SvedbergOpen
DISSEMINATION OF KNOWLEDGE
International Journal of Artificial Intelligence and Machine Learning
Publisher's Home Page: https://www.svedbergopen.com/
Research Paper Open Access

Ensemble Optimization Based Feature Selection (Eofs) And Ensemble Deep Learning (Edl) Classifier For Stock Market Prediction

Deepa Raghunathan[1*], M.Krishnamoorthi[2], Boobalakrishnan Sadasivam[3]

[1]Research Scholar,Department of Information Technology, Dr. N.G.P Institute of Technology, Coimbatore - 641048 Corresponding author Email id: deeparaghunathan2324@gmail.com Orchid id : 0000-0003-2434-2198
[2]Professor,Department of Information Technology, Dr.N.G.P Institute of Technology, Coimbatore - 641048
Email id: krishnamoorthi@drngpit.ac.in Orchid id : 0000-0001-9989-6764
[3]Research Scholar,Department of Computer Science, Christ University, Bangalore - 560029
Email id: Boobalkrishnan@gmail.com Orchid id : 0000-0001-8171-6622

ABSTRACT: The dynamic and complex financial markets are impacted by a large variety of social, political, and economic variables, stock market prediction has never been easy. This research proposes an advanced stock market forecasting structure by connecting the Ensemble optimization-based feature selection (EOFS) and the Ensemble Deep Learning (EDL) classifiers. The most relevant characteristics of the larger amount of financial data, the EOFS approach used the capabilities of many Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO), and genetic algorithm (GA). The EDL classifier is then trained using its selected characteristics, which improves the accuracy and credibility of the overall forecasts, with the forecasts of many DEEP Wanda education models, such as the Conventional Neural Network (CNN), long short -term memory (LSTM) and Deep Neural Network (DNN). Experimental data shows traditional stock forecasting models' maximum accuracy, precision, and recall. Indicated technology is a property for financial analysts and investors as it provides significant growth in predicting stock prices and trends.

Keywords: Deep neural networks, convolutional neural networks, long short-term memory, Ensemble Learning, Ant colony optimization, particle swarm optimization, genetic algorithms, Stock Market Prediction, Feature Selection, Deep Neural Networks.

1. INTRODUCTION

The stock market is inherently unpredictable and dynamic, it has long been one of the most difficult to forecast. Financial analysts, traders and investors can make a great deal of profit from accurate stock prices as they help them decide on buying, disposing of or maintaining [1]. However, the traditional statistical and machine stock market data is complex, non-linear and noisy, making it difficult to learn algorithms [2]. Nevertheless, traditional technologies such as autoregressive integrated moving average (ARIMA) models have been used in the past, which are often stuck in the middle of the complex patterns and dependence, especially when the market or unexpected circumstances. This promotes the need for more advanced techniques that can handle larger, multi-dimensional datasets and accurately predict future market trends.

The Ensemble Optimization-Based Facility Selection (EOFS) is an advanced technique that addresses the challenge of choice from a substantial collection of the most relevant and preacher stock market data [4]. Technical indicators, trading volume, historical hymn prices, and macroeconomic variables are some of the many characteristics that are often included in stock market data. Machine learning models' efficacy may be blocked by the sound represented by many of these characteristics, many of which are unnecessary or futile. EOFS Multiple OPTIM PTMISSION Algorithms such as PSO [5], GA [6], and WOA for convenience and convenience. Faces to identify the most significant features. This multi-layered method reduces parameters, improves the prediction model's precision and interpretation, and assures that the selected characteristics are the most consistent [8].

In combination with EOF, Ensemble Deep Learning (EDL) classifiers offer powerful remedies for the stock market forecast. To provide a forecasting structure that is accurate and dependable, EDL integrates the capabilities of many deep learning models. CNN [10] and LSTM [11] standard deep learning

systems may find complex time-series patterns and connections. By connecting these models to a connection configuration, the system can better normalize in different market conditions. LSTM is perfect for learning long -term dependence, which makes them suitable for modelling sequences like stock prices, while CNN is exceptionally effective in identifying the spatial patterns in financial data. EDL prevents overfitting and improves predictions by using several deep learning models, especially during unstable market conditions [12].

The objective of this research is to integrate EDL and EOFS classifiers in a unified structure for a more precise and effective forecast of the stock market. The EOFS technology best improves the accuracy of the forecast by applying the convenience selection process by applying various Optimization algorithms, while the EDL component improves the accuracy of the forecast by benefiting the combination of deep learning models. By integrating these two methods, the proposed approach offers an entire solution that responds to the stock market's instability and improves prediction accuracy. The following sections will evaluate the stock market prediction approach, which will go into methods, experiments and results.

2. LITERATURE REVIEW

Khan et al [2022] [13] selection and spam tweet reduction on datasets were implemented in an attempt to enhance the quality of performance and forecasts. To establish which stock markets were more susceptible to social media and financial news and which were harder to forecast, tests were done. The objective was to constantly find classifieds by comparing the output of many methods. Eventually, many classifiers were connected and to get the maximum prediction accuracy, deep learning was used. According to the results of the experiment, thus, financial news and social media produce accuracy 80.53% and 75.16% were the largest predictions. Additionally, the results demonstrated the unpredictability of the Red Hat and New York stock markets, that social media affected more IBM and New York stocks, and financial news was more affected on Microsoft and London stocks. It was found that the Random Forest classifier was dependable, and with 83.22% efficiency, its ensemble performed most effective.

Bansal et al [2022] [14] provided five forecasting techniques for 12 leading Indian stock companies: K-nearest neighbours, long short-term memory, decision tree regression, support vector regression, and linear regression. Twelve company's stock price datasets over the previous seven years were collected and employed in a data-intensive implementation that followed substantial study on different elements of using the stock market to apply machine learning. Additionally, several effective and reliable methods for predicting stock market movements are highlighted in the paper. A step-by-step discussion of the approach used to get the results was provided. To assist in better analysis, a comprehensive comparative analysis of the previously shown algorithms' stock price prediction performances was also conducted. The findings were shown in easily readable tabular and graphical formats. After presenting the results of this innovative, comprehensive study, it was decided that DL. The algorithm works better than any other algorithm to predict the stock price or time range, providing results with a high degree of accuracy.

Ampomah et al [2021] [15] the Gaussian Naïve Bayes (GNB) machine learning method suggested for predicting stock prices by existing research in release has not been sufficient attention. To overcome that distance, this study is a GNB. The algorithm stock is assessed in the way the various features are combined with scaling and convenience extraction methods in the forecast for movement. Kendal's compatibility test was used to rate the influence of GNB models in a series of assessment measures. The predictive model using the GNB algorithm and Linear Discriminant Analysis (GNB_LDA) outperformed all other GNB models in accuracy, F1-score, and AUC. Similar to this, the prediction model that used specificity results and was based on the GNB algorithm, Min-Max scaling, and Principal Component Analysis (PCA) obtained the best rank. Additionally, while using the Min-Max scaling strategy instead of standardized scaling strategies, the GNB algorithm performed better.

El Mahjouby et al [2024] [16] adaboost, gradient boosting, logistic regression, bagging, Decision tree, random forest, and Gaussian Naïve Bayes classifiers, and our method, which combined three models, were among the machine-learning techniques used. Predicting the most favourable periods to purchase and sell the euro in relation to the dollar was the objective. To increase the accuracy of the methods and approaches, the training dataset included a number of different technical indicators. With a 0.948 precision, the results of our experiment show that our strategy performs best with competitive approaches in terms of forecasts. To help them determine the maximum time of buying and selling in the market, this research gives investors the ability to make an accurate decision about their future EUR/USD transactions.

Jagadesh et al [2024] [17] suggested an all-encompassing strategy for stock market data analysis that combines effective feature selection, data preparation, and classification techniques. To make the data ready for further examination, the wavelet transform technique is used to clean and minimize noise. To increase model performance and reduce dimensionality, the Dandelion Optimization Algorithm (DOA)

selects the most important input attributes. A novel hybrid model termed 3D-CNN-GRU is created to improve prediction capabilities by combining the capabilities of a Gated Recurrent Unit (GRU) with a 3D CNN. This hybrid approach aims to capture more effectively temporary and regional patterns in stock market data. To further the influence of the model, optimize the hyperparameters adjustment is done using a blood coagulation algorithm (BCA). The approach shows its reliability and effectiveness in the stock market forecast application with impressive prediction accuracy of 99.14%. However, the dataset has only the Nifty 50 index, which restricts the application of the model. To apply it, future work can include additional datasets and test the model in different market conditions. This will not only recognize the results but also increase the generalization of the approach. To guarantee the stability and efficiency of the model under various market situations, future studies may focus on changes to this approach for use under other financial circumstances.

Pagliaro [2023] [18] presented a new technique that uses an Extra Trees Classifier (ETC) for predicting stock returns. The purpose is that there is a percentage difference between closed prices and closing prices for 120 companies across various sectors after 10 trading days. We train our model using technical indicators. Finding a balance between accuracy and utility period 10-day period provides good forecast horizons to traders. Large period of improvement and insufficiency of short people are avoided with this period. In addition to its ability to manage the model's managing the large datasets in the number of input characteristics and reducing overfitting, the Extra Trees Classifier method is especially well-suited for the forecast of the classified method. At an accuracy of .1 86.1%, our results show that the ETC model displays more effectively than random forest technology, which is a more traditional approach. According to these results, additional trees are a useful tool to accurately predict major fluctuations in stock market prices. The overall findings of this study show how classified methods, especially additional trees, can improve the stock market predictions. The results provide valuable insights for merchants and researchers seeking the benefit of machine learning for more accurate market predictions.

