Journal of Neonatal Surgery
ISSN(Online): 2226-0439
Vol. 14, Issue 32s (2025)
https://www.jneonatalsurg.com

OPEN ACCESS

Conditional Generative Adversarial Network (CGAN) Based Data Oversampling And Gaussian Bobcat Optimization Algorithm (GBOA) For Heart Diseases Prediction

M.MadhanGiri[1], Dr. M.Santhalakshmi[2]

[1]Research Scholar, Kamban College of arts and Science, Sulthanpet.

[2]Assistant Professor of Computer Science, Kamban College of arts and Science, Sulthanpet.

.Cite this paper as: M.MadhanGiri, Dr. M.Santhalakshmi, (2025) Conditional Generative Adversarial Network (CGAN) Based Data Oversampling And Gaussian Bobcat Optimization Algorithm (GBOA) For Heart Diseases Prediction. *Journal of Neonatal Surgery*, 14 (32s), 6650-6666.

ABSTRACT

Early detection of cardiac problems and continuous patient monitoring by physicians can help reduce death rates. The classification of imbalanced datasets is an important task in machine learning. The number of samples in each class is not uniformly distributed; one class contains a large number of samples while the other has a small number. This often leads to higher classification accuracy for the majority category and lower classification accuracy for the minority category. In this paper, Conditional Generative Adversarial Network (CGAN) is a type of neural network to generate new data samples similar to a given training sample. CGAN is introduced to translate unbalanced samples from one form to another resulting in balanced samples. Gaussian Bobcat Optimization Algorithm (GBOA) is introduced for the importance of features and provides valuable insights into the relevance and predictive power of each feature in a heart disease dataset. GBOA is derived from the hunting strategy of bobcats during the attack towards the prey and the chase process between them. Natural behaviors of the bobcat in the wild, the strategy of this animal during hunting is much more prominent. Memory-Augmented Deep Autoencoder (MADAE) classifier consists of a fully connected three-layer neural network where the encoder contains input and hidden layers and the decoder part comprises hidden and output layers. By allowing early detection and treatment, accurate heart disease prediction can greatly reduce mortality. Cleveland Heart Disease Database (CHDD) is collected from University of California Irvine (UCI) repository. Experimental results show the effectiveness of the classifiers in terms of Precision, Recall, F-Measure, and Accuracy.

Keywords: *Health care system, Heart disease, Machine learning, Conditional Generative Adversarial Network (CGAN), Gaussian Bobcat Optimization Algorithm (GBOA), Memory-Augmented Deep Autoencoder (MADAE), Cleveland Heart Disease Database (CHDD), and Swarm Intelligence (SI).*

1. INTRODUCTION

The healthcare industry generates a lot of data about patients, illnesses, and diagnoses, but it isn't being used correctly to produce the desired results. Heart disease and stroke are two of the main causes of death. According to a World Health Report (WHO) by 2030, the total number of deaths due to Cardiovascular Disease (CVD) will increase to 23.6 million, mainly from heart disease and stroke. Age, sex, smoking, family history, cholesterol, poor diet, high blood pressure, obesity, physical inactivity, and alcohol intake are considered to be risk factors for heart disease, and hereditary risk factors such as high blood pressure and diabetes also lead to heart disease. Some risk factors are controllable. Apart from the above factors, lifestyle habits such as eating habits, physical inactivity, and obesity are also considered to be major risk factors. There are different types of heart diseases such as coronary heart disease, angina pectoris, congestive heart failure, cardiomyopathy, congenital heart disease, arrhythmias, and myocarditis. It is difficult to manually determine the odds of getting heart disease based on risk factors [1-3].

Due to the recent technological advancements, the field of medical sciences has seen a remarkable improvement over time [3], [4]. Machine Learning (ML) has been widely used in the field of cardiovascular medicine and it has established a potential space [5]. The basic framework of ML is built on models that take input data, and through the usage of some statistical analysis and mathematical optimizations provides the desired prediction results [6]. Early-stage detection of CVD is an important way of reducing this toll. These techniques that relate allow hidden knowledge to be extracted and to identify relationships among attributes within the dataset, and are a promising strategy for CVD classification [5-6].

Various techniques in data mining and ML have been employed to find out the severity of heart disease among humans. The perspective of medical science and data mining are used for discovering various sorts of metabolic syndromes. Data mining with classification plays a significant role in the prediction of heart disease and data investigation. Hence, early and accurate diagnosis and the provision of appropriate treatments are keys to reducing the amount of death causing CVD diseases. However, the inclusion of the large number of data features is not required for reasons related to the curse of dimensionality [7-8].

