on relevant radial basis function. Finally, each node in the output layer computes a weighted linear combination of the hidden nodes’ activations. RBFN’s hidden layer is often nonlinear, whilst the output layer is linear (Dash and Dash 2015a; Fernández-Navarro et al. 2012; Shen et al. 2011).

d. **Elman**: The Elman neural network is a recurrent network with four layers: input-hidden-undertaking-output (Wang et al. 2021). The output of the hidden layer is stored in the undertaking layer which acts as a delay operator.

e. **LR**: The linear regression procedure identifies a linear relationship between dependent and independent variables, i.e., how the value of the dependent variable

---

**Table 7 Results of benchmark functions**

|         | PSO   | DE    | CSA   | GWO   | TLBO  | hCSA  |
|---------|-------|-------|-------|-------|-------|-------|
| **F1**  | Mean  | 7.12E-02 | 9.94E-11 | 1.07E+00 | 4.55E-15 | 2.22E-15 | 0.00E+00 |
|         | σ     | ± 0.318312 | ± 1.93E-11 | ± 1.057 | ± 1.53E-15 | ± 1.21E-30 | ± 0 |
|         | Var   | 0.096256 | 3.54E-22 | 1.061649 | 2.21E-30 | 1.40E-60 | 0 |
| **F2**  | Mean  | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 |
|         | σ     | ± 0   | ± 0   | ± 0  | ± 0  | ± 0  | ± 0  |
|         | Var   | 2.60E-06 | 3.78E-21 | 1.15E-05 | 0  | 0  | 0  |
| **F3**  | Mean  | 3.70E-04 | 0.00E+00 | 1.41E+11 | 2.22E-03 | 0.00E+00 | 0.00E+00 |
|         | σ     | ± 0.00165 | ± 6.319E-11 | ± 0.00347 | ± 0 | ± 0 | ± 0 |
|         | Var   | 0.00047 | 0  | 1.15E-05 | 0  | 0  | 0  |
| **F4**  | Mean  | 9.05E-01 | 9.00E-01 | 9.00E-01 | 9.00E-01 | 9.00E-01 | 9.00E-01 |
|         | σ     | ± 0.02236 | ± 0  | ± 0  | ± 0  | ± 0  | ± 0  |
|         | Var   | 0 | 0  | 0  | 0  | 0  | 0  |
| **F5**  | Mean  | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 |
|         | σ     | ± 0   | ± 0  | ± 0  | ± 0  | ± 0  | ± 0  |
|         | Var   | 0 | 0  | 0  | 0  | 0  | 0  |
| **F6**  | Mean  | 9.58E+00 | 2.66E+01 | 2.02E+01 | 1.59E+01 | 1.01E+00 | 1.81E+01 |
|         | σ     | ± 2.938211 | ± 16.33275 | ± 15.8093 | ± 0.777636 | ± 1.986258 | ± 0.400129 |
|         | Var   | 8.201431 | 253.4207 | 237.437 | 0.5744 | 3.74795 | 0.15209 |
| **F7**  | Mean  | 4.26E-133 | 9.19E-21 | 6.04E-07 | 1.61E-111 | 7.61E-188 | 0.00E+00 |
|         | σ     | ± 9.40E-133 | ± 4.93E-21 | ± 4.665E-07 | ± 3.67E-111 | ± 0 | ± 0 |
|         | Var   | 8.40E-265 | 2.31E-41 | 2.06E-13 | 1.28E-221 | 0 | 0 |
| **F8**  | Mean  | 2.09E-19 | 2.40E+01 | 3.33E+01 | 2.23E-58 | 1.39E-41 | 0.00E+00 |
|         | σ     | ± 2.86E-19 | ± 5.47 | ± 14.22 | ± 6.34E-58 | ± 1.39E-41 | ± 0 |
|         | Var   | 7.77E-38 | 28.51927 | 192.1989 | 3.82E-115 | 1.86E-82 | 0 |
changes in response to the variation in the value of the independent variable. The linear regression model generates a slanted straight line that represents the relationship between the variables.
6.4 Result analysis and performance comparison

This section depicts the results obtained from proposed hCSA-OSELM for stock index trend prediction over three benchmark datasets having two timeframes, comparison of the result with other state-of-the-art baseline prediction models and optimization algorithms. Table 11 represents the most optimal parameter combination for the proposed hCSA-OSELM which is used for analysis.

In this study, six optimization algorithms and nine prediction models are utilized as competitors of hCSA-OSELM. As illustrated in Fig. 3, in phase I, the seven training inputs: six technical indicators and previous day closing price based on three benchmark stock index time series data are normalized. In phase II, the normalized data are then fed to OSELM for stock index trend prediction. Here six distinct activation function-based OSELMs are evaluated and ranked based on prediction and classification performance metrics using three MCDM techniques. In phase III, the inputs and the most suitable activation function obtained from phase I and II are used in hCSA-OSELM. In Fig. 5, the predicted stock index close price during training as well as testing is displayed alongside the actual close price. For both pre-COVID and COVID timeframes, the actual and proposed data series are plotted relatively close to each other for both training and testing data samples. The examination of correlation between the actual and predicted data points are illustrated in Fig. 6 as

