SSRG International Journal of Electronics and Communication Engineering
ISSN: 2348-8549/ https://doi.org/10.14445/23488549/IJECE-V11I6P108

Volume 11 Issue 6, 88-106, June 2024

Original Article

A Novel Model for Predicting Stock Index Trends through Hybrid Observed Mode Decomposition-Based Optimized Dynamic Sequential Extreme Learning Machine

R. Sumathi[1], S. Ashokkumar[2]

[1,2]*Department of Computer Science and Engineering, Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences (Deemed to be University), Chennai, India.*

[1]*Corresponding Author : r.sumathi.sap@gmail.com*

Received: 15 April 2024 Revised: 25 May 2024 Accepted: 10 June 2024 Published: 29 June 2024

Abstract - *In the dynamic field of financial markets, the precise prediction of Stock Index Trends (SIT) has long been a significant objective for investors and traders. As global markets evolve, conventional models encounter challenges in keeping abreast of the intricacies and rapid transformations inherent in today's financial ecosystems. The innovative SI Trend Prediction (SITP) model introduced in this study integrates a novel method by integrating the Hybrid algorithm with Observed Mode Decomposition (OMD) and Optimized Dynamic Sequential Extreme Learning Machine (ODS-ELM), denoted as OMD-ODS-ELM. The Hybrid IHS algorithm is deployed to optimize the model parameters, thereby enhancing the efficiency and convergence nothe decomposed data is input into ODS-ELM, a Machine Learning (ML) algorithm suitable for online learning scenarios, to predict real-time SITs. This hybrid model capitalizes on the IHS algorithm's enhanced optimization capabilities and leverages the strengths of OMD and ODS-ELM for robust and accurate Stock Market Trend Prediction (SMTP). Overall, it is a valuable tool for investors and financial analysts in decision-making. The proposed model significantly contributes to financial prediction by providing a robust and efficient tool for predicting SITs, facilitating informed decision-making for investors and financial analysts.*

Keywords *- Stock index trend predictor, Improved harmony search, ML algorithm, Observed mode decomposition, Dynamic sequential extreme learning machine, Decision-making processes.*

1. Introduction

Predicting SITs is a challenging yet essential task within financial analysis. Investors and traders actively seek reliable models to assist them in making well-informed decisions regarding the future trajectories of SM indices [1]. This introduction presents a SITP model designed to analyze historical data, discern patterns, and brand projections about future market movements. In SM Prediction (SMP), two conventional methods have been employed. The first is essential analysis, which relies on a corporation's techniques and essential data, such as expenses, annual growth rates, and market position. The second approach is technical analysis, emphasizing past Stock Prices (SPs) and values. This method uses historical patterns and charts to predict prices [2]. In the past, financial experts typically handled SMPs. However, with advancements in learning techniques, data analysts have taken on prediction challenges. Additionally, research scholars are incorporating ML methods to enhance prediction model performance and accuracy [3]. SMP poses numerous challenges, and data analysts often encounter issues when developing predictive models. The SM's instability and the association between investment psychology and market behavior pose significant challenges due to their complexity and non-linearity [4]. Additionally, random factors, for example, public perception of companies or the political situation in countries, can influence SM Trends (SMT). Therefore, effective data preprocessing of stock values and applying appropriate algorithms can facilitate predicting stock values and indices. The application of ML in SMP systems assumes a significant role in facilitating investors and traders in their decision-making procedures. The techniques employed in these systems aspire to identify and acquire knowledge about patterns from an extensive pool of data in an automated manner. The algorithms are inherently self-learning and can address the prediction task of price movements to enhance trading strategies [5]. In recent times, there has been a notable advancement in various methodologies utilized for predicting SMT. One such proposed approach involved the integration of ANN (Artificial Neural Networks), GA (Genetic Algorithms), and HMM (Hidden Markov Model) [6]. The primary objective was to adapt daily stock ethics into distinct clusters of prices, which would then serve as inputs to the HMM.

In another study, the predictability of financial trends was explored using an SVM model, with a specific focus on evaluating the weekly inclination of the NIKKEI 225 index [7]. Through a comparative analysis involving (EBNN) Elman Backpropagation Neural Networks, SVM, the (QD) Quadratic Discriminant method, and the (LD) Linear Discriminant method, it was concluded that SVM exhibited superior performance in terms of classification.

Furthermore, a novel financial prediction system based on an SVM ensemble was introduced, considering diversity analysis and individual predictions when selecting base classifiers [8]. The results demonstrated a significant improvement in classification when using the SVM ensemble compared to individual SVM models [9].

In [10], a study was focused on predicting rate trends of the Hang index from the Hong Kong marketplace, ten data mining methods were employed, including Bayesian classification, Tree-based classification, KNN, NN, and SVM. The study's results revealed that SVM outperformed the other predictive models, showcasing its efficacy in predicting value trends in the Hang index. Another study employed a developed Legendre neural network to forecast value fluctuations by considering investors' positions and decisions by analyzing prior data [11]. Notably, the prediction model incorporated a random function (time strength).

1.1. Problem Statement

The research background underscores the effectiveness of each algorithm in addressing stock prediction challenges. The motivation for developing a SITP model using ML arises from the necessity to leverage advanced analytical techniques for informed decision-making in financial markets. Traditional methods often struggle to capture SI movements' intricate patterns and dynamics, hindering investors' ability to make timely and accurate predictions.

ML presents a promising solution with its capability to analyze vast datasets and discern complex patterns. By training models on historical SM data, incorporating various financial indicators, and adapting to changing market conditions, the aim is to create a robust predictive tool. This model aims to assist investors in identifying potential trends, optimizing investment strategies, and ultimately enhancing overall portfolio performance. The overarching objective is to improve market efficiency, mitigate risks, and provide market participants with valuable insights to navigate the complexities of the financial landscape.

1.2. Research Contribution

The principal contribution of the proposed SITP Model lies in its integration of advanced methodologies, specifically HOMD-ODS-ELM, augmented by the incorporation of Improved Harmony Search (IHS). HOMD functions as a robust feature extraction tool, decomposing SI data into intrinsic mode functions with optimized parameters, facilitating the model's effective capture of complex market patterns. HOMD-ODS-ELM ensures adaptability to changing market dynamics by continually updating its learning parameters with new data. The introduction of IHS optimizes the model's performance, refining the parameter tuning process and enhancing overall predictive accuracy. The combination of these components results in a highly efficient and accurate SITP model, providing investors with a sophisticated model for making informed decisions in the unpredictable and dynamic field of financial markets. This research is structured as Section 2 defines related work on SM trend modelling using ML. Section 3 introduces and deliberates the proposed SITP model. Section 4 of the paper presents the final prediction results and a detailed analysis of the findings. This section likely delivers insights into the dissimilar predictive models' performance and accuracy in predicting value trends for the Hang index. On the other hand, Section 5 summarizes the key findings and potentially discusses the implications and future directions for research in this field.

2. Related Works

This section presents an analysis of current models based on the application of the proposed research. Sangeetha and Alfia [12] evaluated an ML technique based on linear regression for predicting financial SMT. Acknowledging the non-linear and discontinuous nature of factors influencing SMs, the study highlighted the importance of selecting a representative set of global financial data as a fundamental step in any SMP model. The primary focus of the research was to apply the ELR-ML (Evaluated Linear Regression-based ML) technique for forecasting the financial standards of the Standard and Poor's 500 (S and P 500) indexes. The study considered various factors such as open, low, high, volume, and close for predicting the values of S and P 500 data. The ELR-ML technique was an ML approach that used linear regression models to predict financial values and was evaluated based on its effectiveness in predicting the S and P 500 index values. The primary limitation of linear regression based financial SMP was its oversimplified assumption of a linear relationship, constraining its ability to capture the intricate and non-linear dynamics of the market.

Deng S et al. [13] developed a methodology for predicting price trends and simulating trades on the Shenzhen Component and Shanghai Stock Exchange indexes. This involved the integration of Boosting, NSGA-II, and Bagging methods, complemented by utilizing the SHAP method for model interpretation. Experimental findings showcased that the approach surpassed benchmark methods regarding maximum drawdown, hit ratio, and accumulated return. These results highlight the method's prowess in achieving high accuracy, stable profit, and low risk when predicting price trends and simulating trades in Chinese SIs. However, a drawback of Boosting, NSGA-II, and Bagging methods in financial SMP was their vulnerability to overfitting, especially when dealing with noisy or non-stationary market data.

Venkateswararao and Reddy [14] proposed a hybrid ML technique for LT-SMF (long-term SM price trend forecast). They utilized IBO (Improved Butterfly Optimization) to preprocess input data and employed scaling, polarizing, and difference percentages for feature selection. A hybrid FEL-DNN was then employed to predict SM price variations, and the model was evaluated using 11 social media data and SM indices. Performance comparison with state-of-the-art models was conducted across various metrics. A potential drawback of the hybrid FEL-DNN SMTP model was increased complexity, which may pose challenges in interpretability and computational resource requirements.

Jiang et al. [15] proposed a two-stage ensemble model for predicting SPs. The model combined Improved Harmony Search (IHS), Variational Mode Decompositions (VMD) or Empirical Mode Decompositions (EMD), and Extreme Learning Machine (ELM) algorithm. The resulting models, VMD-ELM-HIS and OMD-ELM-IHS, were compared with other methods such as OMD-ELM, Support Vector Regression (SVR), ARIMA (Autoregressive Integrated Moving Average), ELM, VMD-ELM, LSTM and MLP (Multi-Layers Perception) models.

Results indicated good performances in relation to stability and accuracy for these models, with the size of the training set and sliding window significantly impacting predictive performance. However, a potential drawback of the new two-stage ensemble models in SMTP was increased complexity and challenges in parameter tuning and interpretability.

Zhang and Chen [16] introduced an innovative two-stage prediction model. In the initial stage, the VMD decomposition algorithm was applied to the SP time series, and three individual models (SVR, ELM, and DNN) made predictions for the sub-series. In the subsequent stage, a non-linear ensemble strategy based on ELM combined the initial SP predictions.

The performance of the model was compared to fourteen other models using improvement percentage, accuracy evaluation, and statistical tests. However, the drawback of combining VMD in SMTP lies in potential complexity, computational demands, and challenges in tuning multiple parameters, which may hinder interpretability and practical implementation.

Lin et al. [17] presented an ML prediction method developed for each pattern based on trained outcomes. They formulated an investment approach using ensemble ML techniques, incorporating six prediction models, such as RF, LR, GBDT, KNN, LSTM, and SVM, with optimized parameters for each model. The effectiveness of their feature engineering was validated through empirical results (China's SM), achieving a prediction accuracy of over 60% for specific trend patterns.

Big data utilization, feature standardization, and non-standard data elimination effectively addressed data noise. Based on their forecasting outline, the investment approach theoretically excelled in individual stock and portfolio routines. The drawback of ensemble models incorporating six commonly used prediction models in SMTP lies in the potential difficulty of managing increased complexity and the need for optimal parameter tuning across diverse algorithms, leading to computational challenges and reduced interpretability.

Yuan et al. [18] showed a learning approach to evaluate the profitability of integrated stock selection models that employed different feature selection techniques and algorithms for predicting SP trends. They utilized feature selection methods to filter the original features and employed the time-sliding window technique for cross-validation to regulate parameters for SP trend prediction systems. This approach aimed to enhance the model's applicability in real investment transactions.