Mostly, medical datasets cover related as well as redundant features. Unnecessary features don't contribute any meaningful information to the prediction task, and also creates noise in the description of target (output class) which leads to prediction errors [9]. Furthermore, such features increase the complexity of ML models and make the system run slowly due to increased training time. To overcome the curse of dimensionality only those features which are closely related with the target should be selected/identified from datasets and provided as inputs to ML models [10]. Relevant feature selection can aid in performance improvement by decreasing the model complexity and increasing prediction accuracy which is very important in medical diagnosis [10].

To overcome problems arising from the high-dimensionality of data, researchers use mainly two approaches. The first one is feature extraction, which comprises the creation of a new feature space with low dimensionality. The second one is feature selection that focuses primarily on removal of irrelevant and redundant features of the original feature set and, therefore, the selection of a small subset of relevant features. The feature selection problem with n features has 2n possible solutions. Feature Selection (FS) is tackled more and more with Swarm Intelligence (SI) algorithms [11,12], because SI has been proved as a technique which can solve NP-hard computational problems [13] and finding an optimal feature subset is certainly this kind of a problem. Despite the proven usage of SI algorithms for FS [13-14], a narrowed search for surveys in this field gives few results. They focused only on six bio-inspired swarm algorithms, i.e., Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), Artificial Fish Swarm Algorithm (AFSA), Artificial Bee Colony Algorithm (ABC), Firefly Algorithm (FA), and Bat Algorithm (BA).

In data mining classification, class imbalance is characterized by the fact that different classes have an obvious difference in the number of samples. Most classifiers typically assume a balanced class distribution or assign equal classification error costs to different classes. Therefore, directly using an imbalanced class will worsen the classification performance [15-16]. The oversampling algorithms can achieve the balance by synthesizing new samples, but the uncontrollable positions of the synthetic samples may aggravate the data overlap and further deteriorate the classification performance [17]. Data-level techniques attempt to reduce the level of imbalance through various data sampling methods. Algorithm-level methods for handling class imbalance, commonly implemented with a weight or cost schema, include modifying the underlying learner or its output in order to reduce bias towards the majority group. Finally, hybrid systems strategically combine both sampling and algorithmic methods.

Despite recent advances in deep learning, it is introduced with its increasing popularity in class imbalance [18-19]. Despite these advances, very small statistical analysis which properly evaluates techniques for handling class imbalance using deep learning and their corresponding architectures [20-21]. The performance of computer vision algorithms can significantly deteriorate when the training dataset is imbalanced. In recent years, Generative Adversarial Network (GAN) has gained immense attention by researchers across a variety of application domains due to their capability to model complex real-world data. It is also a fascinating adversarial learning idea that showed good potential in restoring balance in imbalanced datasets [22-23].

In this work, Conditional Generative Adversarial Network (CGAN) is introduced to handle the class disproportion issue, which generates adequate trials for minority classes. CGAN technique is introduced which percolates the information to guarantee only generating minority classes to improve diagnosis accuracy. Gaussian Bobcat Optimization Algorithm (GBOA) is introduced for the importance of features and provides valuable insights into the relevance and predictive power of each feature in a heart disease dataset. Memory-Augmented Deep Autoencoder (MADAE) model is introduced which consists of an encoder, a decoder, and a memory, and the purpose of the memory is to learn prototypical patterns for the Cleveland dataset. Experimental results show the effectiveness of the classifiers in terms of Precision, Recall, F-Measure, and Accuracy.

2. LITERATURE REVIEW

Ghosh et al., [24] proposed a model with efficient Data Collection, Data Pre-processing and Data Transformation methods to create accurate information for the training model. Dataset is collected from Cleveland, Long Beach VA, Switzerland, Hungarian and Stat log. Suitable features are selected by using the Relief, and Least Absolute Shrinkage and Selection Operator (LASSO) techniques. New hybrid classifiers like Decision Tree Bagging Method (DTBM), Random Forest Bagging Method (RFBM), K Nearest Neighbor Bagging Method (KNNBM), AdaBoost Boosting Method (ABBM), and GradientBoost Boosting Method (GBBM) are developed by integrating the traditional classifiers with bagging and boosting methods, which are used in the training process. Classifiers are measured using metrics like Accuracy (ACC), Sensitivity (SEN), Error Rate, Precision (PRE) and F1 Score (F1) of proposed model, along with the Negative Predictive Value (NPR), False Positive Rate (FPR), and False Negative Rate (FNR).