Table 10 Parameter configuration of prediction models for benchmark datasets

| Framework | Specification | Value |
|-----------|---------------|-------|
| ELM       | Hidden neurons | \( \frac{2^n - 1}{2^n} \) |
|           | Activation function | ELU |
| OSELM     | Hidden neurons | \( \left( N - 2, N - 1, N, \ldots, \frac{2^n}{N} \right) \) |
|           | Activation function | Sigmoid, ISRU, ReLU, PReLU, LReLU, ELU |
| LR        | Model type | Linear, Interactions |
| MLP       | Hidden layer size | 20, 30, 40, 50 |
|           | Training function | GD, LMA, BR, BPCG, RBP, OSSBP |
|           | Epoch | 1000 |
| RBFN      | Spread | 5, 10, 20 |
|           | Maximum neurons | 5, 10, 15, 20 |
| L-SVR     | Box constraint | 0.01, 0.1, 0.5 |
| G-SVR     | Epsilon | 0.0001, 0.001, 0.01 |
| P-SVR     | Order | 1, 2, 3 |
| Elman     | Hidden layer size | 5, 10, 15 |
|           | Training function | GD, LM, BR, BPCG, RBP, OSSBP |

\( GD \) gradient descent; \( LMA \) Levenberg–Marquardt algorithm; \( BR \) Bayesian regularization; \( BPCG \) conjugate gradient with Beale–Powell restarts algorithm; \( RBP \) resilient backpropagation; \( OSSBP \) one step secant backpropagation; \( N \) number of input layer neurons, \( L-SVR \) linear SVR; \( G-SVR \) Gaussian SVR; \( P-SVR \) polynomial SVR

Table 11 Initial parameters for hCSA-OSELM

| Framework | Specification | Value |
|-----------|---------------|-------|
| OSELM     | Hidden neurons | \( \frac{2^n - 1}{2^n} \) |
|           | Activation function | ELU |
| Hybrid CSA | Population | 30 |
|           | Max iteration | 200 |
|           | Dimension | \( h_n \times N \) |
|           | Decision variable interval [LB, UB] | \([-0.5, 0.5]\) |
|           | Chaos map | Logistic map (Chuang et al. 2011; He et al. 2001) |
|           | DE mutation scheme | Canonical DE algorithm: DE/best/1 (Das et al. 2016; Das and Suganthan 2011) |

\( n \) number of input variables or features; \( h_n \) number of hidden layer neurons; \( LB \) and \( UB \) lower and upper bound
scatter plot which shows the distribution of the points coincide relatively closer to the trend line for all benchmark stock index data in both time frames. Although the proposed model demonstrates very promising outcomes in Figs. 5 and 6, there are few erroneous predictions. Thus, to reduce the overestimations or underestimations of closing prices during forecasting, the proposed must still be improved.
6.5 Evaluation of benchmark optimization algorithms

This section evaluates the suggested hCSA’s generalization ability to six other optimization methods: PSO, DE, CSA, GWO, HS, and TLBO used to train OSELM-based stock index predictor. The maximum number of iterations for all optimization algorithms is 200, and the population size is set at 30. To ensure a fair comparability, we’ve used the same OSELM parameter combination for each optimization algorithm. Table 12 summarizes the findings obtained by these optimization algorithms for three benchmark datasets with two distinct timeframes over 20 runs. In addition, the converging graphs of OSELM supervised by the hCSA, as well as other benchmark optimization techniques are illustrated in Fig. 7, indicating that the MSE of hCSA-OSLM reduces to a relatively lower value in few iterations of the algorithm for all benchmark datasets. Table 12 also shows the best, mean MSE values as well as the respective standard deviation produced by OSELM trained by individual benchmark optimization algorithms. In Table 12, the best results are marked in bold face and the Improv represents the percentage improvement in result and | symbol represents an improved result as compared to the second-best model in comparison. The results show that hCSA outperforms other optimization schemes with an improved mean value as well as best result value over all three benchmark stock index datasets for both pre-COVID and COVID timeframe. This suggests the efficiency and usability of hCSA for training OSELM-based predictor.

6.6 Comparison with other prediction approaches

As baselines in this work, nine state-of-the-art prediction models are used: ELM, OSELM, LR, MLP, RBFN, Elman and SVR with various kernel functions: linear, Gaussian, polynomial kernel (L/G/P-SVR). The parameters that are most ideal for each method given in Table 13 are used, which are found using the parameter selection approach presented in Table 10. To provide a reliable and unbiased performance, each model is evaluated 20 times. Based on the results of the independent runs, SVR with polynomial kernel outperformed all other kernel functions; thus, only P-SVR results is used for comparison. Table 14 presents the best, average, and standard deviations of the prediction models, as determined by the average value of two evaluation metrics: PM1 (MSE) for prediction and AM1 (Accuracy) for classification over 20 repeated experiments. Optimal values are highlighted in bold. The ▲ symbol represents the proposed hCSA-OSLM algorithm and the best results along with percentage improvements are
represents an improved result whereas symbol represents a deteriorating result. The results in Table 14 show that, in case of BSE SENSEX the proposed model demonstrated a decline in best as well as mean performance compared to the second-best baseline model producing a decreased AM1, i.e., 0.28% and 0.36% less for pre-COVID and 0.46%, 0.49% less for COVID. Aside from that, the proposed model outperforms competing baseline models in terms of both PM1 and AM1 over both the pre-COVID and COVID timeframes in both the best and average performance. The results verify that the proposed approach is robust and reliable algorithm for stock index price forecasting achieving PM1 (MSE) improvement of 4.6–6.7% and 24.4–30.6% than the competing second best model for pre-COVID and COVID timeframe, respectively, in all benchmark datasets. As an exception, the classification performance of the proposed approach is very low. In terms of stock index movement forecasting, it achieves AM1 (accuracy) improvement of only 0.24%, 0.4% improvement for S&P pre-COVID timeframe data and 0.8%, 1.4% improvement for DJIA COVID timeframe data. However, it lags behind in case of BSE SENSEX data for both the timeframes.