The empirical findings of their study indicated that the optimal performance was achieved when RF) was used for both SP trend forecasting and feature selection. This suggests that RF was particularly effective in this context, demonstrating its potential for improving the accuracy of SP trend predictions. However, a drawback of RF in SMTP was its susceptibility to overfitting, particularly in noisy or high-dimensional data.

2.1. Research Gap

As analyzed, existing methods in SM trend prediction, such as linear regression, ensemble models, and hybrid approaches, often face challenges like oversimplified assumptions, susceptibility to overfitting, difficulties in parameter tuning, and increased computational complexity. OELM addresses these drawbacks by utilizing a single hidden layer feedforward NN with a random hidden node structure, allowing for efficient training and adaptability to changing market conditions.

The optimization process enhances the model's performance, making it particularly suitable for SMTP where accurate capture of complex and dynamic patterns is crucial, with reduced risk of overfitting and improved computational efficiency.

3. Proposed Methodology

The Hybrid OMD Based ODS-ELM for SITP model represents an advanced and innovative methodology intended to enhance the accuracy and adaptability of SM trend predictions. The model amalgamates two pivotal components:

OMD: OMD, a signal processing technique, dissects SI data into Intrinsic Mode Functions (IMFs) by adaptively segregating various frequency components. The hybrid nature involves optimizing the decomposition parameters, augmenting OMD's capability to capture both short-term fluctuations and long-term trends in SM data.

ODS-ELM: ODS-ELM is an online learning algorithm that continually adjusts its parameters with the influx of new data. Rooted in a single hidden-layer neural network, ELM, and the IHS optimization process entails fine-tuning ODS-ELM parameters to enhance learning efficiency and adaptability to dynamic market conditions. By integrating OMD and ODS-ELM, the model aims to extract meaningful features from SI data via adaptive decomposition and continuously employs an online learning approach to enhance predictive capabilities. This hybridization enables the model to capture SMT's intricate and dynamic nature effectively, providing investors with a robust tool for informed decision-making in the ever-changing financial landscape. The optimization of parameters further fortifies the model's overall performance, rendering it resilient and accurate in predicting SI trends, as illustrated in Figure 1.

3.1. Input Dataset Description

Initially, 13 types of one-day patterns are formulated and categorized from a dataset comprising 3,455 stocks in this study. The experiment aimed to assess the generalization performances of the model utilizing three typical stock index data sets: S and P 500, BSE-SENSEX, and DJIA. The data covered the period (January 2014 to 2021), which included the declarations of the COVID-19 epidemic as an international epidemic. To consider the influence of the pandemic, the raw data was segmented into two distinct parts: data preceding the onset of COVID-19 (pre-pandemic) [19] and data encompassing the epidemic period (during the pandemic) [20]. The objective was to assess whether these models could maintain their performance during unforeseen events like the COVID-19 epidemic. The data sets examined and assessed within this work are obtainable at https://finance.yahoo.com/ and https://www.wsj.com/. In this experiment, six popular technical indicators were utilized: Moving Average Convergence and Divergence (MACD), SMA (Simple Moving Average), William's R percent Relative Strength Index (RSI), and Stochastic K and D., These indicators were used as input parameters for the DSELM framework, along with the previous day's closing prices. To prepare the input, the values of the technical indicators and the prior day's prices on closing were normalized using the min-max normalization technique. This transformation ensured that the values ranged between 0 and 1, making them suitable for the prediction model. The prediction model generated continuous values representing the closing price of the SI.

Furthermore, a closing price based on a moving average was computed using the HOMD-ODS-ELM for both the actual and predicted closing prices. Subsequently, these updated, close prices were employed to determine the shift of the SI through a defined procedure. Overall, this approach incorporated technical indicators as input parameters, normalized the data, and used a hybrid optimization method and extreme learning machine to predict the closing price of the SI. The calculated moving average-based closing prices were further utilized to determine the movement of the SI.

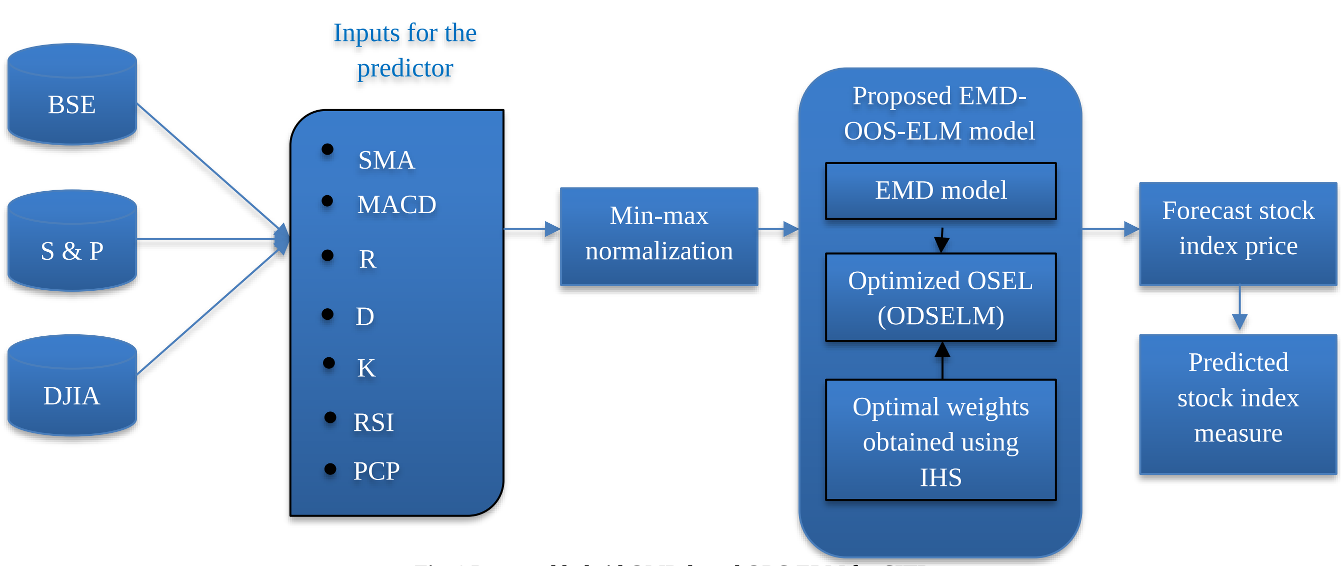

Fig. 1 Proposed hybrid OMD-based ODS-ELM for SITP

3.2. OMD-ODS-ELM for Stock Trend Prediction

Initially, primary predictions $Pred_{n,t}$ of $t > n$ of $Pred_t$ are found using the OMD–ODS-ELM approach under diverse n, $n \in N+$. At another stage, the IHS was utilized to combine numerous nominated $Pred_{n,t}$ by enhancing weights of designated$Pred_{n,t}$, and ending results $Pred_t$ can be obtained. The details of the above two methods are presented.

The pseudocode of Parameter Optimization of DSELM with OMD using the Harmony Search Algorithm for SITP is presented in Table 1. The IHS searches for optimizing the weights of input $\mathbb{w}$ of DSELM in training, which was utilized to test to assess the generalization presentation of the OMD-ODS-ELM for SITP.

Table 1. Pseudocode of parameter optimization of DSELM with OMD using harmony search algorithm for SITP

```
// Load historical SI data
data = load_data()
// Preprocess data (clean, normalize, decompose)
preprocessed_data = preprocess(data)
IMFs=
empirical_mode_decomposition(preprocessed_data)
// Define DSELM parameter ranges
DSELM_params = define_DSELM_parameters()
// Initialize IHS parameters
hsa_params = define_hsa_parameters()
// Initialize harmony memory
harmony_memory=
initialize_harmony_memory(hsa_params)
function fitness(harmony):
   // Train DSELM on each IMF using harmony
parameters
   DSELM_models = train_DSELM_models(IMFs,
harmony)
   // Predict trends for test data
   predicted_trends = predict_trends(DSELM_models,
test_data)
   // Evaluate prediction accuracy using chosen metric
   accuracy = evaluate_accuracy(predicted_trends,
true_trends)
   return accuracy
while not termination_criterion_met:
   // Improvise new harmonies
   new_harmonies = improvise(harmony_memory,
hsa_params)
   // Evaluate fitness of new harmonies
   new_fitness_values=
evaluate_fitness(new_harmonies)
   // Update harmony memory with best harmonies
   update_harmony_memory(harmony_memory,
new_harmonies, new_fitness_values)
// Get optimal DSELM parameters from best harmony
optimal_params=
get_best_harmony(harmony_memory)
// Evaluate model performance with optimal parameters
evaluate_model(optimal_params, test_data)
```

3.2.1. OMD Approach

The OMD approach, as explained in [21], enables the decomposition of a time series within the IMFs set. The procedure involves the following sequential steps:

1. Identify every local extrema of the unique instance data $\mathbb{x}(t)$.
2. Utilize a cubic spline interpolation technique to compute the lower and upper envelopes based on the local minima and maxima, correspondingly.
3. Determine $ag(t)$, which represents the ag value of the two envelopes, and generate a novel time series $nts(t) = \mathbb{x}(t) - ag(t)$.
4. If $h(t)$ contains two situations: (a) ag(t) = 0; (b) the variance among the values of local minima and maxima was lower than 1, then $nts(t)$ was considered as j-th IMF, denoted as IMF$c_j\,(t)$. Repeat the steps by updating $\mathbb{x}(t) = sd(t) - \sum_{j<(j+1)} c_j(t)$. Otherwise, the steps can be reiterated by updating $\mathbb{x}(t) = h(t)$.

Lastly, the $sd(t)$ can be described as:

$$sd(t) = \sum_{j=1}^{j} c_j(t) + \mathbb{x}(t) \qquad (1)$$

Where $\mathbb{x}(t)$ is the residual that cannot be decomposed.

3.3. DSELM Model

Huang et al. [22] presented the ELM as a computational intelligence method. They showcased its advantages over traditional AI techniques, including faster learning speed, reduced computation complexity, and improved global performance when applied to multiclass classification difficulties. Assume a set of N distinct data samples, denoted as (X_i, Y_i), where X_i represents the inputs and Y_i represents the corresponding outputs. A basic neural network with $\mathcal{H}$ hidden neurons and $f(X)$ as the AF (activation functions) could evaluate the N trials with an error closer to 0 if there exist $\mathbb{w}_i$ and $bias_i$ such that:

$$\Rightarrow \sum_{i=1}^{\mathcal{H}} f\left(\mathbb{w}_1 . X_j + bias_i\right)\beta_i = Y_j;\ j = 1,2,3,\dots,N,$$

if $h_{ij} = \mathbb{w}_1 . X_j$, then

$$\sum_{i=1}^{\mathcal{H}} f\left(h_{ij} + b_i\right)\beta_i = Y_j; \qquad (2)$$

In the given Equation, the input weight connecting input neurons is denoted as $\mathbb{w}_i$, the output of the i-th $\mathcal{H}$ given the j-th input is represented as h_{ij}, the bias of the i-th $\mathcal{H}$ is denoted as $bias_i$, and it also represents the output weight connecting the i-th $\mathcal{H}$ to the output neuron. The compact structure of Equation (2) could be expressed as HLβ=Y; here, HL represents the output of the hidden layer. By referring to Equation (3), β is calculated.

$$\hat{\beta} = HL^{\dagger} = Y = (HL^T HL)^{-1} HL^T Y, \qquad (3)$$

DSELM, unlike the single-iteration (BL) Batch Learning approach of ELM, follows sequential or chunk-based training data processing. In contrast to ELM's single BL, DSELM involves two distinct phases: initialization and sequential learning. During the initialization phase of DSELM, a rank $(HL_0) = L$ is required, where HL_0 is the hidden layer output matrix during the initialization phase, and N_0 is the training data sample during initialization with $N_0 \geq \mathcal{H}$.