Ho and Huang [2021] [19] develop a multichannel collaborative network using social media and candlestick chart data to anticipate market movements. To start, extracted social media sentiment features and analyzed Twitter data using the Natural Language Toolkit. A graphic that makes use of the stock's historical time series data with candlesticks was created to demonstrate stock movement patterns. Finally, we combined emotion variables with candlestick charts to anticipate stock price movement over 4, 6, 8, and 10 days. The collaborative network was divided into two branches. The first branch used social media data to classify sentiment using a one-dimensional CNN. A two-dimensional (2D) CNN that examined 2D candlestick chart data was used in the second branch for image classification. The collaborative network produced encouraging results, surpassing single-network models that just relied on emotion data or candlestick charts, according to our evaluation of our model on five highly desirable stocks: Apple, Tesla, IBM, Amazon, and Google. The accuracy of the suggested technique for Apple stock was 75.38%, which was its best performance. We also found that the accuracy of stock price predictions was better over longer time periods than shorter ones, indicating that the model works better for longer-term trend prediction. This approach provides a robust framework for leveraging both social media sentiment and technical indicators in stock market predictions.

Worasucheep [2022] [20] the suggested ensemble model incorporates Extreme Gradient Boosting (XGBoost), Light Gradient Boosting Machine (LightGBM), Random Forest, Support Vector Machine (SVM), and Artificial Neural Networks (ANN). To select a maximum of 45 technical indicators to use as input characteristics of the model, the convenience selection is required. To improve its influence, each basic classification is subject to a complete hyperparameters change process. To obtain the result of the final prediction, the forecasts of five basic classifieds The updated majority voting procedure is used to combine that prefers the top three classifications with the best accuracy. Using three overlapping datasets spreading at five -year intervals between 2014 and 2020, representing various market conditions, assessing the performance of the ensemble classifier of Stock Exchange Thailand (SET) using daily history of 20 equities. According to the results of the experiment, the suggested Ensemble Classifier performs significantly more effectively in terms of trading returns and sharp ratios than in combination with individual base classifiers and simple majority voting. This stock highlights the strength and effectiveness of the attachment approach for the market forecast.

BL and BR [2023] [21] suggested a sentiment analytical prediction methodology that used sentiment data from stocks and the news. Technical indicators such the relative strength index (RSI), moving average (MA), and moving average convergence divergence (MACD) were taken from the stock data. Pre-processing, which included sentiment categorization and keyword extraction; (2) keyword extraction using WordNet and sentiment categorization; (3) feature extraction, which involved the extraction of suggested holoentropy-based features; and (4) classification, which used a deep neural network to return the sentiment output, were all done concurrently with the processing of the

sentiments by analyzing the news data. To increase sentiment prediction accuracy, the neural network was trained using the self-improved whale optimization algorithm (SIWOA). An improved deep belief network (DBN) predicted market movements using news data sentiment and stock data features. The revised SIWOA modified DBN weights, improving model functionality.

Muhammad et al [2024] [22] a suggested to forecast these five stock market trends: rounded top, double top, upward, down and upwards. Using the average accuracy of 994.9%, the model surpassed the popular benchmark, including the model LR, RF and SVM. Random Forest shows the accuracy of 85.7%, the SVM shows the accuracy of 60.07%, and the logistic regression shows 52.45% accuracy. Furthermore, across four real-world heterogeneous datasets, the suggested model outperformed random forest (77.95%), SVM (21.02%), and logistic regression (46.23%) in terms of F1-score, achieving 94.85%. Explainable AI (XAI) approaches, such SHAP and LIME, were used to improve interpretability and provide stakeholders a better understanding of the determinants of predictions. The top ten significant and impactful characteristics that allowed for feature reduction without sacrificing model performance were identified using the SHAP analysis. There seems to be a trade-off between comprehensiveness and performance, as precision, recall, and F1-score all increased but accuracy somewhat declined with the top 10 characteristics. The results show that the model can make financial decisions while maintaining interpretability and predictive capacity, which helps investors with risk management and long-term planning.

3. PROPOSED METHODOLOGY

In this work, EOFS and EDL have been introduced for feature selection and classification of stock market prediction. EOFS is introduced that combines the procedure of several methods like Adaptive Manta Ray Foraging Optimization (AMRFO), Improved Dung Beetle Optimizer (IDBO) and Modified Donkey and Smuggler Optimization (MDSO) and a Mutual Information (MI) is used to combine the results of these methods. The HDFC developed an EDL-based SMP model for COVID-19, Twitter, stocks, and weather. Overall process of proposed system is illustrated in figure 1.

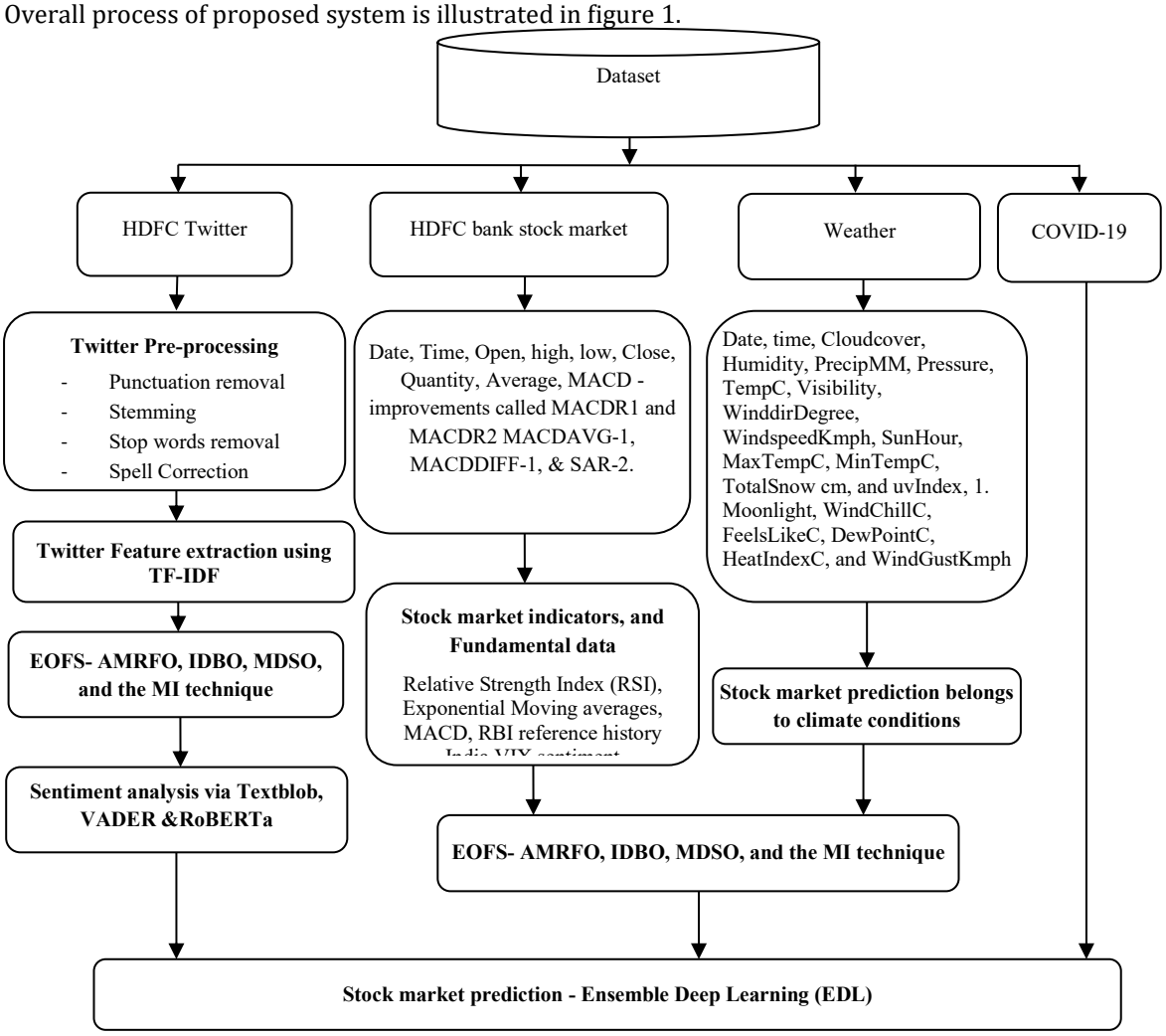

Figure 1. Ensemble Deep Learning (EDL) with a Stock Market Prediction

3.1. Tweet preprocessing methods

The data is first gathered from four distinct domains, including COVID-19, Twitter, banks, and weather. In accordance with their individual processes, the samples are collected and pre-processed. For the text dataset, the pre-processing includes removing punctuation, applying stemming, eliminating stop words, correcting spelling, and finalizing the pre-processed tweet dataset.

3.2. Tweet Feature extraction

This research uses a unigram model to manually categorize the tweets, which gives the dataset sufficient coverage. Training data is used for data pre-processing and feature development to convert the tweets into feature vectors, to identify the qualities that are most relevant. The TF–IDF approach converts text into fixed feature vectors and uses this information to represent text and tweets as vectors. Term frequency (TF) is a measure of how often a phrase is used in a particular piece of text. The reverse of this is the inverse document frequency (IDF), quantifies the frequency with which a certain phrase occurs over the whole corpus or dataset. The computational cost of the TF-IDF approach is linear and depends on the amount of lines of text and words in each line. The dataset comprises a variety of sentiment analysis documents with distinct polarity phrases and words.