In addition, P-SVR, RBFN and basic OSELM have also achieved satisfactory prediction performance, although they are way behind as compared to hCSA-OSELM. Since the estimated PM1 and AM1 in Table 14 only provide predictive performance based on two metrics, Fig. 8 illustrates the values of ten performance measures, for the purpose of graphical comparison and additional information. Pre-COVID and COVID timeframes are layered on top of one other to improve the visualization of results. In terms of prediction metrics, the hCSA-OSELM significantly outperforms other competing models on all benchmark datasets in both timeframes, although the difference in classification is quite marginal, demonstrating that the proposed hCSA-OSELM is a powerful prediction technique with some space for improvement. The ten-performance metrics obtained from all prediction models (seven proposed) for each benchmark datasets in both pre-COVID and COVID timeframes are compared and one proposed (cross-layered) for each benchmark datasets. As an exception, classification is quite marginal, demonstrating that the proposed hCSA-OSELM is a powerful prediction technique with some space for improvement. The ten-performance metrics obtained from all prediction models (seven proposed) for each benchmark datasets in both pre-COVID and COVID timeframes are compared and one proposed (cross-layered) for each benchmark datasets.

### Table 12 Prediction results of OSELM for benchmark stock index data and various training optimization schemes

|                | BSE SENSEX |       | S&P 500 |       | DJIA |       |
|----------------|------------|-------|---------|-------|------|-------|
|                | Pre-COVID  | COVID | Pre-COVID| COVID|
| PSO            | Best       | 1.280E−04 | 5.912E−04 | 1.770E−04 | 9.126E−04 | 2.654E−04 | 1.142E−03 |
| Pradeepkumar and Ravi (2017) | Mean      | 1.285E−04 | 5.915E−04 | 1.774E−04 | 9.130E−04 | 1.774E−04 | 9.130E−04 |
|                |            | 2.412E−07 | 2.350E−07 | 2.47E−07  | 2.492E−07 | 2.465E−07 | 2.492E−07 |
| DE             | Best       | 1.309E−04 | 6.098E−04 | 1.781E−04 | 8.455E−04 | 2.665E−04 | 1.242E−03 |
| Abdual-Salam et al. (2010) | Mean      | 1.313E−04 | 6.102E−04 | 1.786E−04 | 8.460E−04 | 1.786E−04 | 8.460E−04 |
|                |            | 2.311E−07 | 2.256E−07 | 2.52E−07  | 2.351E−07 | 2.525E−07 | 2.351E−07 |
| CSA            | Best       | 1.281E−04 | 5.772E−04 | 1.761E−04 | 8.273E−04 | 2.591E−04 | 1.402E−03 |
| Dash et al. (2021) | Mean      | 1.286E−04 | 5.776E−04 | 1.765E−04 | 8.277E−04 | 1.765E−04 | 8.277E−04 |
|                |            | 2.088E−07 | 2.183E−07 | 2.18E−07  | 1.959E−07 | 2.182E−07 | 1.959E−07 |
| GWO            | Best       | 1.262E−04 | 5.330E−04 | 1.790E−04 | 8.908E−04 | 2.621E−04 | 1.382E−03 |
| Liu et al. (2021) | Mean      | 1.267E−04 | 5.334E−04 | 1.794E−04 | 8.912E−04 | 1.794E−04 | 8.912E−04 |
|                |            | 2.415E−07 | 2.136E−07 | 1.84E−07  | 2.510E−07 | 1.836E−07 | 2.510E−07 |
| HS             | Best       | 1.288E−04 | 5.382E−04 | 1.758E−04 | 8.847E−04 | 2.663E−04 | 1.179E−03 |
| Dash et al. (2014) | Mean      | 1.292E−04 | 5.386E−04 | 1.762E−04 | 8.850E−04 | 1.762E−04 | 8.850E−04 |
|                |            | 2.119E−07 | 2.019E−07 | 2.48E−07  | 2.249E−07 | 2.476E−07 | 2.249E−07 |
| TLBO           | Best       | 1.268E−04 | 5.523E−04 | 1.716E−04 | 8.052E−04 | 2.580E−04 | 1.118E−03 |
| Das and Padhy (2018) | Mean      | 1.272E−04 | 5.526E−04 | 1.721E−04 | 8.056E−04 | 1.721E−04 | 8.056E−04 |
|                |            | 2.374E−07 | 2.227E−07 | 1.75E−07  | 2.362E−07 | 1.751E−07 | 2.362E−07 |
| hCSA           | Best       | 1.258E−04 | 4.666E−04 | 1.706E−04 | 7.497E−04 | 2.394E−04 | 9.923E−04 |
|                | Mean       | 1.262E−04 | 4.670E−04 | 1.710E−04 | 7.501E−04 | 1.710E−04 | 7.501E−04 |
|                |            | 1.819E−07 | 2.239E−07 | 2.24E−07  | 2.064E−07 | 2.239E−07 | 2.064E−07 |
| BestImprv     | 0.29%      | 12.47%   | 0.51%    | 6.88%    | 7.23%    | 11.23%   |
| MeanImprv     | 0.39%      | 12.44%   | 0.62%    | 6.89%    | 7.21%    | 11.23%   |
superiority of the hCSA-OSELM. The ranking scores are shown in Fig. 9 where a lower rank number indicates a better model and a higher rank number indicates a worse model. The rank-score in Fig.9 reveals that proposed model (\( m \)) outranks other baselines models in all three benchmark datasets.