3.3.1. Initialization Phase Steps

During the initialization phase of the DSELM, a portion of the training information $(X_i, Y_i)_{i=1}^{N_0}$ is fed to the network, where $N_0 \geq \hbar$.

a. Allot initial constraints of the network, such as X_i; $\mathbb{w}_i$, and $bias_i$.

b. Compute H_0 for the initial phase:

$$HL_0 = \sum_{i=1}^{\mathcal{H}} f\left(\mathbb{w}_1 . X_j + bias_i\right); \quad (4)$$

$$i = 1;\ 2, \ldots., \mathcal{H}; j = 1,2,3, \ldots N_0$$

c. Determine the weights of output connections$\beta^{(0)} = (HL_0^T HL_0)^{-1} HL_0^T Y_0 = M_0 HL_0^T Y_0$.

d. Modify the quantity of data chunk $k = 0$.

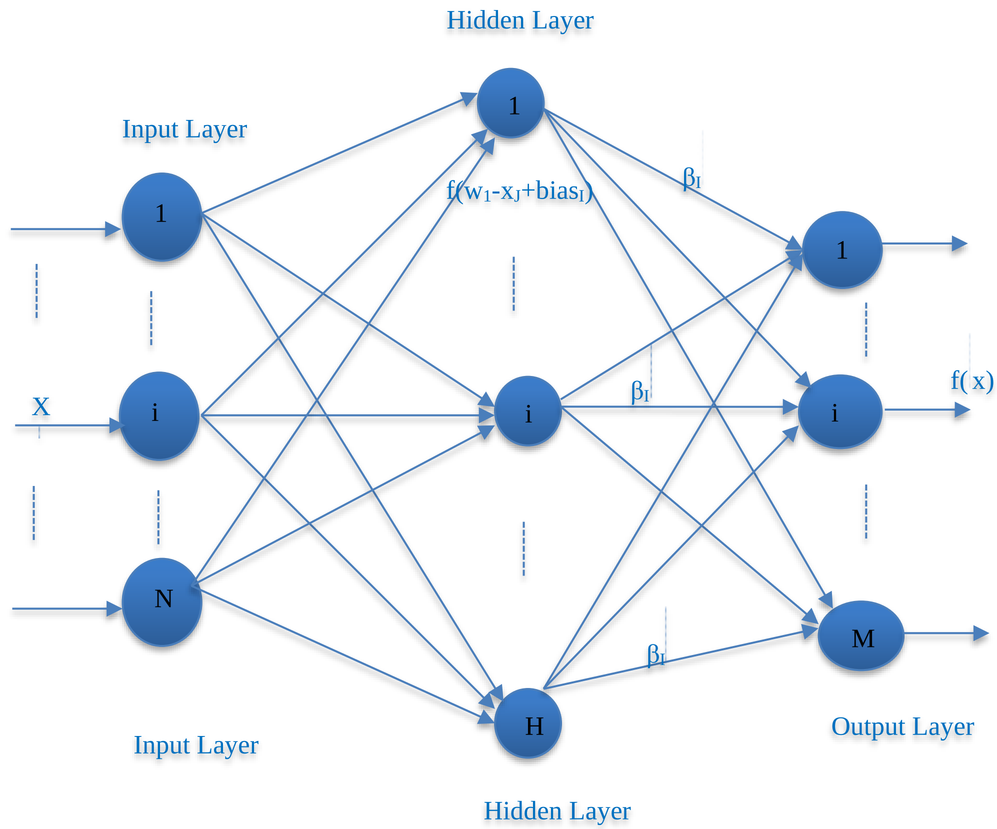

Fig. 2 Architecture diagram of the DSELM model

In the sequential learning phase of the DSELM, the learning process continues with additional chunks of training data, as shown in Figure 2. The steps for this phase are summarized below:

1. Calculate HL_{k+1} for the N_1-th chunk, where N_1 begins from $N_0 + 1$:

$$HL_{k+1} = \sum_{i=1}^{\mathcal{H}} f\left(W_1 . X_{N_0+j} + bias_i\right);\ i = 1,2, \ldots., \mathcal{H} \text{ and } j = 1,2,3, \ldots, N_1 \quad (5)$$

2. Calculate $\beta^{(k+1)}$:

$$\beta^{(k+1)} = \beta^{(k)} + M_{k+1} HL_{k+1} (Y_i^T - HL_{k+1}^T \beta_k) \quad (6)$$

Where,

$$M_{k+1} = M_k - \frac{M_k HL_{k+1} HL_{k+1}^T M_k}{1 + HL_{k=1}^T M_k HL_{k+1}} \quad (7)$$

3. Adjust the quantity of data chunks $k = 1$.

This process is repeated for each subsequent chunk of training data, allowing the model to adapt and learn from new information incrementally.

3.4. Parameter Optimization Using HIS

The HS algorithm, introduced by Geem in 2000, was inspired by the improvisational nature of music. Each inconstant within the result vector is analogized to a musical note in this algorithm.

As described in reference [23], the system emulates adjusting these stock patterns to discover the optimal tune. Let $f(\cdot)$ denote the OF (objective function), and let $X = (x_1, x_2, \dots, x_N)$, where x_i and i = 1, 2, ..., N were the conclusion or enterprise elements associated with accuracy. The fundamental HS algorithm encompasses a few essential parameters, as delineated below.

In Step 1, predefined parameters are established in the algorithm, and the $\mathbb{HM}$ (Harmony Memory) is initialized. The best harmony (X_{best}) and the worst harmony (X_{worst}) for all harmonies are identified.

Moving to Step 2, a novel harmony, X_{new} is generated.

Step 3 involves updating the Harmony Memory ($\mathbb{HM}$) if X_{new} proves to be superior to X_{worst}.

Step 4 entails iterating through Steps 2 and 3 till the specified stop situations are met.

The basic HS algorithm exhibits a relatively slower convergence speed and is susceptible to local minima issues. Therefore, this study introduces an IHS algorithm integrating HS with differential evolution (DE). Two DE parameters, CR (crossover rate) and SF (scaling factor), are incorporated to enhance the algorithm's performance. The structure is outlined in Figure 3, and the primary operations are elucidated as follows.

3.4.1. Initialization Operation

In the algorithm, it is essential to predefine various parameters to ensure its proper functioning. These parameters include:

- Crossover Rate (CR): This parameter determines the probability of crossover occurring during the generation of new solution vectors.
- Scaling Factor (SF): The SF controls the amplification or reduction of the difference between the current solution vector and the harmony memory. It influences the magnitude of the generated trial vectors.
- HMS (Harmony Memory Size): HMS refers to the number of solution vectors stored in the $\mathbb{HM}$. It represents the procedure's memory capacity and affects the diversity of the solutions explored.
- Harmony Memory Considering Rate (HMCR): HMCR signifies the probability of selecting a value from the $\mathbb{HM}$ rather than generating a random value. It influences the algorithm's exploitation-exploration trade-off.
- PAR (Pitch Adjusting Rate): PAR regulates the probability of regulating the selected value from the $\mathbb{HM}$ within a specific range. It introduces randomness and diversity in the solution-generation process.
- Maximum Number of Improvisations (MaxImpr): MaxImpr specifies the maximum number of iterations or improvisations allowed in the procedure. It provides a termination condition for the optimization process.
- Bandwidth Vector (BV): BV represents a vector that defines the range of values for each decision variable in the problem space. It helps in controlling the exploration and exploitation of the search space.

The algorithm can be fine-tuned by predefining these parameters to achieve the desired balance between exploitation and exploration, leading to improved optimization performance. Randomly initialize the $\mathbb{HM}$ as shown in Equation (8).

$$\mathbb{HM} = \begin{bmatrix} x_1^1 & x_2^1 & \cdots & x_N^1 & f(X^1) \\ x_1^2 & x_2^2 & \cdots & x_N^2 & f(X^2) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_1^{\mathbb{HMS}} & x_2^{\mathbb{HMS}} & \cdots & x_N^{\mathbb{HMS}} & f(X^{\mathbb{HMS}}) \end{bmatrix} \quad (8)$$

In this representation, a solution vector $X^j = (x_1^j, x_2^j, \dots, x_N^j)$is defined, where $f(X^j)$ represents the value of the OF at X^j. The variable x_i^j denotes the ith value of the jth solution vector for $j = 1, 2, \dots, \mathbb{HM}S$ based on accuracy prediction.

3.4.2. Fitness Calculation

The fitness (accuracy) of the OF is calculated for each solution vector X^j.

3.4.3. FindingX_{worst}

The solution vector $X_{worst}$is identified based on the calculated fitness values.

3.4.4. Improvisation Operation

The improvisation operation involves the calculation of the BV, according to Keshtegar et al. (2017). The formula for BV is defined as follows:

$$BV_i(Impr) = \frac{x_i^n - x_i^m + 0.001}{10} * exp\left(-10 * \frac{\text{Impr}}{\text{MaxImpr}}\right) \quad (9)$$

The number of improvisations ($Impr$) in the given algorithm is demarcated as the extreme number of iterations [23]. The variables x_i^n and x_i^m represent the lower and upper bounds for the variablex_i. The algorithm follows a specific process described in Table 2 to produce a new harmony. This process involves the use of two functions:

1. $rand(m, n)$: This function generates a random number that varies between m and n, following a normal distribution.
2. $rand_n(m, n)$: This function generates a random number that varies between l and n, following a uniform distribution.

The algorithm calculates the Bandwidth Vector (BV) by utilizing these functions and generates a new harmony. The details of this process can be found in Table 2. Update Operation: The update operation involves comparing (X_{new}) with $f(X_{best})$ and updating X_{best} according to the Equation:

$$Xbest = \begin{cases} X_{new} & f(X_{new}) < f(Xbest), \\ Xbest, & \text{otherwise} \end{cases} \quad (10)$$

3.4.5. Selection Operation

In the given algorithm, a random selection of φ harmonies is performed before the mutation and crossover operations. This step is essential to maintain diversity in the population and avoid premature convergence of the algorithm. It is significant to note that the finest harmony cannot be selected during this random selection process. This is because the best harmony represents the most optimal solution found so far, and selecting it would bias the search towards exploitation and hinder exploration of the search space. By excluding the best harmony from the random selection process, the algorithm can balance exploitation and exploration, improving the optimization routine.

3.4.6. Mutation and Crossover Operations

Two operations, mutation and crossover, are presented in the HS procedure from the DE algorithm. For the mutation, a novel harmony $X_{mutation}$ was formed from the present harmonies according to the formula: In the HS algorithm, two operations, mutation and crossover, are incorporated from the DE algorithm to enhance the search capabilities. The mutation operation involves creating a new harmony, denoted as $X_{mutation}$, based on the current harmonies. This is done using the following formula:

$$X_{mutation} = X^c + SF * (X^a - X^b), \quad (11)$$

The harmoniesX^a, X^b, and X^c are randomly selected in this context. The crossover operation involves crossing $X_{mutation}$ with the equivalent chosen harmony to generate the new harmony, denoted as x_{new}. Additional information can be found in Algorithm 2.

The function $rand_I(m, n)$ is employed to acquire a random integer within the range from m to n. The crossover operation, on the other hand, is not explicitly mentioned in the given information.