Li et al., [25] proposed an efficient and accurate system to diagnose heart disease and the system is based on ML techniques. The system is developed based on classification algorithms includes Support Vector Machine (SVM), Logistic Regression

(LR), Artificial Neural Network (ANN), KNN, Naïve Bayes (NB), and Decision Tree (DT) while standard features selection algorithms have been used such as Relief, Minimal redundancy maximal relevance, LASSO and Local learning for removing irrelevant and redundant features. Fast conditional mutual information (FCMIM) algorithm is introduced to solve the feature selection problem. The features selection algorithms are used for features selection to increase the classification accuracy and reduce the execution time of the classification system. FCMIM algorithm is feasible with a SVM classifier for designing a high-level intelligent system to identify heart disease. The proposed diagnosis system (FCMIM-SVM) achieved good accuracy as compared to recently introduced methods.

Ishaq et al., [26] proposed to find significant features and effective data mining techniques that can boost the accuracy of cardiovascular patient’s survivor prediction. To predict patient’s survival, this study employs nine classification models: DT, Adaptive boosting classifier (AdaBoost), LR, Stochastic Gradient classifier (SGD), Random Forest (RF), Gradient Boosting classifier (GBM), Extra Tree Classifier (ETC), Gaussian Naive Bayes (G-NB), and SVM. The imbalance class problem is handled by Synthetic Minority Oversampling Technique (SMOTE). Furthermore, machine learning models are trained on the highest ranked features selected by RF. Experimental results demonstrate that ETC outperforms other models with SMOTE in prediction of heart patient’s survival.

Wankhede et al., [27] proposed an efficient heart disease prediction based on optimal feature selection. Initially, the data pre-processing process is performed using data cleaning, data transformation, missing values imputation, and data normalisation. Then, Decision Function-based Chaotic Salp Swarm (DFCSS) algorithm is introduced to optimal best features. Then the chosen featrues are given to the Improved Elman Neural Network (IENN) for data classification. Here, the Sailfish Optimisation (SFO) algorithm is used to compute the optimal weight value of IENN. The combination of DFCSS–IENN-based SFO (IESFO) algorithm effectively predicts heart disease. This approach is implemented in the Python environment using two different datasets such as the University of California Irvine (UCI) Cleveland heart disease dataset and CVD dataset.

Saranya and Pravin [28] presented a random forest-feature sensitivity and feature correlation (RF-FSFC) technique for enhanced heart disease prediction. This technique is implemented using the Cleveland heart disease dataset which comprises a total of 120 heart disease patient records. Data imputation was utilized for missing values, and min–max normalization was utilized for data transformation. RF-based classifiers are introduced for coronary heart disease by combining feature sensitivity and correlation analysis. The sensitivity-based feature selection process ranks features according to their value in assessing CHD risk, and the feature correlation analysis. Results are measured using metrics like sensitivity, specificity, Positive Predictive Value (PPV), and Negative Predictive Value (NPV). RF-FSFC approach significantly improves prediction accuracy as compared to other approaches.

Wadhawan and Maini [29] proposed an effective technique for cardiac disease (ETCD) prediction based on machine intelligence. To ensure the success of the proposed model consists of data collection, data pre-processing, and feature selection process to generate accurate data for the training model. Optimal Feature Subset Selection Algorithm (OFSSA) is introduced to extract features from different datasets (Cleveland, Hungarian, Combined dataset, and Z_Alizadeh Saini datasets) having varying properties available at the UCI machine learning repository. ETCD outperformed several current approaches in terms of sensitivity, specificity, precision, F-Score, and accuracy. ETCD returns the best feasible solution among all input predictive models considering performance criteria and improves the effectiveness of the system, therefore can assist doctors and radiologists in a better way to diagnose cardiac patients.

Alghamdi et al., [30] proposed an automated ML model for various kinds of CVD prediction and classification. Firstly, a benchmark dataset is preprocessed using filter techniques. Secondly, a novel arithmetic optimization algorithm (AOA) is implemented as a feature selection technique to select the best subset of features that influence the accuracy of the prediction model. Thirdly, a classification task is implemented using a Multilayer Perceptron (MLP) neural network to classify the instances of the dataset into two class labels, determining whether they have a CVD or not. The proposed ML model is trained on the preprocessed data and then tested and validated using evaluation metrics. Furthermore, for the comparative analysis of the model, various performance evaluation metrics are calculated including overall accuracy, precision, recall, and F1-score.