6.7 Statistical analysis

Using MSE as the evaluation criterion, a statistical test is carried out to further demonstrate the superiority of hCSA-OSELM for stock index trend analysis. The proposed model’s results are compared to the top three best baseline models, which are ELM, OSELM, RBFN, and P-SVR, obtained from Table 14 and Fig. 8. To identify substantial differences between the hCSA-OSELM and the baselines, the paired \( t \) test is used and the significance level is set to 0.05. The null hypothesis \( (H_0) \) considered here is, \( H_0: \mu_{\text{diff}} < 0 \) where,

\[
\mu_{\text{diff}} = SE_{\text{proposed}} - SE_{\text{baseline}} \quad \{SE = \text{squared error}\}
\]

and subsequently, the alternate hypothesis \( (H_1) \) is \( \mu_{\text{diff}} \geq 0 \). The results obtained from the statistical test along with the calculated p-values show that there is a significant difference between hCSA-OSELM and its competitors in terms of MSE. Therefore, \( H_0 \) has been rejected in all three datasets and \( H_1 \) has been accepted represented as \( \checkmark \) in Table 15. The paired \( t \) test is used again to illustrate the efficacy of the proposed hCSA optimization strategy. Table 16 displays the statistical test results, which show a significant difference between hCSA and other benchmark optimization methodologies. Ultimately, the proposed hCSA method outperformed prior optimization algorithms in OSELM training, allowing hCSA-OSELM to outperform the best state-of-the-art stock index trend prediction algorithms.

6.8 Parameter sensitivity analysis

a. Activation function type: As found from the gaps in literature and indicated in Sect. 4.4, activation functions are critical in the creation of OSELM and have a direct impact on the model’s performance. As a result, in this experiment, we examined the impact of six alternative activation functions, as shown in Table 9, on the prediction model.

b. Hidden node count: According to the OSELM literature, the alteration of hidden nodes has an immediate effect on the complexity, training time and output of the network. As a result, we examined the influence of our model with varied numbers of hidden layer neurons varying by 5, 10, 15, 20, and 25 on three datasets.

c. Type of mutation scheme: The research on DE offers a variety of mutation strategies. In this experiment, we looked at five commonly used mutation schemes and four modified variants of the DE algorithm (Das et al. 2016). In all three benchmark datasets, the best mutation strategy was found to be DE/best/1 after 20 independent runs.

d. Type of chaos map: As a wide variety of chaotic maps can be found in literature, in this experiment, we’ve experimented over 10 popular chaos maps such as Chebyshev, Circle, Gauss/mouse, Iterative, Logistic, Piecewise, Sine, Singer, Sinusoidal, Tent (Kohli and Arora 2018). Logistic chaos map was found to be the most effective among all other chaos maps after 20 independent runs.
7 Discussion of outcomes

The aforementioned experimental results reveal that proposed hCSA is able to improve the performance of OSELM-based stock index trend predictor. With its sequential batch wise learning, OSELM supersedes the single-phase learning-based ELM. One significant disadvantage of OSELM is the selection of activation functions, which is addressed using MCDM-based activation function selection and from Tables 8 and 9, ELU is found to be the most suitable activation function. On the other hand, random weight initialization, another shortcoming of OSELM, can be avoided by the suggested hCSA optimization approach to improve the network’s training performance. The proposed hCSA is found to be a robust optimization technique that is unaffected by initial conditions of random crows and resisted local minima. By applying specific additional steps in the algorithm, the hCSA addresses the shortcomings of traditional CSA in terms of faster convergence, better global solution discovery, and superior local exploitation. The addition of phases such as the mutation scheme, as well as chaotic crows and catfish behavior of crows, makes hCSA a superior algorithm in comparison to the earlier benchmark optimization algorithms. Table 7 provides a comparative analysis of hCSA and various optimization methods based on 8 benchmark functions, with hCSA showing superior performance. As depicted in Table 12 and Fig. 7, proposed hCSA-OSELM obtained superior results and faster convergence than benchmark optimization algorithms for stock index price prediction. If the number of crows and the optimization iterations is increased, there may still be room for improvement.

The proposed hCSA-OSELM model produced excellent results during the pre-COVID timeframe, when the market trend was reasonable and less erratic. Furthermore, the suggested model sustained its superior performance for COVID timeframe data, when the market trend was significantly more erratic than before COVID. In terms of these two timeframes, the suggested model outperformed baseline algorithms namely SVR, RBFN, Elman, MLP, LR, and others, as shown in Table 14 and Fig. 8.

In the case of BSE SENSEX pre-COVID data samples, on average, the proposed model (●) is able to achieve 9.35%, 10%, 10%, 27.16%, 6.66%, 17.10%, 34.71% superior MSE as compared to ELM, OSELM, LR, MLP, RBFN, SVR, ELMAN. Furthermore, the mean accuracy improvement of the proposed model is 0%, 0.98%, 1.11%, 0.24%, 0.36%, 1.99%, 0.24% as compared to ELM (Sivalingam et al. 2016), OSELM (Das et al. 2019), LR,
Table 14  Prediction performance of all the models over benchmark stock index datasets during pre-COVID and COVID scenario