3.4.7. Replacement Operation

In the replacement operation, for the selected harmonies, $f\ (X^j_{new})$ is compared with$f\ (X^j)$, and then X^j is replaced according to the Equation:

$$X^j = \begin{cases} X^j_{new}, & f(X^j_{new}) < f(X^j), \\ X^j, & \text{otherwise.} \end{cases} \quad (12)$$

This pseudocode outlines the parameter optimization process for the OMD-DSELM model using the HS algorithm.

1. Initialization: Parameters for the Harmony Search algorithm and OMD-DSELM are initialized. For each parameter i, a decision is made to update it based on HMCR.

Table 2. Pseudo code for IHS.

```
Input: parameter initialization of HS population, parameters of OMD-DSELM
Output: Optimal parameter value selection for OMD-DSELM
for each i ∈ {1,2,...,N} do
   if rand(0,1) ≤ HMCR then
       if rand(0,1) ≤ PAR then
         x_{new,i} ← x_i^{best} + BV * rand_u(−1,1);
           if x_{new,i} > x_i^n, then
     x_{new,i} = x_i^n ;
  end if
Arbitrarily choose φ harmonies (X_{best} cannot be designated);
for every harmony in designated harmonies, do
a = rand_I(0, ℍ𝕄S), b = rand_I(0, HMS) and c = rand_I(0, ℍ𝕄S) (a = b = c);
   X_{mutation} = X^c + SF * (X^a − X^b);
for each i ∈ {1,2,...,N} do
  if x_{mutation,i}, > x_i^n then
    x_{mutation,i} = x_i^n ;
      end if
     if x_{mutation,i} < x_i^m, then
   x_{mutation,i} < x_i^m;
     end if
end for
k = rand_I(1,N);
for each i ∈ {1,2,...,N}
       predicted = rand(0,1);
      if predicted ≤ SF or i = k then
           x_{new,i} = x_{mutation,i} ;
    else
      x_{new,i} = x_i^j ;
     end if
  end for
end for
Return updated X_{new} as optimal value selection for OMD-DSELM
```

2. The update of Harmony Memory: If a randomly generated number is equal to or less than HMCR, a new harmony vector x_{new} is created. If another $random$ ≤PAR, the new value for the parameter is calculated using the best harmony vector X_{best} and a BV. Boundaries are checked, ensuring the new value does not exceed predefined upper and lower bounds.
3. Harmony Selection: A subset of harmonies is randomly selected for mutation and crossover operations, excluding the best harmony vector X_{best}.
4. Mutation Operation: A new harmony vector $X_{mutation}$ is created by combining randomly chosen X^a, X^b, X^c harmonies with an SF.

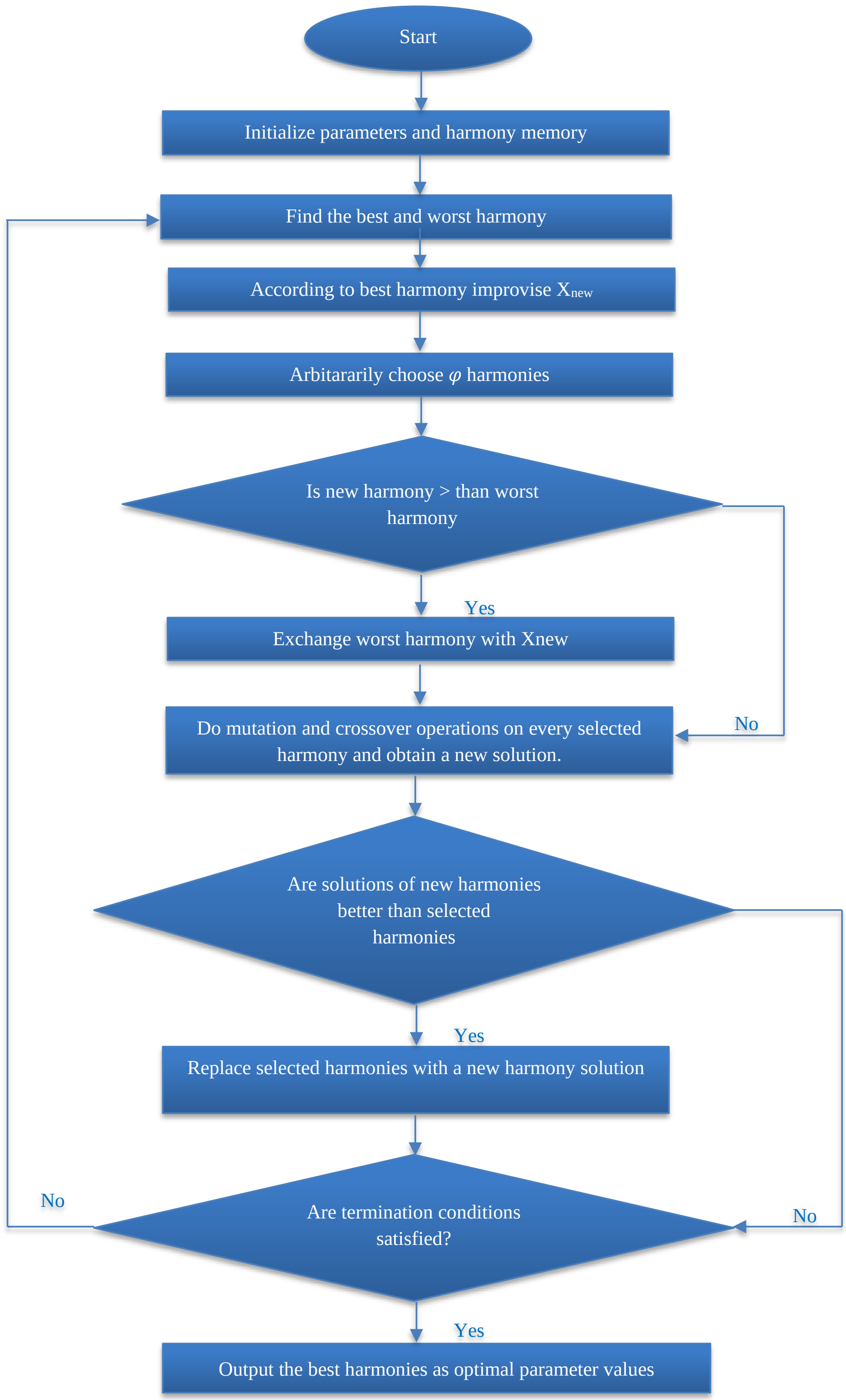

Fig. 3 Flowchart of IHS algorithm for optimal parameter selection of OMD-DSELM

5. Crossover Operation: The mutation vector is combined with a chosen harmony to produce a new harmony, denoted as x_{new}. Random predictions are made to update each element of the harmony vector.
6. Replacement Operation: The new harmony X_{new} is associated with the current harmony X^j for selected harmonies. If the new harmony produces a lower OF value, it replaces the current harmony.
7. Return: The updated X_{new} is returned as the optimal parameter selection for OMD-DSELM.

This pseudocode represents an iteration of the IHS algorithm for optimizing parameters in the OMD-DSELM model. The process is repeated until a stopping criterion is met.

4. Experimental Results and Discussion

The experiments conducted in this study focused on three SI benchmark datasets: the S and P 500 and the DJIA. Section 4.1 of the study provides comprehensive information about these datasets for the BSE SENSEX. The subsequent subsections of the study delve into the empirical validation of the proposed OMD-ODS-ELM. These subsections discuss the baselines used for comparisons and the experimental approaches employed.

The hybrid model, OMD-ODS-ELM, was implemented on a computing system with the following specifications: an Intel® Core™ i7-8750H CPU @ 2.20 GHz, the 64-bit Windows 11 operating system, and 16 GB RAM. The implementation was done using MATLAB R2020a. Figure 4 in the study presents the predicted SI close prices and the actual close prices during both the training and testing phases.

The plots show the proximity between the actual and projected data series for the COVID and pre-COVID timeframes across the training and testing data samples. These findings suggest that the OMD-ODS-ELM hybrid model performs well in predicting SI close prices, displaying accurate results compared to the actual data series.

4.1. Performance Parameters

This segment is dedicated to assessing the presentation of the proposed OMD-ODS-ELM model. In the experiment, a relative analysis is conducted between OMD-ODS-ELM and commonly employed methods in stock trend prediction, namely ELR-ML [12], ELM [16], and SVM [17].

The experimental results affirm that the proposed residual network with multidimensional feature comparison outperforms traditional models. Performance metrics such as precision, sensitivity/recall, specificity, f-measure, and accuracy are employed to assess the performance of the suggested model. True Positives (TPs) denote accurately classified positive stock trends, while False Negatives (FNs) represent negatively classified stock trends. Negative stock trend predictions are labelled as True Negatives (TNs), and their misclassification as positive instances is referred to as False Positives (FPs). The qualitative results of the projected method demonstrate precision in identifying stock trend predictions, showcasing an improved convergence rate of contours compared to previous classification approaches.

Recall is a metric measuring the number of positive class calculations out of all positive instances in the dataset and is defined as follows:

$$Recall = \frac{TP}{TP+FN} \tag{13}$$

Precision, which measures the number of positive class calculations truly belonging to the positive class, is expressed as:

$$Precision = \frac{TP}{TP+FP} \tag{14}$$

F-measure provides a single score balancing both precision and recall, calculated as:

$$F - Measure = \frac{(2 * Precision * Recall)}{(Precision + Recall)} \tag{15}$$

Accuracy, a widely used metric for evaluating classification performance, is defined as the proportion of successfully segmented data relative to the total number of samples:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{16}$$

4.2. Accuracy Performance Analysis

The numerical accuracy results for the proposed and existing methods are presented in Table 3. Figures 5 to 7 illustrate the level of agreement between the suggested OMD-ODS-ELM model and the currently employed models for a given set of samples in a specified database. The OMD-ODS-ELM achieves reduced processing times without compromising accuracy.

Notably, it outperforms all other classification models, including ELR-ML, ELM, and SVM, with accuracies of 99.92%, 99.37%, and 99.98% for BSE, S&P, and DJIA, respectively, without requiring a substantial number of samples during reduction. The IHS Algorithm optimizes the parameters of the DSELM-OMD model, leading to more effective OMD and feature extraction.

This improved feature representation enables the model to capture relevant patterns in SI data better, consequently contributing to higher accuracy in trend prediction. For a given subset of database properties, Figures 8 to 10 show precision results of the proposed OMD-ODS-ELM compared to other models such as ELR-ML, ELM, and SVM in terms of precision. The precision improves with time as measured in epochs.