Dwarakanath et al., [31] proposed a novel Feature Selection with Hybrid Deep Learning-based Heart Disease Detection and Classification (FSHDL-HDDC) model. Preprocessing processes of the FSHDL-HDDC approach are data normalisation and the replacement of missing values. The FSHDL-HDDC method also necessitates the development of a feature selection method based on the Elite Opposition-based Squirrel Search Algorithm (EO-SSA) to determine the optimal subset of features. Moreover, an Attention-Based Convolutional Neural Network with Long Short-Term Memory (ACNN-LSTM) model is utilized for the detection of HD by using medical data. An extensive experimental evaluation is performed to ensure the improved classification performance of the proposed technique.

3. PROPOSED METHODOLOGY

Initially preprocessing is performed using Min-Max normalization to normalize the input data scale. Secondly, Conditional Generative Adversarial Network (CGAN) based oversampling technique is developed for generating new data by learning the distribution of the existing data for unstructured data. Thirdly, Gaussian Bobcat Optimization Algorithm (GBOA) is introduced to select most relevant features for detecting the heart disease. Finally, Memory-Augmented Deep Autoencoder (MADAE) classifier is introduced consists of an encoder network, a decoder network, and a memory for heart disease detection. Overall process of proposed system is illustrated in figure 1.

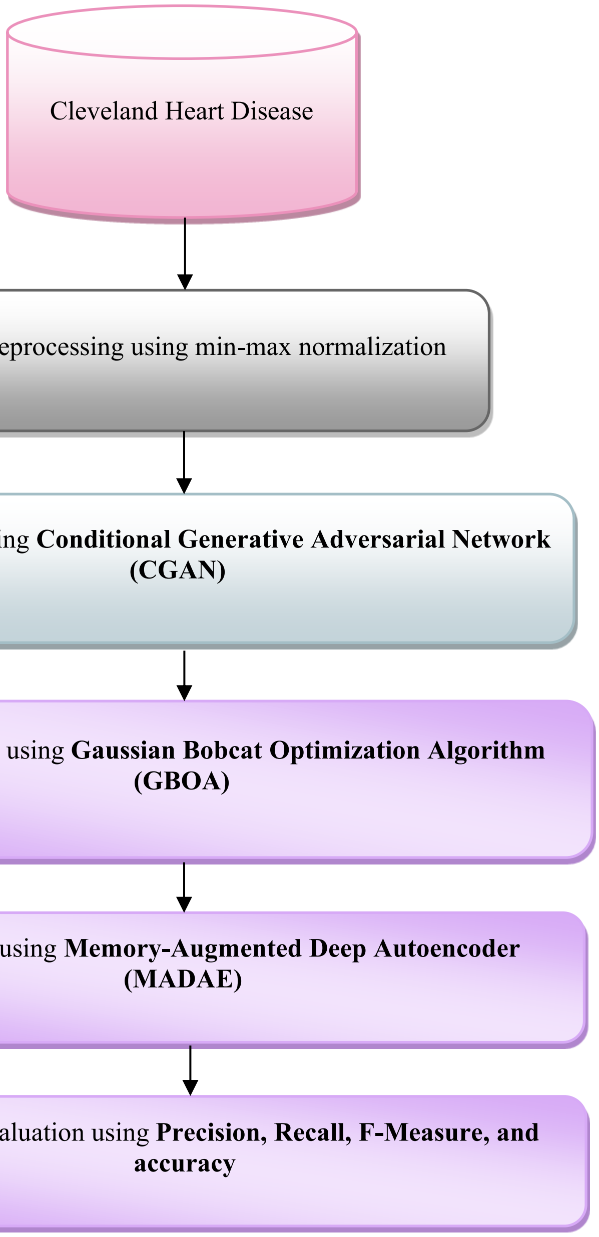
Cleveland Heart Disease
Data preprocessing using min-max normalization
Data balancing using **Conditional Generative Adversarial Network (CGAN)**
Feature selection using **Gaussian Bobcat Optimization Algorithm (GBOA)**
Classification using **Memory-Augmented Deep Autoencoder (MADAE)**
Performance evaluation using **Precision, Recall, F-Measure, and accuracy**