### BSE SENSEX: Pre-COVID

| Model | ELM | OSELM | LR | MLP | RBFN | SVR | ELMAN | Improv  |
|-------|-----|-------|----|-----|------|-----|-------|---------|
| PM1  | Best | 0.000126 | 0.000139 | 0.000114 | 0.000114 | 0.000173 | 0.000135 | 0.000151 | 0.000192 | 6.67% |
| Mean  | 2.68E-07 | 2.36E-07 | 2.04E-07 | 2.08E-07 | 2.28E-07 | 1.75E-07 | 2.27E-07 | 2.06E-07 | -       |
| AM1  | Best | 0.821759 | 0.821759 | 0.8125 | 0.824074 | 0.824074 | 0.805556 | 0.824074 | 0.28%   |
| Mean  | 0.819  | 0.819  | 0.811  | 0.81  | 0.821  | 0.822  | 0.803  | 0.821  | 0.36%  |

### BSE SENSEX: COVID

| Model | ELM | OSELM | LR | MLP | RBFN | SVR | ELMAN | Improv  |
|-------|-----|-------|----|-----|------|-----|-------|---------|
| PM1  | Best | 0.000467 | 0.000751 | 0.000773 | 0.002011 | 0.001579 | 0.000777 | 0.000673 | 0.000924 | 30.60% |
| Mean  | 6.89E-07 | 8.80E-07 | 8.60E-07 | 7.69E-07 | 8.47E-07 | 7.96E-07 | 9.83E-07 | 7.74E-07 | -       |
| AM1  | Best | 0.808999 | 0.812734 | 0.812734 | 0.797753 | 0.805243 | 0.797753 | 0.808999 | 0.46%   |
| Mean  | 0.806  | 0.81   | 0.81   | 0.81 | 0.795   | 0.803   | 0.795   | 0.806   | 0.49%  |

### S&P 500: Pre-COVID

| Model | ELM | OSELM | LR | MLP | RBFN | SVR | ELMAN | Improv  |
|-------|-----|-------|----|-----|------|-----|-------|---------|
| PM1  | Best | 0.000163 | 0.0002 | 0.00017 | 0.000213 | 0.000653 | 0.00019 | 0.000172 | 0.000204 | 4.68% |
| Mean  | 3.19E-07 | 2.20E-07 | 2.56E-07 | 2.83E-07 | 3.28E-07 | 2.74E-07 | 3.04E-07 | 2.86E-07 | -       |
| AM1  | Best | 0.839367 | 0.830317 | 0.832579 | 0.809095 | 0.819005 | 0.80543 | 0.803167 | 0.798643 | 0.81% |
| Mean  | 0.837  | 0.828  | 0.83   | 0.799 | 0.816   | 0.803   | 0.81   | 0.797   | 0.84%  |

### S&P 500: COVID

| Model | ELM | OSELM | LR | MLP | RBFN | SVR | ELMAN | Improv  |
|-------|-----|-------|----|-----|------|-----|-------|---------|
| PM1  | Best | 0.00075 | 0.001189 | 0.0001029 | 0.003877 | 0.001943 | 0.00106 | 0.000996 | 0.002425 | 24.69% |
| Mean  | 1.06E-06 | 1.34E-06 | 1.43E-06 | 1.21E-06 | 1.19E-06 | 1.51E-06 | 1.34E-06 | 1.33E-06 | -       |
| AM1  | Best | 0.902622 | 0.857678 | 0.898876 | 0.82397 | 0.797753 | 0.861423 | 0.861423 | 0.891386 | 0.41% |
| Mean  | 0.9    | 0.855  | 0.86   | 0.822 | 0.796   | 0.859   | 0.859   | 0.888   | 0.44%  |

### DJIA: Pre-COVID

| Model | ELM | OSELM | LR | MLP | RBFN | SVR | ELMAN | Improv  |
|-------|-----|-------|----|-----|------|-----|-------|---------|
| PM1  | Best | 0.000252 | 0.000265 | 0.000282 | 0.000273 | 0.000334 | 0.000268 | 0.000269 | 0.000385 | 4.91% |
| Mean  | 4.48E-07 | 4.40E-07 | 3.99E-07 | 4.51E-07 | 4.58E-07 | 3.65E-07 | 4.77E-07 | 4.47E-07 | -       |
| AM1  | Best | 0.825792 | 0.812217 | 0.791855 | 0.794118 | 0.780543 | 0.823529 | 0.80543 | 0.803167 | 0.27% |
| Mean  | 0.823  | 0.81   | 0.79   | 0.791 | 0.778   | 0.821   | 0.803   | 0.801   | 0.24%  |