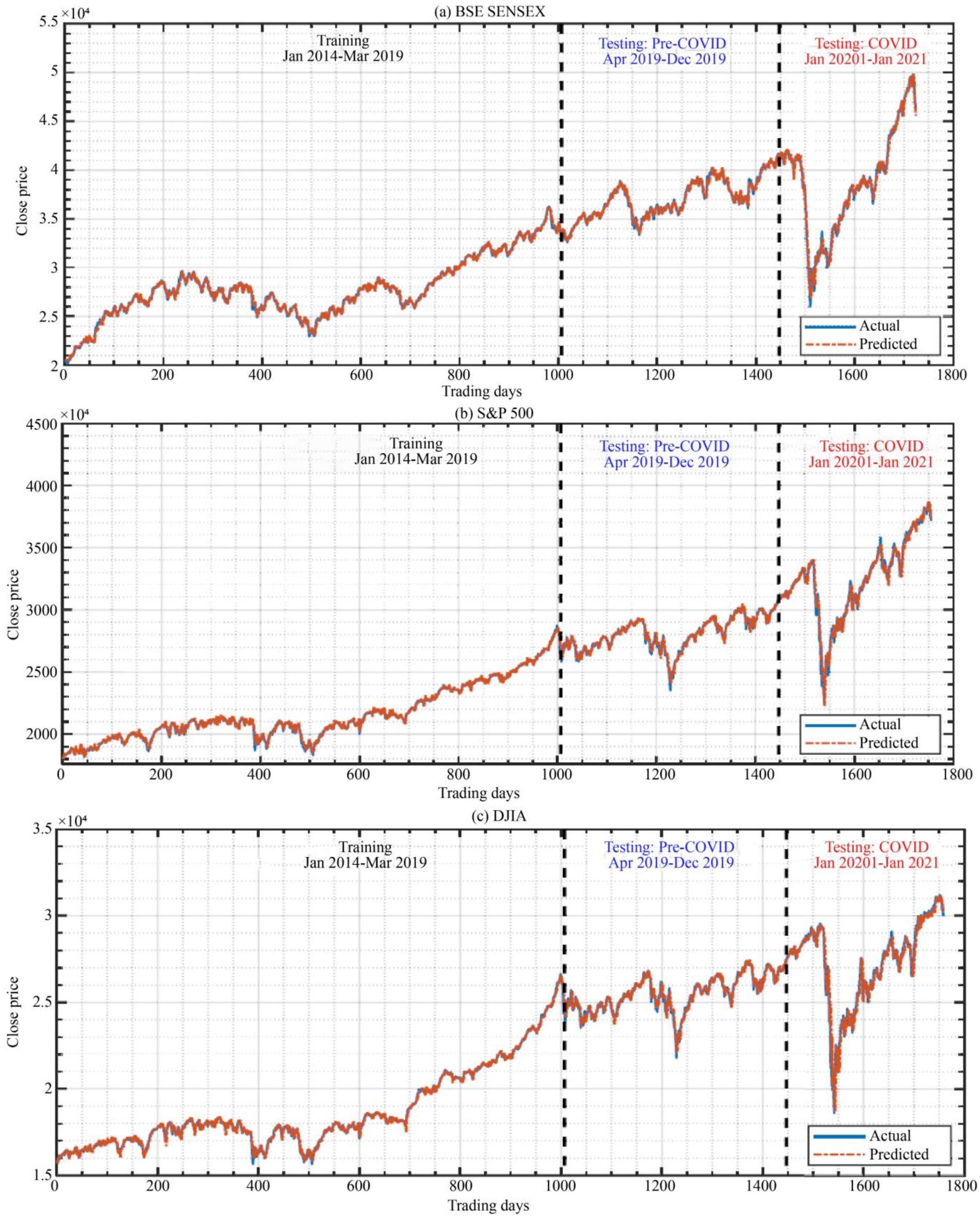

Fig. 4 The prediction values of the proposed OMD-ODS-ELM for three distinct stock indices: a) BSE-SENSEX, b) S&P-500, and c) DJIA

Table 3. The numerical results of accuracy for proposed and existing methods

Number of samples	BSE				S and P				DJIA			
	OMD-ODS-ELM	ELR-ML	ELM	SVM	OMD-ODS-ELM	ELR-ML	ELM	SVM	OMD-ODS-ELM	ELR-ML	ELM	SVM
250	96.47	91.85	90.85	87.85	95.87	91.85	90.15	87.62	96.18	91.45	90.23	87.59
500	98.41	92.35	91.35	88.85	97.81	92.35	90.65	88.62	98.12	91.95	90.73	88.59
750	99.8	92.85	91.85	89.35	99.2	92.85	91.15	89.12	99.51	92.45	91.23	89.09
1000	99.81	93.85	92.35	89.85	99.21	93.85	91.65	89.62	99.82	93.56	92.35	89.95
1250	99.48	94.85	92.85	90.41	98.88	94.85	92.15	90.18	99.49	94.56	92.85	90.51
1500	99.92	96.45	94.05	91.05	99.37	96.45	93.35	90.82	99.98	96.16	94.05	91.15