3.3. Ensemble Optimization Based Feature Selection (EOFS)

Selection of facilities is essential to overlapping and reducing the curse of dimension. It is important to identify the most impressive feature sets can significantly affect target feature forecasts. This process is especially important in the stock market, considering its dynamic environment and distinctive characteristics of each field. Using dynamic methods for facility selection against a stable approach, it is necessary to keep the stock market always with changing conditions and field-specific features. It is important to determine the right combination of features for each scenario. Therefore, focusing on efficient facility management in the forecast for stock prices becomes an important and essential challenge. The EOFS model has been introduced, which combines feature sets of various feature selection algorithms such as AMRFO, IDBO and MDSO in ranking or general sequences. The EOFS model improve the effective feature selection of algorithms and the importance of each feature.

3.4. Adaptive Manta Ray Foraging Optimization (AMRFO)

To select characteristics from the HDFC bank stock dataset and the weather dataset as efficiently as possible, the AMRFO algorithm is predicated on somersault, cyclone, and chain foraging behaviours [23]. The following is an outline of the mathematical models.

Chain foraging: Manta rays may locate plankton using the AMRFO algorithm and approach it. A location's suitability selecting the best features from the dataset of HDFC Bank stocks and the weather dataset increases with the concentration of plankton present [24]. The best answer found so far is treated by the AMRFO algorithm as the plankton that highly concentrated manta rays seek to consume. The head-to-tail alignment of the manta rays, creating a foraging chain to optimally select features from the HDFC bank stock dataset and the Weather dataset. In each iteration, individuals make a move not only towards the optimal selection of features but also towards the feature ahead of it. The best approach found so far is used to update each feature and the optimal features from the HDFC bank stock dataset and the Weather dataset. The expression for the mathematical model of chain foraging is equations (1-2).

$$x_i^d(t+1) = \begin{cases} x_i^d\left(t + r.\left(x_{best}^d(t) - x_i^d(t)\right) + \alpha.\left(x_{best}^d(t) - x_i^d(t)\right)\right) & i = 1 \\ x_i^d\left(t + r.\left(x_{i-1}^d(t) - x_i^d(t)\right) + \alpha.\left(x_{best}^d(t) - x_i^d(t)\right)\right) & i = 1, \dots N \end{cases} \quad (1)$$

$$\alpha = 2.r.\sqrt{|\log(r)|} \quad (2)$$

where r is a weight coefficient and is a random vector [0, 1], $x_{best}^d(t)$ the plankton with the highest amount is the most effective feature extracted from the dataset, and $x_i^d(t)$ is the position in the d^{th} dimension of the i^{th} individual at time t. This foraging activity is shown in a 2-D environment in Figure 2. The positioning $x_{best}(t)$ of the food and the feature position $x_{i-1}(t)$ of the (i-1)th position update of the present individual is defined by the i^{th} feature (individual).

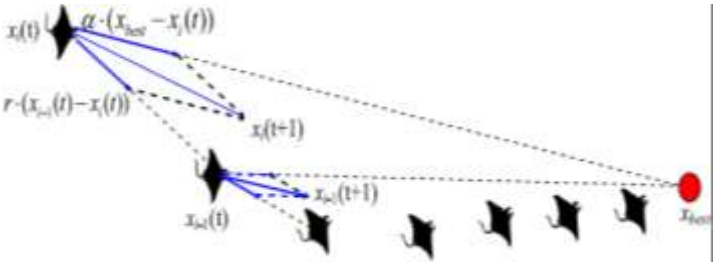

Figure 2. Foraging Behavior of Chains in 2-D

Cyclone foraging: In deep water, when a manta school detects a plankton concentration representing the optimal features in the HDFC bank stock and Weather datasets they organize into an extended foraging chain and spiral inward toward the food source (i.e., the highest-accuracy solution). Manta rays use a cyclone foraging technique in which each the ray swims toward the manta ray in front of it and the food source (which has the best accuracy). In this way, the swarm moves in a coordinated spiral while foraging. Figure 3 depicts cyclone foraging in a two-dimensional space, where each individual (representing the optimal feature selection for the HDFC bank stock and Weather datasets) both trails the one ahead and spirals inward toward the food (i.e., the highest accuracy).

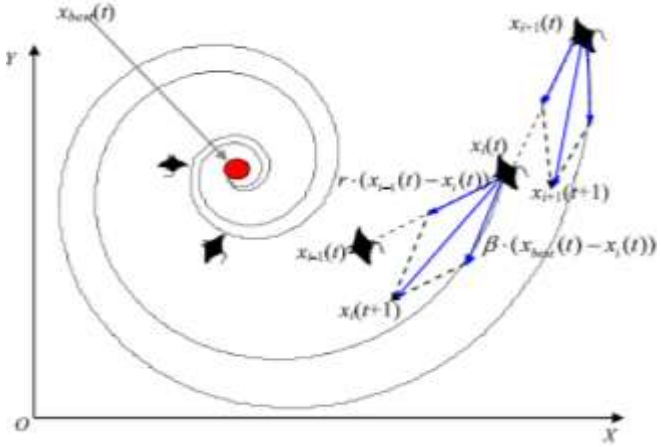

Figure 3 Foraging Behavior of Cyclones in a 2-D Space

To determine the dataset's optimal attributes, the mathematical modelling of the manta rays' spiral movement in two dimensions may be described as equation (3),

$$\begin{cases} X_i(t+1) = X_{best} + r.\left(X_{i-1}(t) - X_i(t)\right) + e^{bwe_m}.cos(2\pi w).\left(X_{best}(t) - X_i(t)\right) \\ Y_i(t+1) = Y_{best} + r.\left(Y_{i-1}(t) - Y_i(t)\right) + e^{bwe_m}.sin(2\pi w).\left(Y_{best} - Y_i(t)\right) \end{cases} \quad (3)$$

where we_m is a mutation weight between range [0, 1]. This weight value is generated based on the weight between the attribute ranges from HDFC bank stock dataset, and Weather dataset. This cyclone foraging mathematical model may be defined simple by equations (4-5),

$$x_i^d(t+1) = \begin{cases} x_{best}^d + r.\left(x_{best}^d(t) - x_i^d(t)\right) + \beta.we_m\left(x_{best}^d(t) - x_i^d(t)\right) \; i = 1 \\ x_{best}^d + r.\left(x_{i-1}^d(t) - x_i^d(t)\right) + \beta.we_m\left(x_{best}^d(t) - x_i^d(t)\right) \; i = 2, \ldots., N \end{cases} \quad (4)$$

$$\beta = 2e^{r_1\frac{T-t+1}{T}}.sin(2\pi r_1) \quad (5)$$

where T is the most iterations, r_1 is the rand number between [0, 1], and β is the weight coefficient. To enable the cyclone foraging strategy to efficiently utilize the area with the best address identified, each (feature) does a random search using the food (accuracy) as their reference point. Additionally, this behavior makes a substantial contribution to improving exploration. In establishing as their reference, a fresh, random, optimal position across the search area, each person is encouraged to investigate an entirely different feature position from the most one at the moment. This method enables the AMRFO algorithm to do a thorough worldwide search and focuses mostly on exploration. The corresponding mathematical equations are given in equations (6-7).

$$x_{rand}^d = Lb^d + r.(Ub^d - Lb^d) \quad (6)$$

$$x_i^d(t+1) = \begin{cases} x_{rand}^d + r.\left(x_{rand}^d(t) - x_i^d(t)\right) + \beta.we_m\left(x_{rand}^d - x_i^d(t)\right) \; i = 1 \\ x_{rand}^d + r.\left(x_{i-1}^d(t) - x_i^d(t)\right) + \beta.we_m\left(x_{rand}^d - x_i^d(t)\right) \; i = 2, \ldots., N \end{cases} \quad (7)$$

x_{rand}^{d} is a randomly generated point in the search space, while Lb^{d} and Ub^{d} represent the lower and upper limits of the dth dimension.

Somersault foraging: The position of the food serves as the focal point of this behavior. In turning and somersaulting to a new feature position, each individual advances towards it. Therefore, they keep updating their feature positions based on the best option thus far provided. Equation (8) serves as the mathematical model for this phenomenon,

$$x_i^d(t+1) = x_i^d(t) + S.\left(r_2.x_{best}^d - r_3.x_i^d(t)\right), i = 1, \ldots, N \quad (8)$$

When S = 2, r_2 and r_3 S is the turnover element responsible for determining the manta rays' declining range, and to the interval [0, 1] and are two random numbers. In a new search space, any feature may relocate to any position in relation to the best feature found so far, among where it is now and where it is symmetric, as shown by equation (8), by specifying the turn-over range. The disturbance applied to the current feature position reduces with the distance between the position of each distinct feature as well as the best feature discovered so far. Every individual travels in the direction of the best feature approach inside the scope of the search. As a result, tumble foraging's range is adaptively decreased over time. The AMRFO algorithm's somersault foraging behavior diagram is shown in Figure 4.

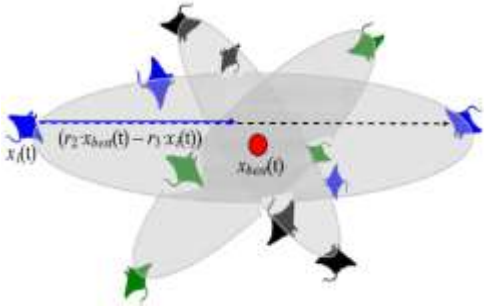

Figure 4 Somersault Foraging Behavior in AMRFO

In Oppositional-Based Learning (OBL), the populations adjust their positions by moving to their present placement's mirror or opposite position as follows:

$$x_{i,j}^{o}(t+1) = Ub_j + Lb_j - X_{i,j}, i = 1, \ldots, N \quad (9)$$

For the most suitable feature selection from the HDFC bank stock dataset and the weather dataset, the AMRFO method starts with creating a random population. Each individual (feature) modifies its position during each iteration by taking into account both the reference position and the one in front of it. Exploration and exploitation may be balanced throughout the search process when from 1/T approaches 1, the value of t/T decreases. The best feature solution at present for t/T < rand is the exploitation reference position. On the other hand, the reference position for exploration is selected at random from the search space when t/T > rand. The AMRFO algorithm may simultaneously alternate between cyclone and chain foraging behavior depending on the random number. The individuals (features) subsequently modify their positions in accordance with the most effective somersault foraging strategy. Until the stoppage condition is met, every computation and update is performed interactively. It returns the most noticeable person's fitness and feature position. AMRFO's pseudocode is provided in Algorithm 1.