### DJIA: COVID

| Model | ELM | OSELM | LR | MLP | RBFN | SVR | ELMAN | Improv  |
|-------|-----|-------|----|-----|------|-----|-------|---------|
| PM1  | Best | 0.000992 | 0.001433 | 0.001591 | 0.005946 | 0.002938 | 0.001426 | 0.001313 | 0.001695 | 24.45% |
| Mean  | 1.79E-06 | 1.76E-06 | 1.82E-06 | 2.00E-06 | 1.46E-06 | 1.82E-06 | 1.98E-06 | 1.57E-06 | -       |
| AM1  | Best | 0.794776 | 0.761194 | 0.779851 | 0.75 | 0.735075 | 0.783582 | 0.772388 | 0.779851 | 1.42% |
| Mean  | 0.792  | 0.758  | 0.778  | 0.748 | 0.733   | 0.782   | 0.77   | 0.777   | 1.27%  |
Fig. 8  Performance comparison of different prediction models over ten evaluation criteria (five prediction-based, five accuracy-based) for three benchmark stock indices.
MLP (Ravi et al. 2017), RBFN (Dash & Dash 2015a), SVR (Vilela et al. 2019), ELMAN (Wang et al. 2021). Similarly, for BSE SENSEX during-COVID data samples, on average, the proposed model is able to achieve 37.84%, 39.53%, 76.73%, 70.37%, 39.84%, 30.66%, 49.40 superior MSE as compared to ELM, OSELM, LR, MLP, RBFN,
SVR, ELMAN. Again, the mean accuracy improvement of the proposed model over baselines is found to be 0.49%, 0.49%, 0.49%, 1.38%, 0.37%, 1.38% as compared to ELM, OSELM, LR, MLP, RBFN, SVR, ELMAN. In the case of S&P 500 pre-COVID data samples, on average, the proposed model is able to achieve 18.40%, 4.65%, 23.36%, 74.88%, 13.68%, 5.20%, 20% superior MSE as compared to ELM, OSELM, LR, MLP, RBFN, SVR, ELMAN. Furthermore, the mean accuracy improvement of the proposed model is 1.08%, 0.84%, 4.75%, 2.57%, 4.23%, 4.49%, 5.019% as compared to ELM, OSELM, LR, MLP, RBFN, SVR, ELMAN. Similarly, for S&P 500 during-COVID data samples, on average, the proposed model is able to achieve 36.91%, 27.13%, 80.61%, 61.33%, 29.25%, 24.72%, 69.01% superior MSE as compared to ELM, OSELM, LR, MLP, RBFN, SVR, ELMAN. Again, the mean accuracy improvement of the proposed model over baselines is found to be 5.26%, 0.44%, 9.48%, 13.06%, 4.77%, 4.77%, 1.35% as compared to ELM, OSELM, LR, MLP, RBFN, SVR, ELMAN. In the case of DJIA pre-COVID data samples, on average, the proposed model is able to achieve 5.26%, 10.63%, 8.02%, 24.77%, 5.97%, 6.66%, 34.54% superior MSE as compared to ELM, OSELM, LR, MLP, RBFN, SVR, ELMAN. Furthermore, the mean accuracy improvement of the proposed model is 1.60%, 4.17%, 4.04%, 5.78%, 0.24%, 2.49%, 2.74% as compared to ELM, OSELM, LR, MLP, RBFN, SVR, and ELMAN. Similarly, for DJIA during-COVID data samples, on average, the proposed model is able to achieve 30.64%, 37.51%, 83.26%, 66.13%, 30.30%, 24.31%, 41.34% superior MSE as compared to ELM, OSELM, LR, MLP, RBFN, SVR, ELMAN. Again, the mean accuracy improvement of the proposed model over baselines is found to be 4.48%, 1.79%, 5.88%, 8.05%, 1.27%, 2.85%, 1.930501931% as compared to ELM, OSELM, LR, MLP, RBFN, SVR, ELMAN. The hCSA-ONSELM can capture the non-linearity in stock index time-series data more effectively and quickly. The statistical analysis presented in Tables 15 and 16 shows results in favor of the suggested hCSA optimization as well as hCSA-ONSELM-based stock index trend analysis.

Despite the fact that the suggested hCSA-ONSELM-based stock index trend predictor outperforms existing baselines in terms of prediction, the improvements in its classification performance are limited, which we will address in the future. In conclusion, the performance of hCSA is found to be superior to that of standard optimization techniques such as CSA, DE, TLBO, GWO, HS, etc. In addition, we discovered favorable outcomes for stock index trend analysis over three benchmark datasets throughout the pre-COVID and COVID timeframes by applying hCSA for OSELM weight optimization.

8 Conclusion
Predicting the price of a stock index is advantageous to investors and brokerage firms. In addition to price prediction, if a model can anticipate whether the trend will be upwards or downwards, it will be an added benefit. However, due to the non-linearity and erratic nature of stock index price movements, efficient forecasting of stock index trends with better reliability and robustness is difficult. In this study, an improved OSELM-based stock index trend predictor is proposed by employing six different activation functions such as Sigmoid, ISRU, ReLU, PreLU, LReLU, and ELU and assessing their aggregated performance based on 10 performance evaluation metrics, five of which belong to prediction and the other five to classification. The appropriate activation function selection problem is represented as an MCDM problem, with the primary goal of selecting the most suitable activation function among these six for the OSELM. Furthermore, OSELM was trained for stock index trend prediction employing the hCSA optimization technique rather than arbitrary weight assignment. Finally, the hCSA-ONSELM prediction results for the three benchmark datasets are tested and differentiated from those of OSELM trained using six benchmark optimization algorithms and eight baseline ML models. In most test cases, the hCSA-ONSELM model outperforms the SVR with three different kernels, RBFN, MLP, LR, and Elman models, in terms of MSE, MAPE, MAE, \( R^2 \), RMSLE, accuracy, precision, recall, f-measures, and g-mean. Furthermore, hCSA outperforms other benchmark optimization techniques with faster convergence in terms of exploration and exploitation of solutions. Hence, the suggested approach may significantly enhance the reliability and preciseness of stock index trend predictions.

While the hCSA-ONSELM generates more appealing outcomes for stock index price prediction, there is still significant space to improve the ML approach’s performance in terms of stock index price movement classification. Although we explored few of the most optimal user-defined parameters of hCSA, such as chaos maps and mutation schemes, it is imperative to analyze it systematically in the future. The hCSA can be extended for multi-objective financial time-series prediction using other ML models as well as OSELM.

Author contributions The first author designed the model, interpreted the data, conducted experiments, and analysis of the results. The second author explored the research area and was a major contributor in writing the manuscript. The author(s) read and approved the final manuscript.

Funding This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.
Data availability The datasets analyzed and experimented during the current study are available at https://www.wsj.com/, https://finance.yahoo.com/, openly.