FIGURE 1. OVERALL PROCESS OF PROPOSED MODEL

3.1 DATASET

Cleveland database is collected from UCI (https://archive.ics.uci.edu/dataset/45/heart+disease). This database contains 76 attributes, but all published experiments refer to using a subset of 14 of them. In particular, the Cleveland database is the only one that has been used by ML researchers to date. The "goal" field refers to the presence of heart disease in the patient. It is integer valued from 0 (no presence) to 4. Experiments with the Cleveland database have concentrated on simply attempting to distinguish presence (values 1,2,3,4) from absence (value 0). The names and social security numbers of the patients were recently removed from the database, replaced with dummy values.

3.2 DATA PREPROCESSING USING MIN-MAX NORMALIZATION

Normalization is a data preprocessing technique used to transform the values of numeric columns in the dataset to a common scale, without distorting differences in the ranges of values or losing information. Min-max normalization performs a linear transformation on the original data. This technique gets all the scaled data in the range (0, 1). It has been described by equation (1),

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{1}$$

Min-max normalization preserves the relationships among the original data values.

3.3 DATA BALANCING USING CONDITIONAL GENERATIVE ADVERSARIAL NETWORK (CGAN)

A class imbalance problem occurs when a dataset is decomposed into one majority class and one minority class. This problem is critical in the machine learning domains because it induces bias in training machine learning models. One popular method to solve this problem is using a sampling technique to balance the class distribution by either under-sampling the majority class or over-sampling the minority class. Traditional oversampling approaches are based on Synthetic Minority Oversampling TEchnique (SMOTE), which focus on local information but generate insufficiently realistic data. In contrast, the Conditional Generative Adversarial Network (CGAN) captures the true data distribution to generate data for the minority class [32].

CGAN is directly derived from GAN. GAN are two networks like generative network G and discriminative network D, in which G is used to establish the true data distribution, and D is used to estimate the probability that a generated sample comes from the dataset using G. In order to obtain the distribution, p_g, of the dataset, x, G will establish a mapping from the prior noise distribution, $p_z(z)$, to the data space (x), G outputs a scalar, D(x,θ_d) to represent the probability that x comes from the dataset. During the training, the parameter updates of G and D are carried out simultaneously. When adjusting the parameters of G, the goal is to minimize log(1 − D(G(z))), and when adjusting the parameters of D, the goal is converted to log D(G(z)). They counteract each other and optimize the following functions in the opposite direction [32],

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}\left[\log\left(1 - D\big(G(z)\big)\right)\right] \tag{2}$$

Compared with GAN, it also takes the condition y as the input of the generator and discriminator. The generic framework of CGAN could be seen in Figure 2.

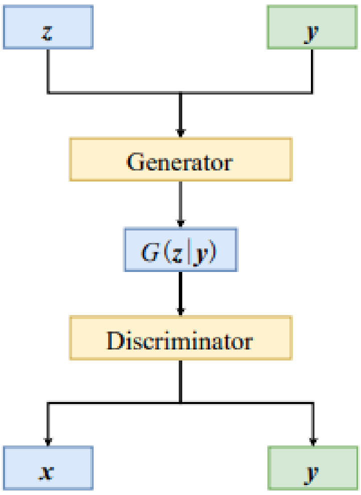

FIGURE 2. GENERIC FRAMEWORK OF CGAN

If the above generative network and discriminative network are conditional on some additional information y, then GAN will be extended to CGAN. Here, y can be any additional information, such as category labels or other modal data. In the design of network architecture, y is often used as the input for both the generative network and discriminative network. For G, the prior noise $p_z(z)$ and the condition y are concatenated for the hidden layer; for D, sample data or generated data x and condition y are concatenated as the input. Therefore, the objective function becomes [32],

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{data}(x)}[\log D(x|y)] + \mathbb{E}_{z\sim p_z(z)}\left[\log\left(1 - D\big(G(z|y)\big)\right)\right] \tag{3}$$

To be specific, given two types of data (domain X and domain Y), the input of the generative network will be random vector z and samples x of the domain X during the training process.