Algorithm 1. Adaptive Manta Ray Foraging Optimization (AMRFO)

Select on the population size (N), the maximum number of repetitions (T), and the starting values for each manta ray

$x_i(t)=x_l+rand(x_u-x_l)$

for i-1,...N and t=1

Determine the fitness of each individual using the equation $f_i=f(x_i)$

Determine the best answer so far, x_{best}. The top and lowest limits of feature space are denoted by x_u and x_l

WHILE the stop requirement is not met, do

For i=1 to N do

IF rand<0.5 then// cyclone foraging

IF t/T_{max}<rand THEN

$$x_{rand} = x_l + rand.(x_u - x_l)$$

$$x_i(t+1) = \begin{cases} x_{rand} + r.\left(x_{rand} - x_i(t)\right) + \beta.we_m\left(x_{rand} - x_i(t)\right) & i = 1 \\ x_{rand} + r.\left(x_{i-1}(t) - x_i(t)\right) + \beta.we_m\left(x_{rand} - x_i(t)\right) & i = 2, \ldots, N \end{cases}$$

ELSE

$$x_i(t+1) = \begin{cases} x_{best} + r.\left(x_{best} - x_i(t)\right) + \beta.we_m\left(x_{best} - x_i(t)\right) i = 1 \\ x_{best} + r.\left(x_{i-1}(t) - x_i(t)\right) + \beta.we_m\left(x_{best} - x_i(t)\right) i = 2, \ldots., N \end{cases}$$

Calculate out each individual's fitness $f\left(x_i(t+1)\right)$

IF $f\left(x_i(t+1)\right) < f(x_{best})$

THEN $x_{best} = x_i(t+1)$

FOR $i = 1\ to\ N$ DO

Feature positions of HDFC stock market data and weather around the best position found so far by equation (8)

Calculate each the individual's level of fitness $f\left(x_i(t+1)\right)$ and it solution is not achieved apply oppositional-based learning (OBL) by equation (9)

IF $f\left(x_i(t+1)\right) < f(x_{best})$

THEN $x_{best} = x_i(t+1)$

END FOR

END WHILE

Return x_{best}, the most significant solution founding so far.

3.5. Improved Dung Beetle Optimization (IDBO)

In the behaviours of dung beetles that inspired the IDBO algorithm were rolling, dancing, foraging, stealing, and reproduction. Four methods for reviving the population, derived from these behaviours, are developed to optimize feature selection from the HDFC bank stock dataset and Weather dataset.

Ball-Rolling Dung bettles: Equations (10-11) give a mathematical model of dung beetle behavior, which involves constant positional modifications in response to environmental conditions such as wind direction or the sunlight.

$$x_i(t+1) = x_i(t) + \alpha \times k \times x_i(t-1) + b \times \Delta x \quad (10)$$

$$\Delta x = |x_i(t) - X^w| \quad (11)$$

Iteration number t, the tth repetition is where the ith dung beetle, and if the dung beetle's direction deviates are represented by $x_i(t)$ and α, respectively. Based on the probability, its value is either set a value of 1 or -1, where the difference is denoted by −1 and no deviation by 1. The coefficient of deflection is k ∈ (0, 0.2], it shows the world's worst position. X^w, where Δx simulates the changes in lighting level, and one of the constants in (0,1) is b. There is a chance that rolling dung beetles may change their path by performing when they come against an obstacle that restricts them from proceeding. Equation (12) provides a mathematical definition of dance behavior.

$$x_i(t+1) = x_i(t) + tan(\theta)|x_i(t) - x_i(t-1)| \quad (12)$$

Values of θ are π/2 and π, the dung beetles' will be no update to the position where $\theta \in [0, \pi]$.

Breeding Dung Beetles: To make sure their offspring are protected, dung beetles choose secure areas for reproduction. This safe spawning zone is represented by an approach to boundary selection, which is detailed in Equations (13-16),

$$R = 1 - t/T_{max} \quad (13)$$

$$Lb^* = max(X^* \times (1-R), Lb) \quad (14)$$

$$Ub^* = min(X^* \times (1+R), Ub) \quad (15)$$

The borders among the lower and larger spawning areas are denoted by Lb^* and $Ub^*$for optimal feature selection, T_{max}, X is the local optimal position and specifies the maximum number of iterations, R the convergence factor, Lb and Ub the objective function's lower and upper boundaries. The outermost large circle symbolizes the optimization issue's top and bottom borders, while the inside circle depicts where dung beetles breed (Figure 5). The breeding balls' locations are shown by the red dots, these blue dots show where the rolling dung beetles are positioned, the borders are marked by the yellow dots, and X is represented by the black ball. With each iteration for the best feature selection, in the designated nesting region, one egg is produced by each female dung beetle.

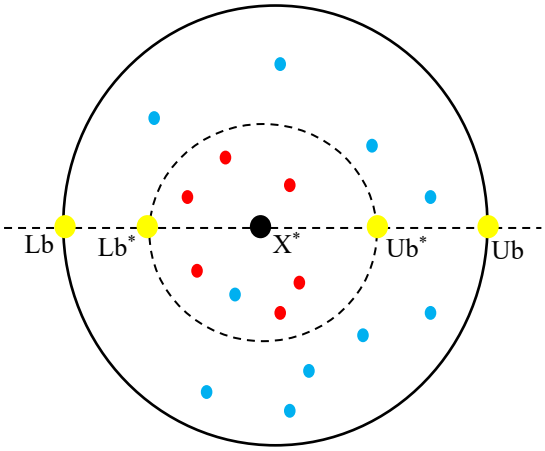

Figure 5. Strategy for Boundary Selection

The spawning region evolves dynamically to R, according to Equation (13), causing the egg-laying positions to shift accordingly. The exact formula for determining equation (16) provides the spawning the position.,

$$B_i(t+1) = X^* + b_1 \times (B_i(t) - Lb^*) + b_2 \times (B_i(t) - Ub^*) \quad (16)$$

The ith breeding ball's position at the t[th] iteration is indicated by $B_i(t)$, the independent random vectors b_1 and b_2 stand for 1× D, the symbol "×" denotes element-wise multiplication between two vectors, whereas D symbolizes the optimization problem's dimensionality. To ensure efficient feature selection, the breeding ball's position is restricted to the spawning area.

Foraging Dung Beetles: The young dung beetles must also recognize when foraging the optimal area that will direct their foraging efforts, as specified in Equations (17–18).

$$Lb^b = max(X^b \times (1-R), Lb) \quad (17)$$

$$Ub^b = min(X^b \times (1+R), Ub) \quad (18)$$

where X^b indicates the most desirable position in the globe for the selection of features, and the $Lb^b$and Ub^bsub-tables represent the top and lower limits of the optimum foraging area. Equation (19) defines the small dung beetle's location update,

$$x_i(t+1) = x_i(t) + C_1 \times (x_i(t) - Lb^b) + C_2 \times (x_i(t) - Ub^b) \quad (19)$$

$C_1, C_2 \in (0,1)$is denoted as the random number.

Stealing Dung Beetles: Dung balls from individuals are being used by dung beetles to get the best feature selection is reflected in the behavior. Throughout the iterative process, the method for updating a beetle's position information is specified by Equation (20),

$$x_i(t+1) = X^b + S \times g \times (|x_i(t) - X^*|) + |x_i(t) - X^b| \quad (20)$$

The random vector g has a normal distribution and a size of S, which is a constant.

A new global search strategy is proposed, which reduces the dependence on algorithm parameters and enhances the ability to search worldwide. Smaller dung beetles' foraging activity while the larger ones are foraging is perturbed with a t-distribution, balancing exploration and exploitation, thereby improving the convergence speed.

The position of one of the excrement balls is randomly detected using a new global exploration approach, and it follows a roll for the best feature selection. The following is the formulation of the first stage's new global exploration strategy in equation (21):

$$x_{i,j}^{p1} = \left(1 - r_{i,j}\right).x_{i,j} + r_{i,j}.\left(SF_{i,j} - I_{i,j}.x_{i,j}\right) \quad (21)$$

whereas $x_{i,j}^{p1}$ the i[th] current iteration's feature position vector for each dung beetle individual is shown. $x_{i,j}$ in the preceding iteration, depicts the specific dung beetle's feature position vector. The symbol for a random number between 0 and 1 is $r_{i,j}$. $SF_{i,j}$ represents a better feature selected from if there is no solution, the present population based on the fitness value is considered to be higher than the present person, $SF_{i,j}$is set to the global optimum. $I_{i,j}$is a randomly generated integer (1 or 2) that randomly decides whether to move closer (if $I_{i,j}$is 1) or farther (if $I_{i,j}$is 2) toward $SF_{i,j}$ during the update process. In the initial stage, a novel global exploration strategy supplants the position update formula used in the rolling step of the original dung beetle method. This strategy introduces randomness and enhanced global search capabilities into the DBO, enabling the simulated dung beetles to update their feature positions not solely based on the poorest feature solution, but also to explore other potentially superior locations at random.