Declarations

Ethics approval and consent to participate Both authors gave their approval and consent to participate.

Consent for publication Both authors gave their consent for publication.

References

Abdual-Salam ME, Abdul-Kader HM, Abdel-Wahed WF (2010) Comparative study between differential evolution and particle swarm optimization algorithms in training of feed-forward neural network for stock price prediction. In: 2010 the 7th international conference on informatics and systems (INFOS), pp 1–8

Amaral TM, Costa APC (2014) Improving decision-making and management of hospital resources: an application of the PROMETHEE II method in an emergency department. Oper Res Health Care 3(1):1–6

Askarzadeh A (2016) A novel metaheuristic method for solving constrained engineering optimization problems: crow search algorithm. Comput Struct 169:1–12

Bas E, Egrioglu E, Kolemen E (2022) Training simple recurrent deep artificial neural network for forecasting using particle swarm optimization. Granular Comput 7(2):411–420

BBC News (2020) BBC News, coronavirus: FTSE 100, Dow, S&P 500 in worst day since 1987

Brais J-P (1982) L’ingénierie de la décision: l’élaboration d’instruments d’aide à la décision. Université Laval, Faculté des sciences de l’administration

Cavalcante RC, Brasileiro RC, Souza VLF, Nobrega JP, Oliveira ALI (2016) Computational intelligence and financial markets: a survey and future directions. Expert Syst Appl 55:194–211

Chuang L-Y, Tsai S-W, Yang C-H (2008) Catfish particle swarm optimization. IEEE Swarm Intell Symp 2008:1–5

Chuang L-Y, Tsai S-W, Yang C-H (2011) Chaotic catfish particle swarm optimization for solving global numerical optimization problems. Appl Math Comput 217(16):6900–6916

Dana AN (2016) Modelling and estimation of volatility using ARCH/GARCH models in Jordan’s stock market. Asian J Fin Account 8(1):152–167

Das S, Mullick SS, Suganthan PN (2016) Recent advances in differential evolution—an updated survey. Swarm Evol Comput 27:1–30

Das SP, Padhy S (2018) A novel hybrid model using teaching–learning-based optimization and a support vector machine for commodity futures index forecasting. Int J Mach Learn Cybern 9(1):97–111

Das SR, Mishra D, Rout M (2019) Stock market prediction using Firefly algorithm with evolutionary framework optimized feature reduction for OSELM method. Expert Syst Appl X 4:100016

Das SR, Mishra D, Rout M (2020) A hybridized ELM-Jaya forecasting model for currency exchange prediction. J King Saud Univ Comp Inf Sci 32(3):345–366

Das S, Sahu TP, Janghel RR, Sahu BK (2021) Effective forecasting of stock market price by using extreme learning machine optimized by PSO-based group oriented crow search algorithm. Neural Comput Appl 34(1):555–591

Das S, Suganthan PN (2011) Differential evolution: a survey of the state-of-the-art. IEEE Trans Evol Comput 15(1):4–31. https://doi.org/10.1109/TEVC.2010.2059031

Dash R, Dash PK (2015a) A comparative study of radial basis function network with different basis functions for stock trend prediction. In: Power, communication and information technology conference (PCITC). IEEE, pp 430–435

Dash R, Dash PK (2015b) Stock price index movement classification using a CEFLANN with extreme learning machine. In: Power, communication and information technology conference (PCITC). IEEE, pp 22–28

Dash R, Dash PK, Bisoi R (2014) A self-adaptive differential harmony search based optimized extreme learning machine for financial time series prediction. Swarm Evol Comput 19:25–42

Dash R, Routray A, Dash R, Routray R (2021) Designing an efficient predictor model using PSNN and crow search based optimization technique for gold price prediction. Intell Decision Technol 15(2):281–289

Dash R, Samal S, Dash R, Routray R (2019) An integrated TOPSIS crow search based classifier ensemble: in application to stock index price movement prediction. Appl Soft Comp J. https://doi.org/10.1016/j.asoc.2019.105784

Dash R, Samal S, Routray R, Dash R (2018) A TOPSIS approach of ranking classifiers for stock index price movement prediction. In: Advances in intelligent systems and computing, vol 758. https://doi.org/10.1007/978-981-13-0514-6_63

Ecer F, Ardabilii S, Band SS, Mosavi A (2020) Training multilayer perceptron with genetic algorithms and particle swarm optimization for modeling stock price index prediction. Entropy 22(11):1239

Fan M-H, Chen M-Y, Liao E-C (2021) A deep learning approach for financial market prediction: utilization of Google trends and keywords. Granular Comput 6(1):207–216

Fan S, Zhang J, Blanco-Davis E, Yang Z, Yan X (2020) Maritime accident prevention strategy formulation from a human factor perspective using Bayesian Networks and TOPSIS. Ocean Eng 210:107544

Fernández-Navarro F, Hervás-Martínez C, Ruiz R, Riquelme JC (2012) Evolutionary generalized radial basis function neural networks for improving prediction accuracy in gene classification using feature selection. Appl Soft Comput 12(6):1787–1800

Gandomi AH, Yang X-S (2014) Chaotic bat algorithm. J Comput Sci 5(2):224–232

Guleria A, Bajaj RK (2020) A robust decision making approach for hydrogen power plant site selection utilizing (R, S)-norm Pythagorean fuzzy information measures based on VIKOR and TOPSIS method. Int J Hydrogen Energy 45(38):18802–18816. https://doi.org/10.1016/j.ijhydene.2020.05.091