3.4 FEATURE SELECTION USING GAUSSIAN BOBCAT OPTIMIZATION ALGORITHM (GBOA)

GBOA approach is introduced to update the population of the algorithm in the feature selection searching space, it is inspired by the hunting strategy of bobcats in the wild. In this strategy, the bobcat first tracks the position of the prey and moves towards it. Then it ambushes and attacks the prey at the right time and finally catches it after a chasing process. According to this, changes in the feature position of the bobcat in its habitat during the hunting process can be considered in two parts: (i) tracking and moving towards the prey and (ii) chasing and catching the prey. Inspired by this natural strategy in the lifestyle of bobcats, in the GBOA design, the position of the population members are updated in each iteration in two phases (i) exploration based on the simulation of the bobcat's position change while moving towards the prey and (ii) exploitation based on the simulation of the bobcat's position change during the chase process to catch the prey [33].

Initialization: GBOA approach is a population-based optimizer that can achieve suitable solutions for feature selection in an iteration-based process by benefiting from the searching power of its members in the problem solving space. GBOA design inspiration, the wildlife habitat of the bobcats corresponds to the feature selection space, and the location of the bobcats in this habitat corresponds to the feature position of the GBOA members in the feature selection search space. GBOA, each bobcat as a member of the population according to the feature position it creates in the searching space, determines the values for the decision variables. Hence, the position of each bobcat represents a feature solution to the problem, which can be modeled from a mathematical point of view using a vector, where each element of this feature vector represents a decision variable. Together, bobcats form the population of the algorithm, which can be modeled from a mathematical point of view using a matrix according to equation (4). The primary position of bobcats in the feature search space is initialized randomly using equation (5) [33],

$$X = \begin{bmatrix} x_{1,1} & \cdots & x_{1,d} & \cdots & x_{1,m} \\ \vdots & \cdots & \vdots & \cdots & \vdots \\ x_{i,1} & \cdots & x_{i,d} & \cdots & x_{i,m} \\ \vdots & \cdots & \vdots & \cdots & \vdots \\ x_{N,1} & \cdots & x_{N,d} & \cdots & x_{N,m} \end{bmatrix}_{N\times m} \tag{4}$$

$$x_{i,d} = lb_d + r \cdot (ub_d - lb_d) \tag{5}$$

Here, X is the GBOA population matrix, X_i is the ith bobcat (feature selection solution), $x_{i,d}$ is its dth dimension of feature in search space, N is the number of bobcats, m is the number of features, $r \in [0,1]$ is a random number in interval, lb_d, and $ub_d$are the lower bound and upper bound of the dth dimension of feature respectively. The position of each bobcat represents a feature solution for the heart disease detection, corresponding to which the objective function. The set of evaluated values for the objective function can be represented using a feature vector according to equation (6) [33].

$$F = \begin{bmatrix} F(X_1) \\ \vdots \\ F(X_i) \\ \vdots \\ F(X_N) \end{bmatrix}_{N\times 1} \tag{6}$$

Here, F is the vector of evaluated fitness function and F_i is the evaluated fitness function based on the ith bobcat. These evaluated values for the fitness function of the feature selection are suitable criteria to measure the quality of each GBOA member in providing feature solutions. According to this, the best evaluated value for the fitness function corresponds to the best GBOA member and similarly the worst feature evaluated value for the fitness function corresponds to the worst GBOA member. In GBOA design, the position of the bobcats in the feature selection search space is updated, and accordingly, the feature solutions and the fitness function values are also updated. Each iteration based on the comparison of fitness function values, the best GBOA member should also be updated [33].

PHASE 1: Tracking and moving towards prey (exploration phase)

In the first phase of GBOA, the position of the population members in the feature selection search space is updated based on the simulation of tracking and movement of bobcats towards prey during hunting. Modeling the movement of bobcat towards the prey leads to extensive changes in the position of the population members in the feature selection search space and thus increases the exploration ability of GBOA to manage the global search. In GBOA design, for each bobcat, the position of other population members who have a better value for the fitness function is considered as the prey position. The candidate prey set for each bobcat is determined using equation (7) [33],

$$CP_i = \{X_k : F_k < F_i \; and \; k \neq i\}, \text{where i} = 1, \ldots, \text{N} \;\&\; k \in \{1, \ldots, \text{N}\} \tag{7}$$