Adaptive t-distribution is an improved probability distribution is used to enhance exploration and convergence performance [25]. The t-distribution mutation perturbation alters the dung beetle's foraging behavior, where the iteration number variation formula acts as a flexible parameter for the t-distribution. DBO improves via early global development followed by local investigation. Such a strategy accelerates the algorithm's convergence rate. The detailed method for location updating is outlined below the equation (22-23),

$$X_{new}^{j} = X_{best}^{j} + t(C_{iter}).X_{best}^{j} \quad (22)$$

$$C_{iter} = \frac{1}{exp\left(-4 \times \left(\frac{t}{M}\right)^{2}\right)} \quad (23)$$

The X_{new}^{j} in this iteration, the ith dung beetle individual's location vector was indicated by j. The global best solution discovered during the current iteration phase was indicated by the value X_{best}^{j}. An adaptive t-distribution's parameter is C_{iter}, and its random number is $t(C_{iter})$. M is a symbol for the maximum number of iterations. The adaptive t-distribution modulates the randomness in the dung beetle's movement based on the current iteration count, a key factor in augmenting the exploration and convergence potential of the DBO. This adaptive technique harmonizes the algorithm's exploratory and developmental capacities, accelerates its convergence rate, and enhances both its efficiency and effectiveness. The IDBO algorithm's general procedure is explained in Algorithm 2.

Algorithm 2. Improved Dung Beetle Optimization (IDBO) algorithm

Input: The maximum iteration T_{max}, the population size N

Output: Optimal position X^{b} and its fitness value

1.Initialize the population i← 1, 2, 3, ...N
2.**While** t ≤ T_{max} **do**
3.**For** i= 1, 2, ..., N **do**
4.**If** i== Ball-Rolling Dung Beetles **Then**
δ =rand (1);
5. **If** δ <0.9 **Then**
Ball-Rolling Dung Beetles are updated by equations (10-11)
6. **Else**
Ball-Rolling Dung Beetles are updated by Equation (21)
7.**End**
8. **End**
9. R = 1 − t/T_{max};
10. **If** i== Breeding Dung Beetles **Then**
Breeding Dung Beetles are updated by Equations (13-16)
11. **End**
12.**If** i== Foraging Dung Beetles **Then**
Foraging Dung Beetles are updated by Equations (17-19)
13.**End**
14. **If** i== Stealing Dung Beetles **Then**
Equation (20) updates beetles and changes their foraging behavior via t-distribution variation Equations (22-23)
15. **End**
16. **End**
17. **End**

3.6. Modified Donkey and Smuggler Optimization (MDSO)

MDSO draws inspiration from the way donkeys search to effectively select optimal features from a dataset. This algorithm simulates real-world transportation behaviours, such as how donkeys explore and choose movement paths. It employs two distinct modes to carry out the search process and route selection, obtaining to extract the most valuable characteristics from the weather and HDFC bank stock datasets. The two modes involved are Smuggler and Donkeys. In the Smuggler mode, all potential features are explored to identify the most suitable ones for optimal feature selection from the HDFC bank stock and weather datasets. Run, Face & Suicide, and Face & Support are only a few of the donkey-inspired behaviours used in the Donkeys mode for selecting features [26] , [27] that represented in figure 6.

Part I: Smuggler (Non-Adaptive): The Non-Adaptive component of the algorithm remains unchanged in response to variations. In this phase, the smuggler evaluates all possible paths (features) from the source

to the destination. Considering certain standards like prediction, accuracy, and error rate performance of each path, the optimal route is selected, and the donkey proceeds along that chosen path. In the Smuggler phase, each solution is assessed, and its fitness is determined according to its accuracy. The feature solutions are then grouped according to their fitness scores. The most optimal solution is selected, and the donkey is directed to follow that solution.

Part II: Donkey (Adaptive): The present status of the network provides the decisions made in adaptive routing and the position of features in the dataset. The Donkey's activities will serve as the basis for this response in the ways listed below,

1. **Run:** Switch to an alternative optimal path (i.e., feature set). If the features identified as best during the non-adaptive phase are no longer optimal, they are discarded and replaced with a new best feature solution based on the updated information.
2. **Face and Suicide**: The selection route for identifying the top feature from the stock and weather datasets remains fixed. At the same time, if the current path (feature) becomes blocked, it's discarded in favor of the next best feature for stock market prediction and no fitness re-evaluation is performed for the remaining candidates. If modifications affect the algorithm's accuracy and the optimum solution determined in the initial stage is no longer the best, and there is a desire to retain that path, the current feature solution is discarded. It will be reintroduced once it returns to its optimal state and the second-best feature solution is identified within the set of solutions.
3. **Face and Support**: If the smuggler fails to find a solution, instead of discarding it, assign the second-best feature as a replacement.

Algorithm 3 demonstrates the general procedure of proposed Modified Donkey and Smuggler Optimization (MDSO).

Algorithm 3. MDSO Algorithm
Input: Number of features f in the dataset D
Output: optimal selected features from the D

Smuggler Part

1. **Begin**
2. An initial population of solutions should be generated at random
3. For row=1 to n_1
4. For col=1 to n_2
5. parameters (row, col) =$rand_i$ ([decision variable ranges], n_1, n_2)
6. End for
7. End for
8. For (e=1 to n_1) do
9. Assess each solution's accuracy to determine its fitness.
10. Update the population of the possible feature for stock market prediction
11. End for
12. Set the best features for each dataset
13. Display the best selected features to SMP
14. Provide the donkey portion the best feature solution that was selected.
15. **End**

Donkey Part

1. **Begin**
2. Determine whether or not there was a change in fitness.
3. If the optimal feature solution's fitness develops less
4. Update the optimal solution.

$$Best_{suicid\ solution} = f(x_i) - f(cbest_{solution}) \quad (24)$$

$f(cbest_{solution})$ denotes the crossover best results of features based on the dataset and SMP. Crossover create a new feature solution by selecting parameters or features from two parent solutions the equation (24)

5. The optimum feature solution is determined by Face & Suicide to be the one with the solution population's second-best feature fitness.
6. Face & Support utilizes the population's second-best feature fitness to support the highest feature solution, meaning both are employed for the same objective.

No updates are made to the population's fitness (as indicated by Equations (25-26)).

$$second\ best_{solution} = f(cbest_{solution}) - f(x_i) \quad (25)$$

$$best_{support\ solution} = cbest_{solution} + second\ best_{solution} \quad (26)$$

7. End If
8. End

Figure 5.9 shows the general flowchart of the suggested MDSO method.

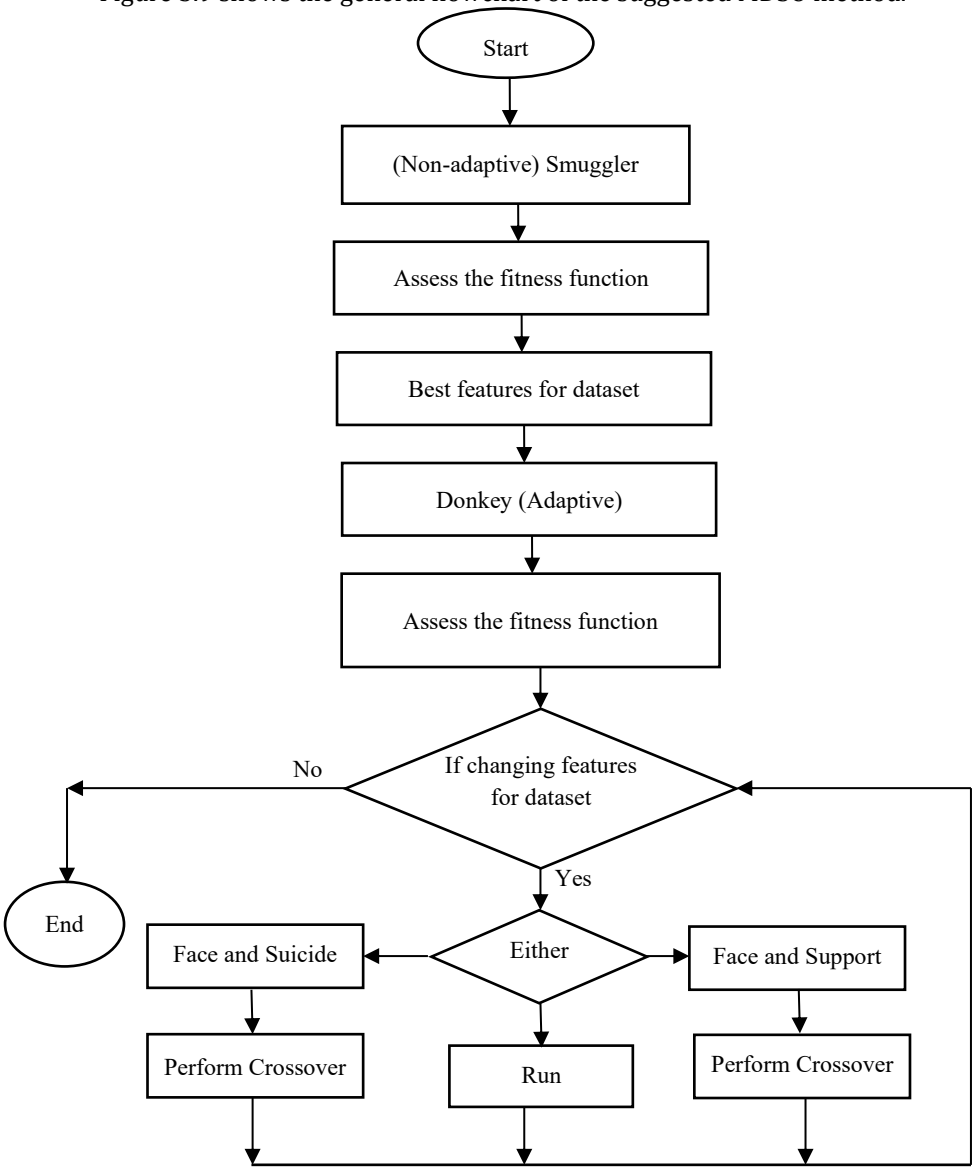

Figure. 6. Modified Donkey and Smuggler Optimization (MDSO) Algorithm

3.6. Mutual Information (MI) approach

Selecting an appropriate subset of relevant features F^' where (i) F^'⊆F and (ii) a classifier produces an optimal prediction accuracy is the issue for a certain dataset D with a feature of dimension d set F={f_1,f_2,…,f_d }. For each pair $(f_i, f_j) \in F'$, a subset of features with the feature-class mutual information with the highest value and the lowest. Mutual Information (MI) is used for selecting a set of relevant qualities. The best feature set is determined by analyzing features and classes connect information. MI measures how dependent two variables are on one another, indicating how they are correlated. The mutual information $MI\ (V, Y)$ represents the combined probability distribution's Kullback–Leibler (KL) divergence $p\ (x, y)$ and the distributions of marginal probabilities $p(x)$ and $y(y)$, assuming that two random variables' combined probability distribution, X and y, is $p\ (x, y)$, and the probability of marginal distributions are $y(y)$ and $y(y)$ [28] in the equation (27),