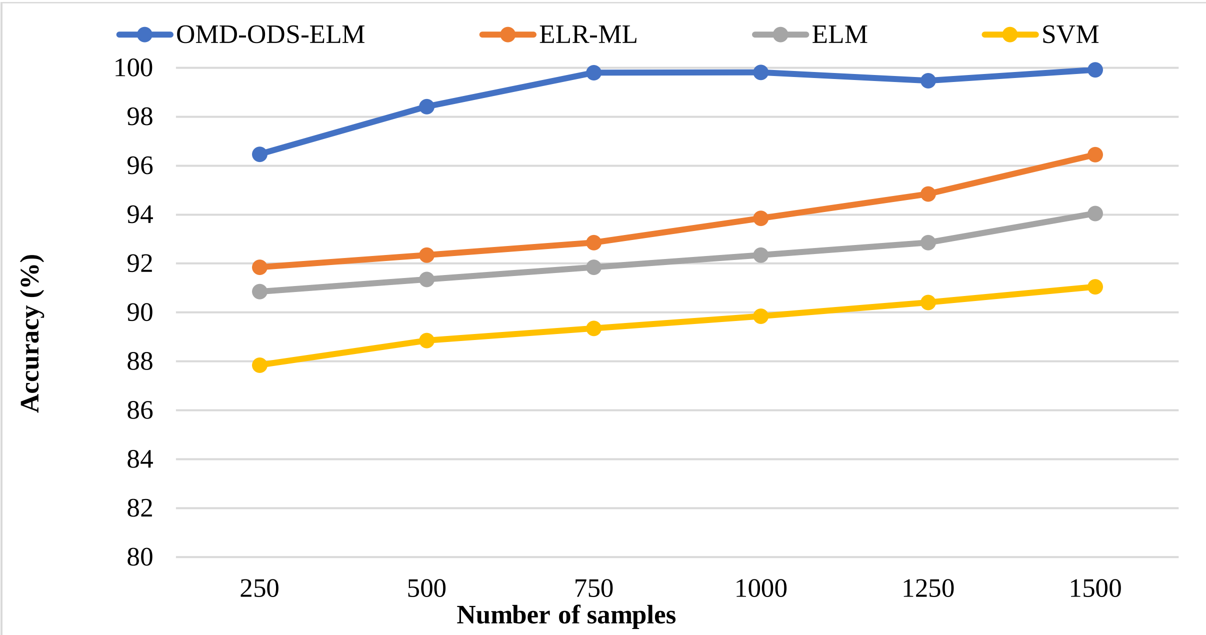

Fig. 5 Accuracy results comparison for BSE

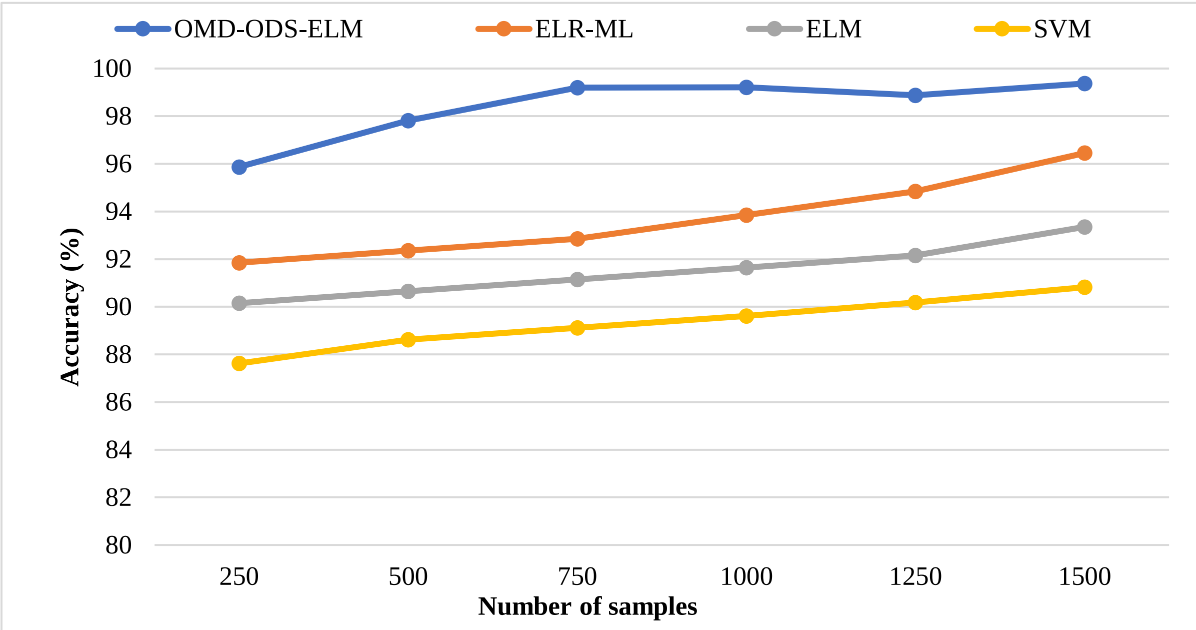

Fig. 6 Accuracy results comparison for S&P

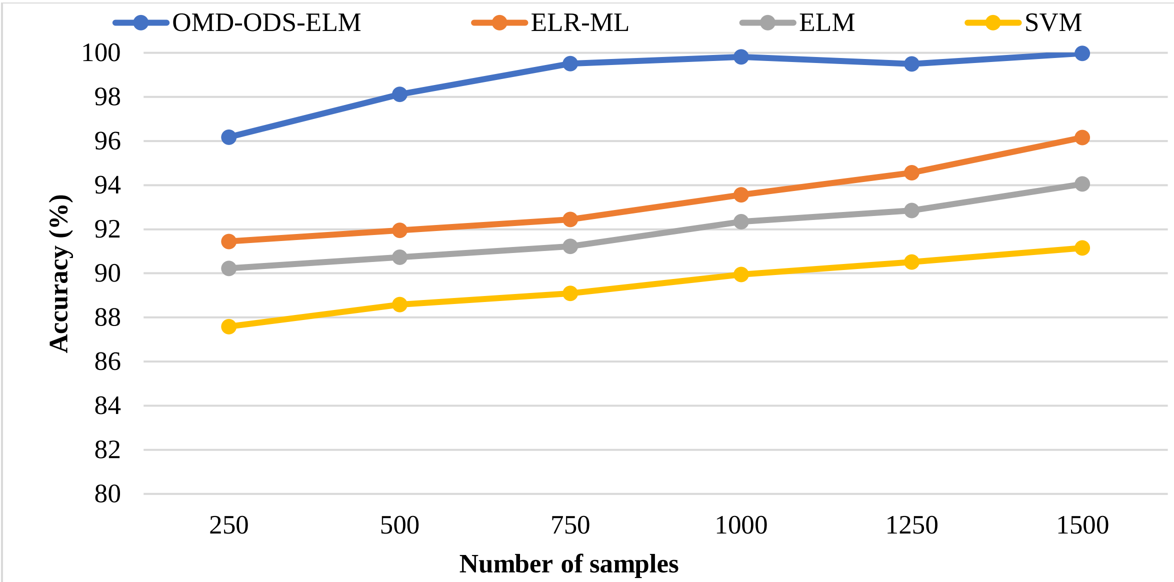

Fig. 7 Accuracy results in comparison for DJIA

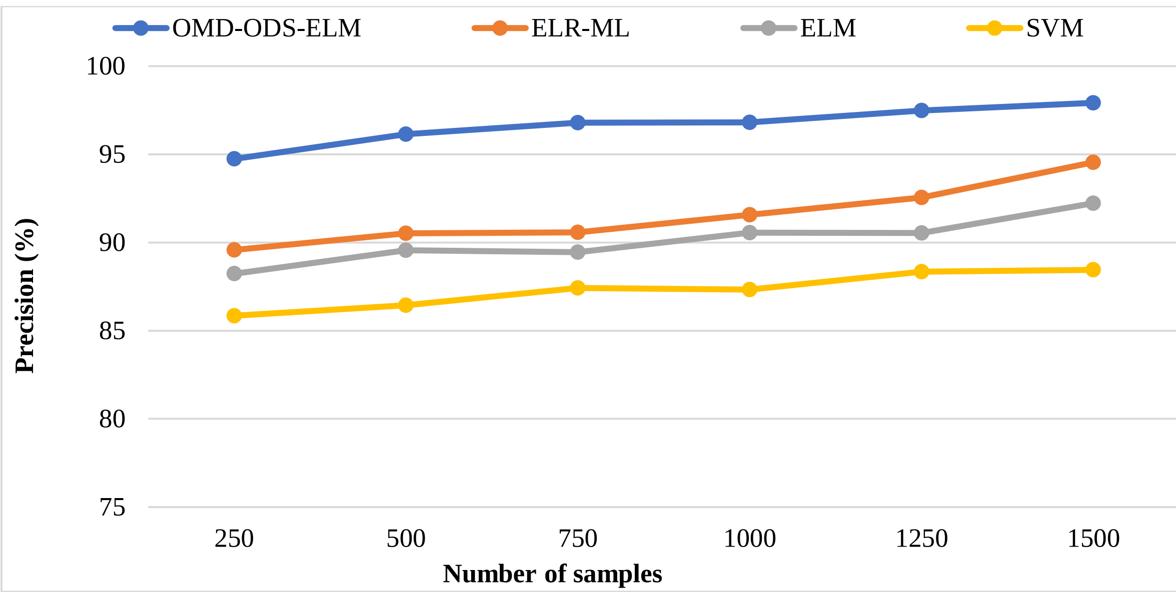

Fig. 8 Precision results comparison

For example, the OMD-ODS-ELM has higher precision than earlier approaches (97.92%, 95.07%, and 98.74% for BSE, S&P, and DJIA, respectively). When SPC-CNN and OMD-ODS-ELM are combined, loss values and overall performance across precision metrics produce the most significant results. Financial data can often be noisy and unpredictable.

An improved optimization algorithm may enhance the model's ability to filter out noise and focus on relevant patterns, contributing to better SITP precision. The fact that the proposed OMD-ODS-ELM technique outperforms the standard model demonstrates its importance. The numerical results of precision for the proposed and existing methods are shown in Table 4.

Traditional ML models are susceptible to the randomness of hyperparameters. The proposed IHS optimization algorithm expedites the convergence of the OMD-ODS-ELM model by providing optimal hyperparameters. Figures 11-13 depict the recall for the proposed OMD-ODS-ELM and other contemporary models, such as ELR-ML, ELM, and SVM, across a range of feature counts. As the number of patterns increases, the recall also rises. For instance, the OMD-ODS-ELM achieves recall values of 97.65%, 96.72%, and 98.18% for BSE, S&P, and DJIA, respectively, surpassing the performance of any prior approaches.

Table 4. The numerical results of precision for projected and existing methods

Number of samples	BSE				S&P				DJIA			
	OMD-ODS-ELM	ELR-ML	ELM	SVM	OMD-ODS-ELM	ELR-ML	ELM	SVM	OMD-ODS-ELM	ELR-ML	ELM	SVM
250	94.74	89.58	88.23	85.85	92.78	87.62	86.27	83.89	95.74	90.58	89.23	85.25
500	96.14	90.53	89.56	86.45	92.9	87.29	86.32	83.21	97.59	94.78	93.81	87.25
750	96.8	90.58	89.45	87.42	92.55	86.33	85.2	83.17	98.25	94.83	93.7	89.41
1000	96.81	91.58	90.56	87.34	92.57	87.34	86.32	83.1	98.26	95.83	94.81	91.52
1250	97.48	92.55	90.54	88.34	94.21	89.28	87.27	85.07	98.93	96.8	94.79	92.45
1500	97.92	94.55	92.23	88.45	95.07	91.7	89.38	85.6	98.74	97.58	95.23	93.85

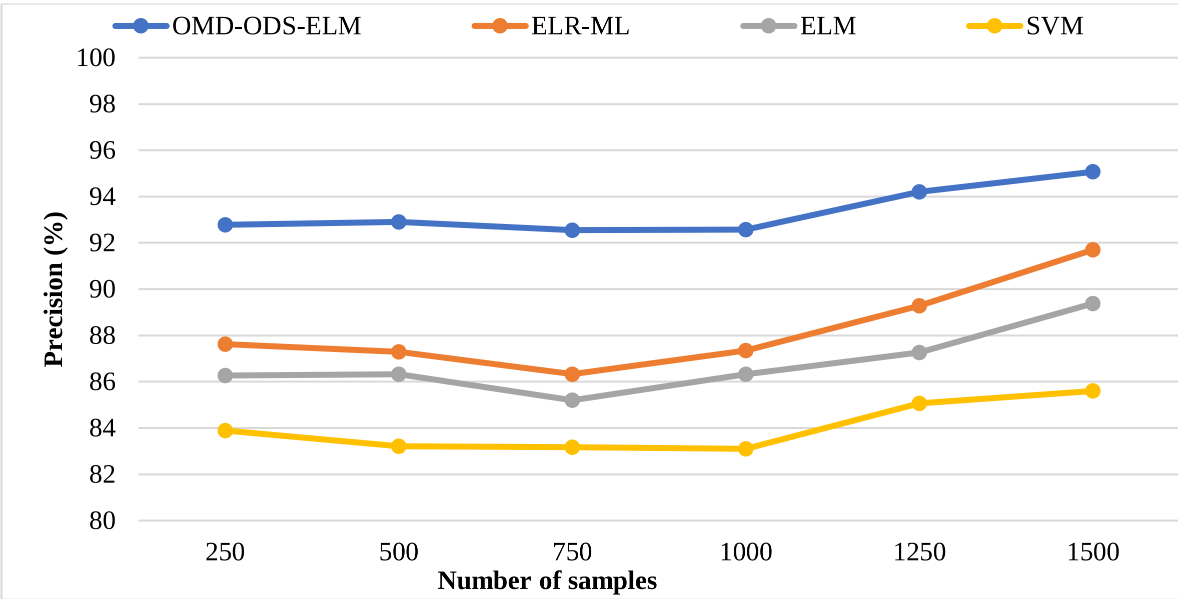

Fig. 9 Precision results comparison for S& P

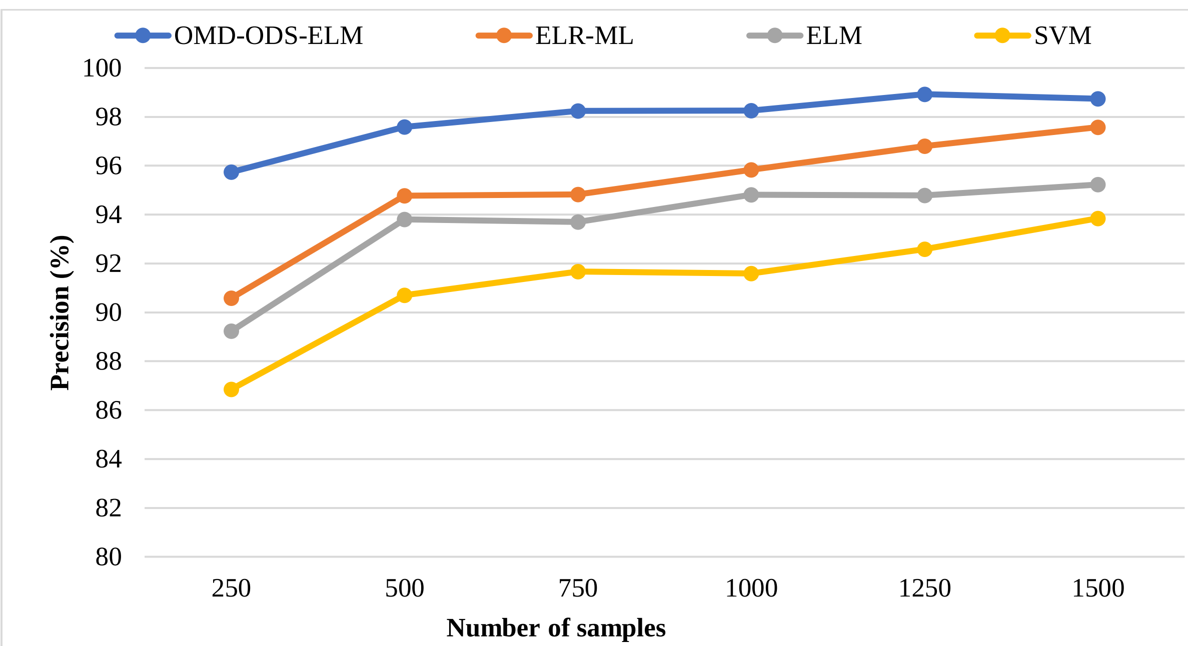

Fig. 10 Precision results comparison for DJIA

The optimized DSELM-OMD model, facilitated by an advanced Harmony Search Algorithm, exhibits faster convergence and improved generalization. This makes it more suitable for real-time prediction applications, enabling traders and investors to make timely decisions based on the latest market information. According to the findings, the proposed OMD-ODS-ELM model demonstrates superior stock trend prediction performance compared to all other models. The numerical recall of precision for both proposed and prevailing methods is presented in Table 5.

Applying the OMD-ODS-ELM method independently to the weight and bias vectors, the proposed SPC-CNN and OMD-ODS-ELM model undergo fine-tuning. The proposed OMD-ODS-ELM exhibits a higher F1-score for the number of patterns in the provided databases compared to existing models such as OMD-ODS-ELM, ELR-ML, ELM, and SVM (refer to Figures 14-16). Both the epoch count and the F-measure are concurrently tuned. Compared to other models, the OMD-ODS-ELM achieves an F-measure of 98.91%, 97.98%, and 99.44% for BSE, S&P, and DJIA, respectively. The enhanced algorithm may contribute to better generalization of the DSELM-OMD model to unseen data, a crucial aspect of robust SITP.

This improvement can mitigate overfitting and enhance the model's adaptability to market conditions. OMD allows for a detailed examination of various oscillatory patterns in the time series, facilitating the capture of inherent structures and patterns in SP movements. This, in turn, contributes to more accurate predictions, leading to the achieved high F1 score. The numerical results of the F1-score for both proposed and prevailing methods are presented in Table 6.