Guo W, Xu T, Tang K, Yu J, Chen S (2018a) Online sequential extreme learning machine with generalized regularization and adaptive forgetting factor for time-varying system prediction. Math Problems Eng 2018:1–22

Guo Y, Han S, Shen C, Li Y, Yin X, Bai Y (2018b) An adaptive SVR for high-frequency stock price forecasting. IEEE Access 6:11397–11404

Hasnain S, Ali MK, Akhter J, Ahmed B, Abbas N (2020) Selection of an industrial boiler for a soda-ash production plant using analytical hierarchy process and TOPSIS approaches. Case Stud Therm Eng 19:100636

He D, He C, Jiang L-G, Zhu H, Hu G (2001) Chaotic characteristics of a one-dimensional iterative map with infinite collapses. IEEE Trans Circuits Syst I Fundam Theory Appl 48(7):900–906

Springer
Hezer S, Gelmez E, Özceylan E (2021) Comparative analysis of TOPSIS, VIKOR and COPRAS methods for the COVID-19 Regional Safety Assessment. J Infect Public Health 14(6):775–786

Huang G-B, Liang N-Y, Rong H-J, Saratchandran P, Sundararajan N (2005) On-line sequential extreme learning machine. Comput Intell 2005:223–237

Huang G-B, Zhou H, Ding X, Zhang R (2011) Extreme learning machine for regression and multiclass classification. IEEE Trans Syst Man Cybern B (cybern) 42(2):513–529

Huang G-B, Zhu Q-Y, Siew C-K (2006) Extreme learning machine: theory and applications. Neurocomputing 70(1–3):489–501

Hussien AG, Amin M, Wang M, Alsadn A, Gumaei A, Chen H (2020) Crow search algorithm: theory, recent advances, and applications. IEEE Access. https://doi.org/10.1109/ACCESS.2020.3024108

Hwang C-L, Yoon K (1981) Methods for multiple attribute decision making. In: Multiple attribute decision making. Springer, pp 58–191

Jiang W (2021) Applications of deep learning in stock market volatility using particle swarm optimization trained quantile regression neural network. Appl Soft Comput 58:35–52. https://doi.org/10.1016/j.asoc.2017.04.014

Pradeepkumar D, Ravi V (2017) Forecasting financial time series volatility using particle swarm optimization trained quantile regression neural network. Appl Soft Comput 58:35–52. https://doi.org/10.1016/j.asoc.2017.04.014

Ravi V, Pradeepkumar D, Deb K (2017) Financial time series prediction using hybrids of chaos theory, multi-layer perceptron and multi-objective evolutionary algorithms. Swarm Evol Comput 36:136–149

Sahani M, Dash PK, Samal D (2020) A real-time power quality events recognition using variational mode decomposition and online-sequential extreme learning machine. Measurement 157:107597

Samal S, Dash R (2021a) A TOPSIS-ELM framework for stock index price movement prediction. Intell Dec Technol 15:201–220. https://doi.org/10.3233/IDT-200013

Samal S, Dash R (2021b) A TWV classifier ensemble framework. In: Intelligent and cloud computing. Springer, pp 255–262

Sánchez JS, García V, Marqués AI (2012) Assessment of financial risk prediction models with multi-criteria decision making methods. In: International conference on neural information processing, pp 60–67

Shen W, Guo X, Wu C, Wu D (2011) Forecasting stock indices using radial basis function neural networks optimized by artificial fish swarm algorithm. Knowl-Based Syst 24(3):378–385

Sivalingam KC, Mahendran S, Natarajan S (2016) Forecasting gold prices based on extreme learning machine. Int J Comput Commun Control 11(3):372–380

Storn R, Price K, Price K (1997) Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. J Glob Optim 11(4):341–359. https://doi.org/10.1023/A:1008028213278

Tian D, Zhao X, Shi Z (2019) Chaotic particle swarm optimization with sigmoid-based acceleration coefficients for numerical function optimization. Swarm Evol Comput 51:100571

Wang J (2020) Extreme learning machine with applications in fashion retailing. Decis Support Syst 123:103095

Wang J, Athanasopoulos G, Hyndman RJ, Wang S (2018a) Crude oil price forecasting based on internet concern using an extreme learning machine. Expert Syst Appl 159:113438

Wang J, Sun Z-L, Choi T-M, Au K-F, Yu Y (2008) Sales forecasting using LSTM-RNN stock price prediction algorithm with leading indicators. Multimedia Syst. https://doi.org/10.1007/s00530-008-00758-w

Wu JM-T, Li Z, Herencsar N, Vo B, Lin JC-W (2021a) A graph-based CNN-LSTM stock price prediction algorithm with leading indicators. Multimedia Syst. https://doi.org/10.1007/s10462-018-9663-3

Wu JM-T, Sun L, Srivastava G, Lin JC-W (2021b) A novel synergetic lstm-ga stock trading suggestion system in internet of things. Mobile Inf Syst 2021:1–15

Xue J, Zhou S, Liu Q, Liu X, Yin J (2018) Financial time series prediction using 1RF-ELM. Neurocomputing 277:176–186

Yu H, Ming LJ, Sumei R, Shuping Z (2020) A hybrid model for financial time series forecasting—integration of EWT, ARIMA with the improved ABC optimized ELM. IEEE Access 8:84501–84518

Zhang R, Dong ZY, Xu Y, Meng K, Wong KP (2013) Short-term load forecasting of Australian national electricity market by an ensemble model of extreme learning machine. IET Gen Transm Distrib 7(4):391–397

Zheng F, Zhong S (2011) Time series forecasting using an ensemble model incorporating ARIMA and ANN based on combined...