$$MI(X,Y) = -\int_x \int_y f_{x,y}(x,y) log \frac{f_{x,y}(x,y)}{f_x(x) f_y(y)} \quad (27)$$

According to the definition above, when variables are entirely independent, their mutual information is at its lowest, resulting in a value of 0. A higher mutual information between features indicates a stronger correlation between them. In data processing, the target variable should be selected with characteristics that have a high mutual information to enhance the algorithm's predictive performance, while eliminating features with low mutual information to minimize data redundancy. This strategy is referred to as feature selection based on most redundancy and greatest correlation [29]. The proposed model also incorporates TextBlob, VADER, and RoBERTa for sentiment analysis.

3.7. Stock market prediction using Ensemble Deep Learning (EDL)

EDL classification is a method for machine learning that integrates many models, such as EBi-LSTM, Entropy Denoising Autoencoder (EDAE), and Entropy Generative Adversarial Network (EGAN) to improve system performance as a whole. To construct a more accurate and robust model, the idea is to train many base models and then combine their predictions. EDL was used for SMP analysis of stock, COVID, and weather data. The network generates SMP via SA and regular dataset proportions, while the inputs include weather, stock, and COVID data.

3.8. EBi-LSTM

Long-term dependencies within time steps and data sequences were the initial purpose of LSTM's construction. In the Bi-LSTM classifier, LSTM layers for forward and backward data are used. LFFSSO uses the social spider's cooperative behavior to maximize classifier parameter selections.

3.9. Entropy Denoising Autoencoder (EDAE)

A neural network that can recreate its input is known as an autoencoder (AE). It has an encoder and decoder. Encoders reduce input data's dimension, and the original input is then reconstructed by the decoder using the compressed data. For an input dataset X, the reconstructed signal X_{deco} is calculated using Equation (28-29),

$$X_{enco} = Net_e(X) \quad (28)$$

$$X_{deco} = Net_d(X_{enco}) \quad (29)$$

where Net_e and Net_d indicate the establishment of the encoder and decoder networks, and X_{enco} indicates the encode of the dataset X. A DAE is used in this work to increase the AE's resilience. In this approach, the input dataset X is stochastically reduced in dimensionality. The reduced dataset is then used as input, while the original training objective is the dataset autoencoder. The model is intended to capture more consistent connects between the dataset and the expected output to recreate the original dataset from the reduced-dimensional dataset. As a result, to prevent the problem of overfitting, the DAE is probably more accurate than the easy AE [30].

According to [31], the DAE creates a reduced-dimensionality dataset of the input samples after learning the usual behavior from data under normal settings. The reconstruction error (RE), an important status indicator for SMP, is provided by the AE along with the reconstructed version of the instance when a new instance is introduced in the equation (30).

$$RE = \frac{1}{n} \sum_i \left(x^{(i)} - x_{deco}^{(i)}\right)^2 \quad (30)$$

The elements of X and X_{deco} are indicated by $x^{(i)}$ and $x_{deco}^{(i)}$, respectively. Let the number of items in X be denoted by n. The SMP model uses the output of the AE as training data once it has been constructed and trained using SMP data. The status indicator that is used is the reconstruction error (RE). The difference between the model's output and the real labels is evaluated by the loss function [32]. Optimizing model parameters by reducing the model fits training data effectively because of the loss function data and performance on new, untested data. A major factor in the performance and training results from the model include its selection of loss function.

The loss function for cross-entropy contrasts the model's anticipated probability distribution to the actual probability distribution. Equation (31) presents the cross-entropy loss function's mathematical formulation considering multi-class categorization difficulties,

$$L = \frac{1}{N} \sum_i L_i = -\frac{1}{N} \sum_i \sum_{c=1}^{M} w_c \times y_{ic} log(p_{ic}) \quad (31)$$

In the multiclass classification, w_c is the weight given to the c^{th} class. The following is the equation (32) for computation.,

$$w_c = bias + (1 - bias).e^{-\alpha.size_c} \quad (32)$$

where, $bias$ represents a bias term that prevents high-frequency class weights from dropping to zero periodically. The technique used to calculate the parameter α is outlined as follows the equation (33),

$$\alpha = \frac{bias}{size} \quad (33)$$

where bas represents the function's decay rate, which affects the curve's form, and $jize$ represents the dataset's total number of samples.

3.10. Entropy Generative Adversarial Network (EGAN)

Game theory ideas form the basis of GANs, in which the discriminator and generator engage in competition with one another throughout training in an effort to achieve a Nash equilibrium. While the discriminator (D) concentrates on precisely distinguishing between produced and actual data, the generator (G) simulates data dispersion. A predicted sample G(z), a multi-dimensional vector is generated by converting a random noise vector z into a new data space. Both produced samples from the generator G and actual samples from the dataset enter the discriminator D, which performs the function of a binary classifier. The chance that a given sample is genuine rather than produced is indicated by its output [33], [34]. When discriminator D is unable to distinguish data created by the generator from data that is actual, the ideal state is reached. At this stage, the distribution of the actual data has been correctly learnt by the generator model G.

The generator and discriminator are two opposing individuals in the structure of game theory, each with its own loss function, J(D) and J(G) are the corresponding symbols. According to [35], utilizing cross-entropy, the discriminator D's loss function is defined as a binary classifier in the equation (34).

$$J^{(D)} = -\frac{1}{2}\mathbb{E}_{x\sim pdata}\, log\, D(x) - \frac{1}{2}\mathbb{E}_z\, log(1 - D\big(G(z)\big) \quad (34)$$

Here, x denotes a real sample, G(z) represents the generator's output, while z stands for a random noise vector. E is an initials for expectancy. D(x) is the probability that x will be categorized as actual data by the discriminator, while D(G(z)) is a possibility that G's data will be recognized as authentic by the discriminator. The generator G wants to make this value approach 1, while the discriminator D strives to precisely identify the data's origin by aiming for D(G(z)) to be near 0. A zero-sum game results from the conflict between the two models caused by this conflicting objective. Consequently, the output of the discriminator may be used to define the generator's loss [36] in the equation (35),

$$J^{(G)} = -J^{(D)} \quad (35)$$

As a result, the optimization challenge in GANs is reformulated as a minimax game, as illustrated below the equation (36),

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim pdata}[log\, D(x)] + \mathbb{E}_{z\sim p(z)}\left[log\left(1 - D\big(G(z)\big)\right)\right] \quad (36)$$

During training, the parameters of G are adjusted in tandem with those of D. When D(G(z)) reaches 0.5, it shows that the produced and actual distributions are unable to distinguished by the discriminator. At this point, the model attains its global optimal solution.

3.11. Ensemble bagging classification

Ensemble learning involves combining multiple training datasets by considering various subsets. Each data subset is assigned to a base classifier, which is then integrated into a new ensemble classifier. It uses the ensemble bagging approach to create the training sets while dealing with incremental data. The construction of an ensemble classifier involves analyzing how classifiers reflect emerging information to create different datasets from the sample gathering. A majority vote approach is used to build and assess a classifier based on the subset of learning. This method facilitates the gradual use of an ensemble learning algorithm.

The aggregator (S) obtains self-sampling using T^{th} round aggregator sample S. To extract a subset from the aggregator T^{th} each sample subset, given an aggregator value S in the interval (t=1, 2, T), S_k incorporates Nth sample. New learning training sample in base classifier is S_t utilizing the bagging technique. All subsets' weak classifiers $S_t$denoted as $\phi(x, S_k)$ Equation (37) exhibits the calculation of the weak classifier error rate ϕ (x, S_k) in addition,

$$\varepsilon_I = \sum_{(x_t,y_t)} [\phi_i(x, S_k) \neq y_i]/|S_I| \quad (37)$$

Estimating the tth weak classifier and $\phi(x, S)$ as an effective classifier, we extract the training set distribution S_I. Strong processing results in the final conclusion classifier $\phi(x, S)$ to accomplish the tth weakly classification value objective $\phi(x, S_k)$.