Table 5. The numerical results of recall for proposed and existing methods

Number of samples	BSE				S&P				DJIA			
	OMD-ODS-ELM	ELR-ML	ELM	SVM	OMD-ODS-ELM	ELR-ML	ELM	SVM	OMD-ODS-ELM	ELR-ML	ELM	SVM
250	93.79	88.63	87.28	84.9	94.03	88.87	87.52	85.14	93.79	88.63	87.28	86.96
500	95.29	89.68	88.71	85.5	94.55	88.94	87.97	86.52	95.44	92.63	91.66	88.55
750	96.17	89.95	88.82	86.47	94.2	89	88.65	87.98	96.52	92.54	92.14	89.42
1000	96.73	91.8	89.79	86.39	94.22	89.65	89.21	88.52	96.72	94.29	93.27	90.05
1250	97.2	93.83	91.51	87.39	95.86	90.93	89.65	88.96	97.83	95.7	93.69	91.49
1500	97.65	94.2	92	89.65	96.72	93.35	91.03	89.52	98.18	97.02	94.67	93.29

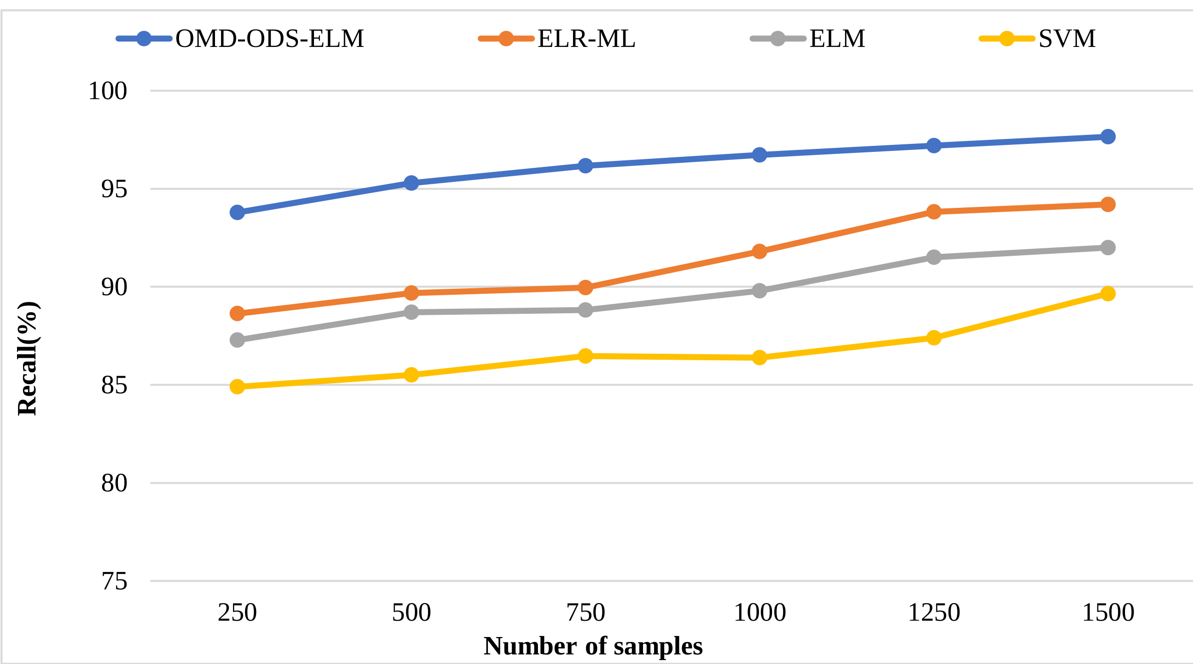

Fig. 11 Recall results comparison for BSE

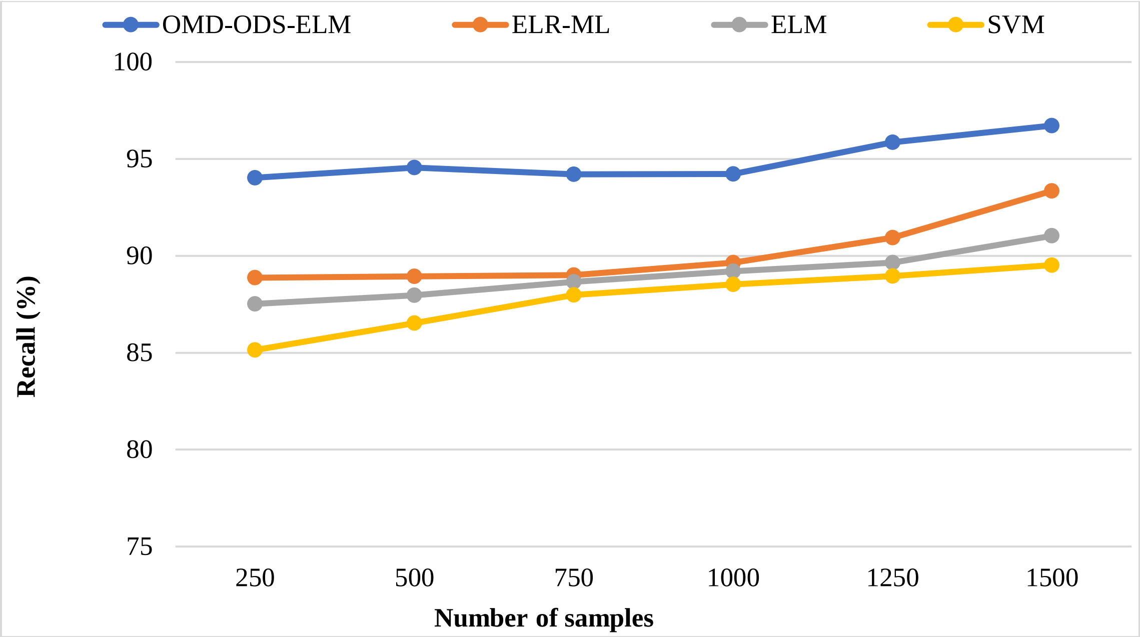

Fig. 12 Recall results comparison for S and P

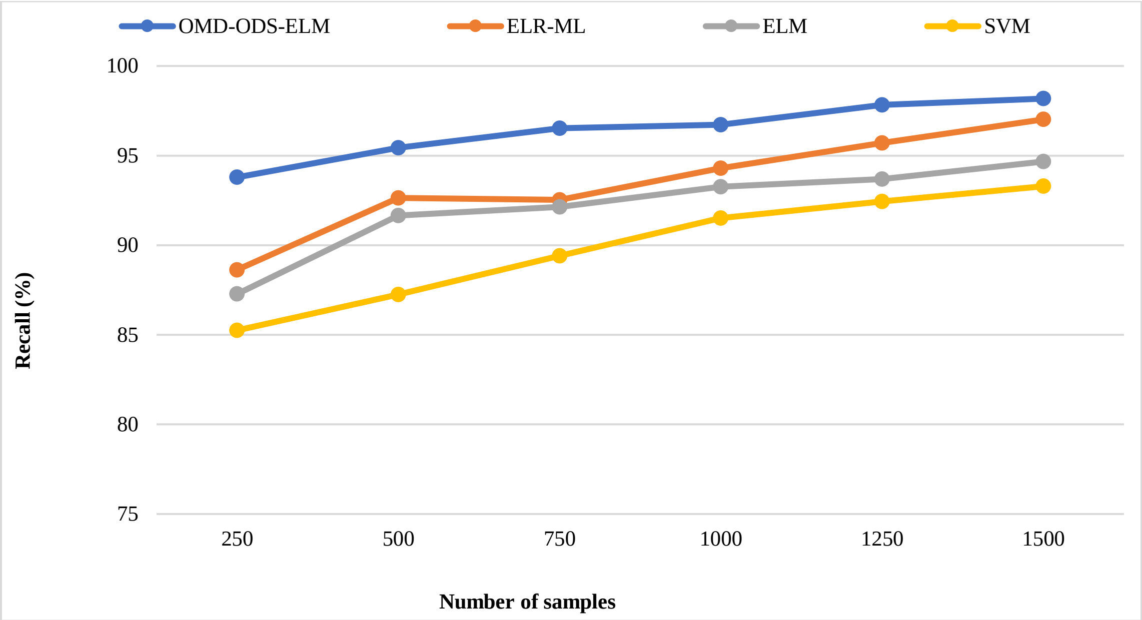

Fig. 13 Recall results comparison for DJIA

Table 6. The numerical results of the F1-Score for proposed and existing methods

Number of samples	BSE				S&P				DJIA			
	OMD-ODS-ELM	ELR-ML	ELM	SVM	OMD-ODS-ELM	ELR-ML	ELM	SVM	OMD-ODS-ELM	ELR-ML	ELM	SVM
250	95.31	90.15	88.8	86.42	95.55	90.39	89.04	86.66	95.31	90.15	88.8	86.42
500	96.43	91.03	89.96	86.64	95.79	90.08	89.53	87.66	96.58	93.77	92.8	89.69
750	97.73	91.51	90.57	88.03	95.85	90.56	90.4	89.43	98.08	94.1	93.37	90.98
1000	97.95	93.02	91.14	88.34	95.97	90.87	91.16	90.04	98.67	95.51	94.49	91.27
1250	98.62	95.25	93.07	89.04	97.28	92.35	92.68	92.96	99.25	97.12	95.11	92.91
1500	98.91	96.52	93.89	90.91	97.98	94.61	93.78	93.52	99.44	98.28	95.93	94.55