4. RESULTS AND DISCUSSION

In this section, experimentation results are compared between the proposed classifier, and existing classifiers. Twitter and Kaggle were used to gather the Housing Development Finance Corporation (HDFC) dataset, which includes COVID-19, stocks, weather, and Twitter. EBi-LSTM, EDAE, EGAN, and EDL with the HDFC stock and complete dataset results are discussed in the table 1. The proposed classifier's precision, recall, F1-score, and accuracy in the HDFC stock dataset are 93.41%, 94.82%, 94.10%, and 94.67%, respectively. In the entire dataset, the suggested classifier produces a higher F1-score, recall, and precision values of 94.83%, 95.67%, 95.25%, and 96.86%.

Table 1. Comparing Classifier with Dataset Results

Dataset	Classifiers	Precision (%)	Recall (%)	F1-Score (%)	Accuracy (%)
HDFC stock	EBi-LSTM	88.26	90.43	89.33	92.63
	EDAE	91.34	91.56	91.44	93.05
	EGAN	92.18	92.34	92.26	93.52
	EDL	93.41	94.82	94.10	94.67
Stock+ Weather + COVID 19 +Twitter	EBi-LSTM	90.44	93.18	91.78	94.36
	EDAE	92.53	93.75	93.13	93.85
	EGAN	93.68	94.45	94.06	94.93
	EDL	94.83	95.67	95.25	96.86

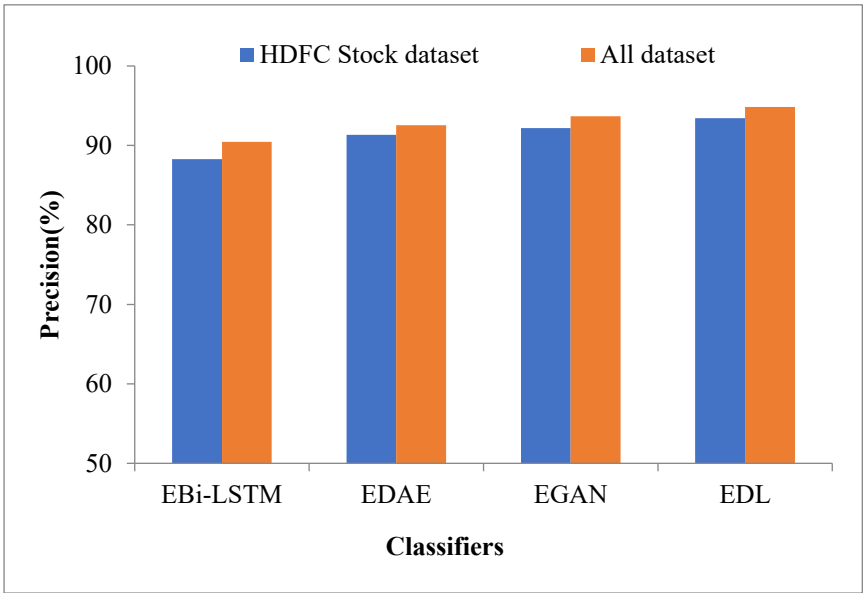

Figure 7 Precision Results Comparison Vs. Datasets

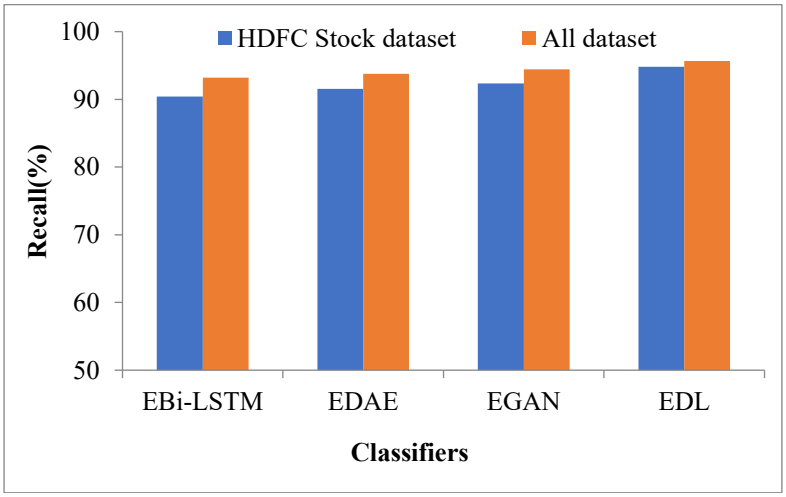

Figure 8. Recall Results Comparison Vs. Datasets

Comparison of classifiers with precision, including EBi-LSTM, EDAE, and EGAN, and EDL to HDFC stock and the complete dataset with regard to accuracy is shown in Figure 7. The suggested methodology provides a better accuracy value of 94.83% for the whole dataset, in contrast to alternative approaches such EBi-LSTM, EDAE, and EGAN give precision values of 90.44%, 92.53%, and 93.68% respectively. EBi-LSTM, EDAE, EGAN, and EDL concerning recall among HDFC stock and the complete dataset are illustrated in Figure 8. The suggested approach has a higher recall rate of 95.67% than alternative methods, like as EBi-LSTM, EDAE, and EGAN are 93.18%, 93.75%, and 94.45% respectively.

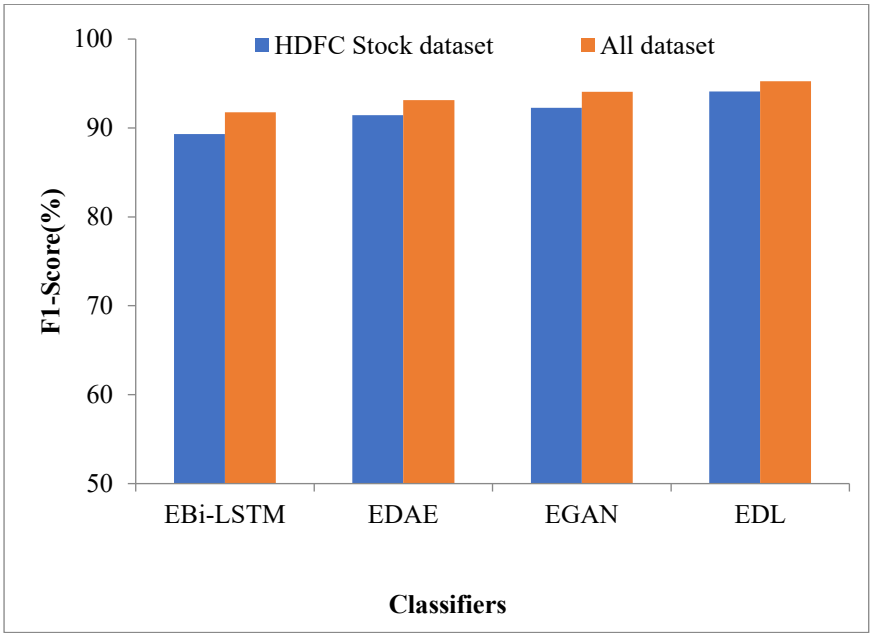

Figure 9. F1-Score Results Comparison Vs. Datasets

The HDFC stock and all datasets' F1-score comparison with regard to classifiers like EBi-LSTM, EDAE, EGAN, and EDL are illustrated in Figure 9. The suggested method has a higher F1-score of 95.25%, according to the results. EBi-LSTM, EDAE, and EGAN givea 91.78%, 93.13%, and 94.06%, respectively.

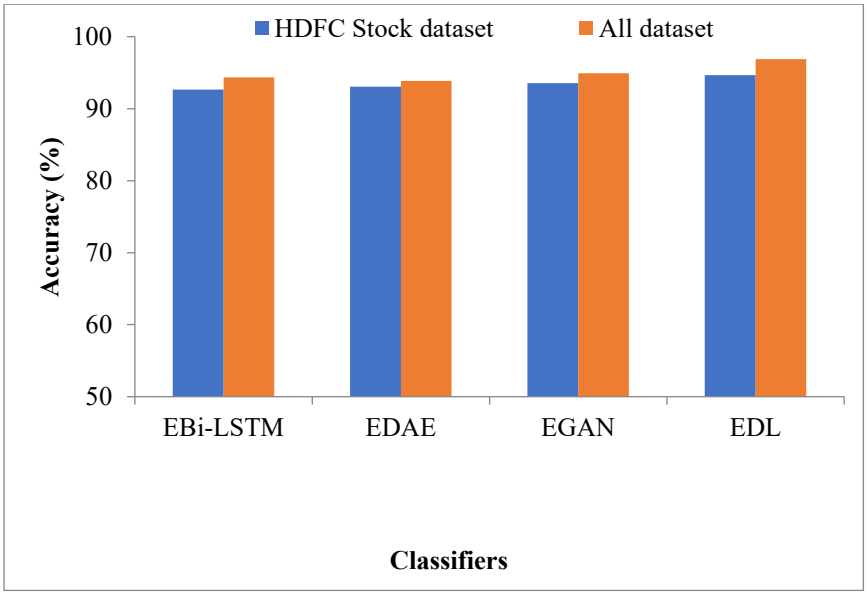

Figure 10. Accuracy Results Comparison Vs. Datasets

A precise assessment of all datasets for classifiers, including the HDFC stock, EBi-LSTM, EDAE, EGAN, and EDL is illustrated in Figure 10. The suggested method has a better accuracy rate of 96.86%, compared to alternative techniques such EBi-LSTM, EDAE, and EGAN gives an accuracy of 94.36%, 93.85%, and 94.93%.

CONCLUSION

The contribution, EOFS and EDL have been introduced for feature selection and classification of stock market prediction. EOFS is introduced that combines the procedure of several methods like AMRFO, Improved Dung Beetle Optimizer (IDBO) and Modified Donkey and Smuggler Optimization (MDSO). Mutual Information (MI) serves to integrate the results of these methods. Three foraging behaviours serve as the foundation for the AMRFO algorithm: chain foraging, cyclone foraging, and somersault foraging, aimed at optimal feature selection. Dung beetle behaviours, including as rolling, dancing, consuming, stealing, and reproduction, are the basis for the IDBO algorithm. MDSO is motivated by the searching behavior of donkeys for effective feature selection from the dataset. Additionally, a measure of how dependent two traits are on one another is known as mutual information (MI), primarily indicating their correlation. For sentiment analysis of tweet data, TextBlob, VADER, and RoBERTa are utilized. Machine learning methods like EBi-LSTM integrate many models to provide EDL classification, Entropy Denoising Autoencoder (EDAE), and Entropy Generative Adversarial Network (EGAN) to enhance the overall system performance. Ensemble bagging decision is obtained by combining the results of these classifiers. In the dataset, the suggested classifier produces better accuracy, f1-score, recall, and precision results of 94.83%, 95.67%, 95.25%, and 96.86%.

References

1. Julian, T., Devrison, T., Anora, V. and Suryaningrum, K.M., 2023. Stock price prediction model using deep learning optimization based on technical analysis indicators. Procedia Computer Science, vol. 227, pp.939-947.
2. Ruhal, R. and Prashar, E.V., 2023. A Comparative Study of Statistical Methods and Machine Learning Approaches for Stock Price Prediction. Journal of Emerging Technologies and Innovative Research (JETIR).
3. Kukreja, V., Thapliyal, N., Aeri, M. and Sharma, R., 2024. An Efficient Auto Regressive Integrated Moving Average Model for AAPL, MSFT, NTFX, and GOOGL Stock Price Prediction. In 2024 IEEE International Conference on Information Technology, Electronics and Intelligent Communication Systems (ICITEICS), pp. 1-5.
4. Chou, J.S. and Pham, T.B.Q., 2025. Leveraging ensemble learning-based stock preselection with multiobjective investment optimization for stepwise decision-supported portfolio management. Journal of Big Data, vol. 12, no. 1, pp.1-49.
5. Liu, B. and Sun, D., 2024. Stock price forecasting based on improved particle swarm optimization neural network algorithm. In Proceedings of the 2024 9th International Conference on Intelligent Information Processing, pp. 326-331.
6. Thakkar, A. and Chaudhari, K., 2024. Applicability of genetic algorithms for stock market prediction: A systematic survey of the last decade. Computer Science Review, vol. 53, p.100652.
7. Das, S., Nayak, M., Senapati, M.R. and Satapathy, J., 2021. New approaches in stock market prediction using velocity enhanced whale optimization algorithm. In 2021 19th OITS International Conference on Information Technology (OCIT), pp. 105-109.
8. Ansari, Y., Gillani, S., Bukhari, M., Lee, B., Maqsood, M. and Rho, S., 2024. A multifaceted approach to stock market trading using reinforcement learning. IEEE Access, vol. 12, pp.90041-90060.
9. Baswaraj, D., Natchadalingam, R., Rekha, R.S., Sahni, N., Divya, V. and Reddy, P.C.S., 2023. An Accurate Stock Prediction Using Ensemble Deep Learning Model. In 2023 International Conference on Research Methodologies in Knowledge Management, Artificial Intelligence and Telecommunication Engineering (RMKMATE), pp. 1-7.
10. Mehtab, S. and Sen, J., 2020. Stock price prediction using convolutional neural networks on a multivariate timeseries. arXiv preprint arXiv:2001.09769.
11. Moghar, A. and Hamiche, M., 2020. Stock market prediction using LSTM recurrent neural network. Procedia computer science, vol. 170, pp.1168-1173.
12. Chowdhury, M.S., Nabi, N., Rana, M.N.U., Shaima, M., Esa, H., Mitra, A., Mozumder, M.A.S., Liza, I.A., Sweet, M.M.R. and Naznin, R., 2024. Deep learning models for stock market forecasting: A comprehensive comparative analysis. Journal of Business and Management Studies, vol. 6, no. 2, p.95.
13. Khan, W., Ghazanfar, M.A., Azam, M.A., Karami, A., Alyoubi, K.H. and Alfakeeh, A.S., 2022. Stock market prediction using machine learning classifiers and social media, news. Journal of Ambient Intelligence and Humanized Computing, vol. 13, no. 7, pp.3433-3456.
14. Bansal, M., Goyal, A. and Choudhary, A., 2022. Stock market prediction with high accuracy using machine learning techniques. Procedia Computer Science, vol. 215, pp.247-265.
15. Ampomah, E.K., Nyame, G., Qin, Z., Addo, P.C., Gyamfi, E.O. and Gyan, M., 2021. Stock market prediction with gaussian naïve bayes machine learning algorithm. Informatica, 45(2).

16. El Mahjouby, M., Bennani, M.T., Lamrini, M., Bossoufi, B., Alghamdi, T.A. and El Far, M., 2024. Machine learning algorithms for forecasting and categorizing euro-to-dollar exchange rates. IEEE Access, vol. 12, pp.74211-74217.
17. Jagadesh, B.N., RajaSekhar Reddy, N.V., Udayaraju, P., Damera, V.K., Vatambeti, R., Jagadeesh, M.S. and Koteswararao, C., 2024. Enhanced stock market forecasting using dandelion optimization-driven 3D-CNN-GRU classification. Scientific Reports, vol. 14, no. 1, p.20908.
18. Pagliaro, A., 2023. Forecasting significant stock market price changes using machine learning: Extra trees classifier leads. Electronics, vol. 12, no. 21, p.4551.
19. Ho, T.T. and Huang, Y., 2021. Stock price movement prediction using sentiment analysis and CandleStick chart representation. Sensors, vol. 21, no. 23, p.7957.
20. Worasucheep, C., 2022. Ensemble classifier for stock trading recommendation. Applied Artificial Intelligence, vol. 36, no. 1, p.2001178.
21. BL, S. and BR, S., 2023. Combined deep learning classifiers for stock market prediction: integrating stock price and news sentiments. Kybernetes, vol. 52, no. 3, pp.748-773.
22. Muhammad, D., Ahmed, I., Naveed, K. and Bendechache, M., 2024. An explainable deep learning approach for stock market trend prediction. Heliyon, vol. 10, no. 21.
23. Gharehchopogh, FS, Ghafouri, S, Namazi, M & Arasteh, B 2024, 'Advances in manta ray foraging optimization: A comprehensive survey,' Journal of Bionic Engineering, vol.21, no.2, pp.953-990.
24. Hemeida, MG, Ibrahim, AA, Mohamed, AAA, Alkhalaf, S & El-Dine, AMB 2021, 'Optimal allocation of distributed generators DG based Manta Ray Foraging Optimization algorithm (MRFO),' Ain Shams Engineering Journal, vol.12, no.1, pp.609-619.
25. Lyu, L, Jiang, H & Yang, F 2024, 'Improved Dung Beetle Optimizer Algorithm with Multi-Strategy for global optimization and UAV 3D path planning,' IEEE Access, vol.12, pp.69240 – 69257.
26. Shamsaldin, AS, Rashid, TA, Al-Rashid Agha, RA, Al-Salihi, NK & Mohammadi, M 2019, 'Donkey and smuggler optimization algorithm: A collaborative working approach to path finding,' Journal of Computational Design and Engineering, vol.6, no.4, pp.562-583.
27. Mohapatra, S, Panda, M, Sarangi, PP, Mishra, BSP, Parihar, N & Sahoo, B 2024, 'MLP-FOX: A Metaheuristic Approach-Based Training Multilayer Perceptron for Data Classification,' In 2024 1st International Conference on Cognitive, Green and Ubiquitous Computing (IC-CGU), pp. 1-6.
28. Huang, D, Liu, Z & Wu, D 2023, 'Research on Ensemble Learning-Based Feature Selection Method for Time-Series Prediction,' Applied Sciences, vol.14, no.1, pp.1-16.
29. Zhou, H, Wang, X & Zhu, R2022, 'Feature selection based on mutual information with correlation coefficient,' Applied intelligence, vol.52, no.5, pp.5457-5474.
30. Jia, X, Han, Y, Li, Y, Sang, Y & Zhang, G 2021, 'Condition monitoring and performance forecasting of wind turbines based on denoising autoencoder and novel convolutional neural networks,' Energy Reports, vol.7, pp.6354-6365.
31. Liu, X, Zhang, H, Niu, Y, Zeng, D, Liu, J, Kong, X & Lee, K.Y 2020, 'Modeling of an ultra-supercritical boiler-turbine system with stacked denoising auto-encoder and long short-term memory network,' Information Sciences, vol.525, pp.134-152.
32. Song, Z, Shi, Z, Yan, X, Zhang, B, Song, S & Tang, C 2024, 'An Improved Weighted Cross-Entropy-Based Convolutional Neural Network for Auxiliary Diagnosis of Pneumonia,' Electronics (2079-9292), vol.13, no.15, pp.1-15
33. Creswell, A, White, T, Dumoulin, V, Arulkumaran, K, Sengupta, B & Bharath, AA 2018, 'Generative adversarial networks: An overview,' IEEE signal processing magazine, vol.35, no.1, pp.53-65.
34. Aggarwal, A, Mittal, M & Battineni, G 2021, 'Generative adversarial network: An overview of theory and applications,' International Journal of Information Management Data Insights, vol.1, no.1, p.100004.
35. Gui, J, Sun, Z, Wen, Y, Tao, D & Ye, J 2021, 'A review on generative adversarial networks: Algorithms, theory, and applications,' IEEE transactions on knowledge and data engineering, vol.35, no.4, pp.3313-3332.
36. Jabbar, A, Li, X & Omar, B 2021, 'A survey on generative adversarial networks: Variants, applications, and training,' ACM Computing Surveys (CSUR), vol.54, no.8, pp.1-49.