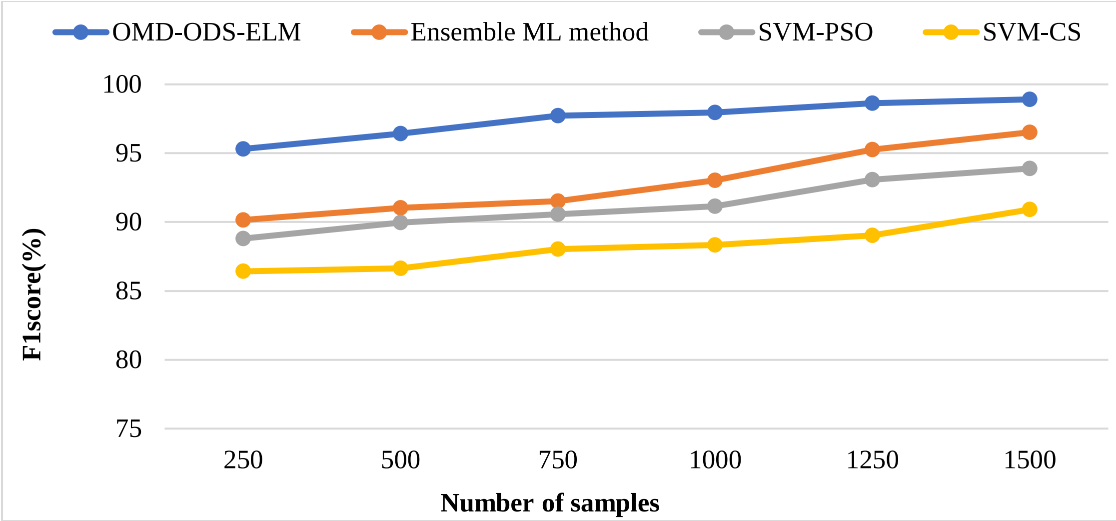

Fig. 14 F1-score result comparison for BSE

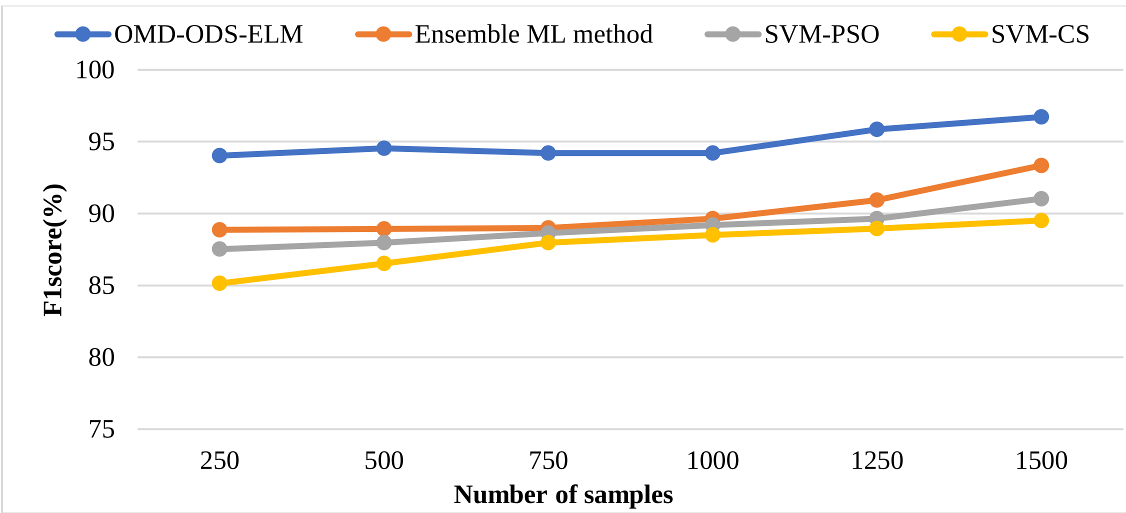

Fig. 15 F1-score result comparison for S& P

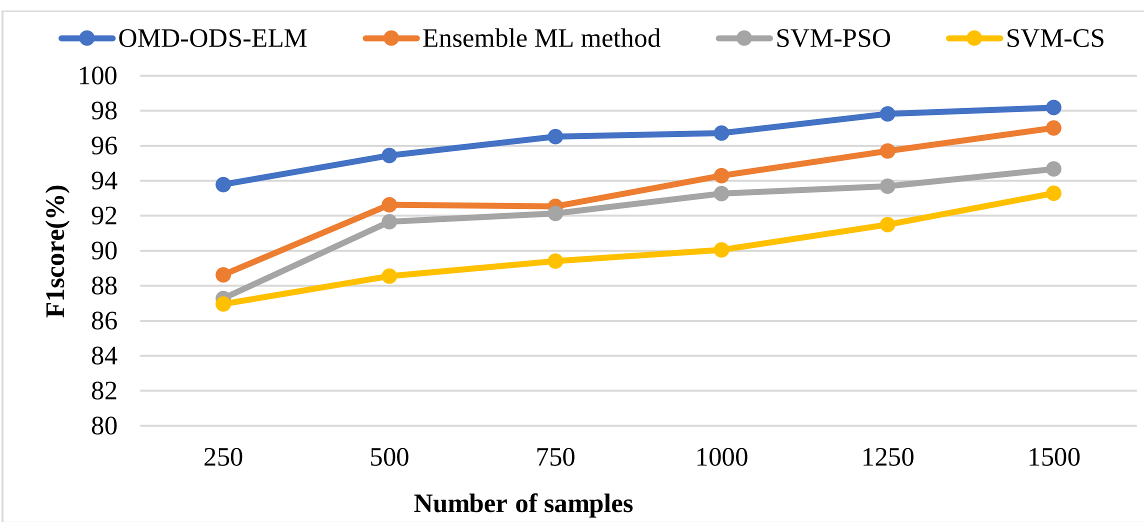

Fig. 16 F1-score result comparison for DJIA

5. Conclusion

Predicting SI values and directional trends is highly beneficial for investors and brokerage firms. However, accurately forecasting the intricate, non-linear, and erratic nature of SI time series poses significant challenges. This study introduces a predictor for SIT based on the OMD-ODS-ELM model. This model utilizes the IHS optimization technique, which avoids arbitrary weight assignment and enhances the training of the DSELM for SITP. The study demonstrated that the OMD-ODS-ELM outperformed baseline machine learning approaches across three benchmark datasets in most test cases. It performs better than ELR-ML, ELM, and SVM in various evaluation metrics. Furthermore, the IHS optimization technique exhibits quicker convergence in exploring and exploiting solutions than other benchmark optimization techniques. Overall, the proposed OMD-ODS-ELM-based approach has the potential to enhance the reliability and precision of SITPs significantly. It offers an effective solution for addressing the challenges posed by the intricate and erratic nature of SI time-series data. Indeed, there exists a prospective way for future research in the domain of SITP utilizing ML: Investigate the application of deep learning architectures, such as LSTMs, recurrent neural networks (RNNs), and transformer models, to enhance time-series prediction. These sophisticated architectures can capture intricate temporal dependencies within financial data, potentially surpassing the performance of conventional ML models. Explore advanced techniques for feature engineering, including sentiment analysis of financial news, social media data, and macroeconomic indicators. A diverse range of features can offer a more holistic understanding of market dynamics, potentially improving predictions. Emphasize the development of interpretable ML models to enhance transparency and trust in predictions. Unravelling the contributing factors to predictions can empower users to make more informed decisions. Methods such as LIME (Local Interpretable Model-agnostic Explanations) or SHAP (SHapley Additive exPlanations) can be explored for this objective.

Acknowledgements

The authors would like to thank the Department of Computer Science and Engineering, Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences (Deemed to be University), Chennai, India, for their support and motivation throughout this research.

References

[1] Galia Novakova Nedeltcheva, "Forecasting Stock Market Trends," *Economic Quality Control*, vol. 30, no. 1, pp. 21-38, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[2] Zelin Ying et al., "Predicting Stock Market Trends with Self-Supervised Learning," *Neurocomputing*, vol. 568, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[3] Wasiat Khan et al., "Predicting Stock Market Trends Using Machine Learning Algorithms via Public Sentiment and Political Situation Analysis," *Soft Computing*, vol. 24, no. 15, pp. 11019-11043, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[4] Rui Ren, Desheng Dash Wu, and Tianxiang Liu, "Forecasting Stock Market Movement Direction Using Sentiment Analysis and Support Vector Machine," *IEEE Systems Journal*, vol. 13, no. 1, pp. 760-770, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[5] Ashwini Saini, and Anoop Sharma, "Predicting the Unpredictable: An Application of Machine Learning Algorithms in Indian Stock Market," *Annals of Data Science*, vol. 9, no. 4, pp. 791-799, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[6] G. Kavitha, and A. Udhayakumar, "GA Based Stochastic Optimization for Stock Price Forecasting Using Fuzzy Time Series Hidden Markov Model," *International Journal of Pure and Applied Mathematics*, vol. 117, no. 1, pp. 143-171, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[7] Yuling Lin, Haixiang Guo, and Jinglu Hu, "An SVM-Based Approach for Stock Market Trend Prediction," *The 2013 International Joint Conference on Neural Networks*, Dallas, TX, USA, pp. 1-7, 2013. [CrossRef] [Google Scholar] [Publisher Link]

[8] Rudra Kalyan Nayak, Debahuti Mishra, and Amiya Kumar Rath, "A Naïve SVM-KNN Based Stock Market Trend Reversal Analysis for Indian Benchmark Indices," *Applied Soft Computing*, vol. 35, pp. 670-680, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[9] Lixuan Zhang et al., "A Hybrid Forecasting Method for Anticipating Stock Markets Trend via a Soft-Thresholding De-Noise Models and Support Vector Machines (SVM)," *World Basic and Applied Science Journal*, vol. 13, no. 3, pp. 597-602, 2023. [Google Scholar] [Publisher Link]

[10] Sahaj Singh Maini, and K. Govinda, "Stock Market Prediction Using Data Mining Techniques," *2017 International Conference on Intelligent Sustainable Systems*, Palladam, India, pp. 654-661, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[11] Fajiang Liu, and Jun Wang, "Fluctuation Prediction of Stock Market Index by Legendre Neural Network with Random Time Strength Function," *Neurocomputing*, vol. 83, pp. 12-21, 2012. [CrossRef] [Google Scholar] [Publisher Link]

[12] J. Margaret Sangeetha, and K. Joy Alfia, "Financial Stock Market Forecast Using Evaluated Linear Regression Based Machine Learning Technique," *Measurement: Sensors*, vol. 31, pp. 1-7, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[13] Shangkun Deng et al., "An Integrated Approach of Ensemble Learning Methods for Stock Index Prediction Using Investor Sentiments," *Expert Systems with Applications*, vol. 238, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[14] K. Venkateswararao, and B. Venkata Ramana Reddy, "LT-SMF: Long Term Stock Market Price Trend Prediction Using Optimal Hybrid Machine Learning Technique," *Artificial Intelligence Review*, vol. 56, no. 6, pp. 5365-5402, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[15] Manrui Jiang et al., "The Two-Stage Machine Learning Ensemble Models for Stock Price Prediction by Combining Mode Decomposition, Extreme Learning Machine and Improved Harmony Search Algorithm," *Annals of Operation Research*, vol. 309, pp. 553-585, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[16] Jun Zhang, and Xuedong Chen, "A Two-Stage Model for Stock Price Prediction Based on Variational Mode Decomposition and Ensemble Machine Learning Method," *Soft Computing*, vol. 28, no. 3, pp. 2385-2408, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[17] Yaohu Lin et al., "Stock Trend Prediction Using Candlestick Charting and Ensemble Machine Learning Techniques With a Novelty Feature Engineering Scheme," *IEEE Access*, vol. 9, pp. 101433-101446, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[18] Xianghui Yuan et al., "Integrated Long-Term Stock Selection Models Based on Feature Selection and Machine Learning Algorithms for China Stock Market," *IEEE Access*, vol. 8, pp. 22672-22685, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[19] Sidharth Samal, and Rajashree Dash, "Developing a Novel Stock Index Trend Predictor Model by Integrating Multiple Criteria Decision-Making with an Optimized Online Sequential Extreme Learning Machine," *Granular Computing*, vol. 8, no. 3, pp. 411-440, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[20] Rajashree Dash et al., "An Integrated TOPSIS Crow Search Based Classifier Ensemble: In Application to Stock Index Price Movement Prediction," *Applied Soft Computing*, vol. 85, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[21] Ramendra Prasad et al., "Soil Moisture Forecasting by a Hybrid Machine Learning Technique: ELM Integrated with Ensemble Empirical Mode Decomposition," *Geoderma*, vol. 330, pp. 136-161, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[22] Guang-Bin Huang, Dian Hui Wang, and Yuan Lan, "Extreme Learning Machines: A Survey," *International Journal of ML and Cybernetics*, vol. 2, pp. 107-122, 2011. [CrossRef] [Google Scholar] [Publisher Link]

[23] D. Manjares et al., "A Survey on Applications of the Harmony Search Algorithm," *Engineering Application of Artificial Intelligences*, vol. 26, no. 8, pp. 1818-1831, 2013. [CrossRef] [Google Scholar] [Publisher Link]

[24] Osama Moh'd Alia, and Rajeswari Mandava, "The Variants of the Harmony Search Algorithm: An Overview," *Artificial Intelligences Review*, vol. 36, pp. 49-68, 2011. [CrossRef] [Google Scholar] [Publisher Link]

[25] Liyun Fu et al., "Hybrid Harmony Search Differential Evolution Algorithm," *IEEE Access*, vol. 9, pp. 21532-21555, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[26] Jin Yi et al., "An Efficient Modified Harmony Search Algorithm with Intersect Mutation Operator and Cellular Local Search for Continuous Function Optimization Problems," *Applied Intelligence*, vol. 44, pp. 725-753, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[27] Daniel J. Poole, and Christian B. Allen, "Constrained Niching Using Differential Evolution," *Swarm and Evolutionary Computation*, vol. 44, pp. 74-100, 2019. [CrossRef] [Google Scholar] [Publisher Link]