Here, $CP_i$is the set of candidate preys' locations for the ith bobcat, $X_k$is the is the population member with a better fitness function than ith bobcat, and F_k is the its fitness function. In the GBOA design, it is assumed that each bobcat randomly chooses anyone of feature from these preys and attacks it. Based on the modeling of the bobcat's position change while moving towards the prey in this strategy, a new feature position is calculated for each GBOA member using equation (8). This new feature position replaces the previous feature position of the corresponding member if it improves the value of the objective function according to equation (9) [33],

$$x_{i,j}^{P1} = x_{i,j} + (1 - 2r_{i,j}) \cdot (SP_{i,j} - I_{i,j} \cdot x_{i,j}) \tag{8}$$

$$X_i = \begin{cases} X_i^{P1}, F_i^{P1} \leq F_i \\ X_i, else \end{cases} \tag{9}$$

Here, $SP_i$is the selected prey by ith bobcat, $SP_{i,j}$ is its jth dimension, $X_i^{P1}$is the new position calculated for the ith bobcat based on exploration phase of the proposed GBOA, $x_{i,j}^{P1}$ is its jth dimension (feature), $F_i^{P1}$is its fitness function value, $r_{i,j} \in [0,1]$ are random numbers from the interval, and $I_{i,j}$ are numbers which are randomly selected as 1 or 2 [33].

PHASE 2: Chasing to catch prey (exploitation phase)

In the second phase of GBOA, the position of the population members in the feature selection space is updated based on the chase simulation between the bobcat and the prey during hunting. This process of chasing happens near the hunting place so that finally the bobcat catches the prey. Modeling the movement of bobcat during the process of chasing and catching prey leads to small changes in the feature position of population members in the feature selection search space and thus increases the exploitation ability of GBOA to manage local search. In GBOA design, based on the modeling of bobcat position change during the chase process, a new position for each GBOA member near the hunting place is calculated using equation (10). Then, this new position, if it improves the value of the fitness function, replaces the previous position of the corresponding member according to equation (11) [33],

$$x_{i,j}^{P2} = x_{i,j} + \frac{1 - 2r_{i,j}}{1 + t} \cdot x_{i,j} \tag{10}$$

$$X_i = \begin{cases} X_i^{P2}, F_i^{P2} \leq F_i \\ X_i, else \end{cases} \tag{11}$$

Here, X_i^{P2} is the new feature position calculated for the ith bobcat based on exploitation phase of the proposed GBOA, $x_{i,j}^{P2}$ is its jth dimension (feature), F_i^{P2} is its fitness function, $r_{i,j} \in [0,1]$ are random numbers from the interval, and t is the iteration counter [33]. In the equation (8)& (10), $r_{i,j}$ is generated via Gaussian distribution. Gaussian distribution is a continuous function which approximates the exact binomial distribution of random values by equation (12),

$$y = \frac{1}{\sqrt{2\pi\sigma}} e^{-(r-\mu)^2/2\sigma^2} \tag{12}$$

where r is random value, y is the relative frequency of x, μ is the center of the distribution, σ is the standard deviation of the distribution, π is the constant 3.1416, and e is the constant 2.7183.

Population diversity of GBOA refers to how population members are spread out within the feature selection space is crucial for finding optimal best solution. Evaluating the diversity within the GBOA population allows for assessing and adjusting the algorithm ability to effectively explore and exploit as a group. It is described diversity using equations (13-14) [33],

$$Diversity = \frac{1}{N} \sum_{i=1}^{N} \sqrt{\sum_{d=1}^{m} (x_{i,d} - \bar{x}_d)^2} \tag{13}$$

$$\bar{x}_d = \frac{1}{N} \sum_{i=1}^{N} x_{i,d} \tag{14}$$

Here, N represents the number of population members, m is the number of feature dimensions, and $x_d$is the mean of the population in the dth dimension (feature). Therefore, the extent of exploration and exploitation within the population for each iteration can be defined by equations (15-16) [33],

$$Exploration = \frac{Diversity}{Diversity_{max}} \tag{15}$$

$$Exploitation = \frac{|Diversity - Diversity_{max}|}{Diversity_{max}} \quad (16)$$

The first iteration in the proposed GBOA approach is completed after updating the position of all bobcats in the feature selection search space based on exploration and exploitation phases. After that, the algorithm enters the next iteration with updated values for the position of bobcats and the fitness function, and the process of updating bobcats continues until the last iteration of the algorithm based on equations (7)–(16). In each iteration, the best selected feature solution is obtained until that iteration is updated and saved. After the full implementation of the algorithm, the best feature solution obtained during the iterations of the algorithm is presented as the GBOA solution for the given disease detection. The implementation steps of GBOA are shown as a flowchart in Figure 3 and its pseudocode is presented in Algorithm 1 [33].

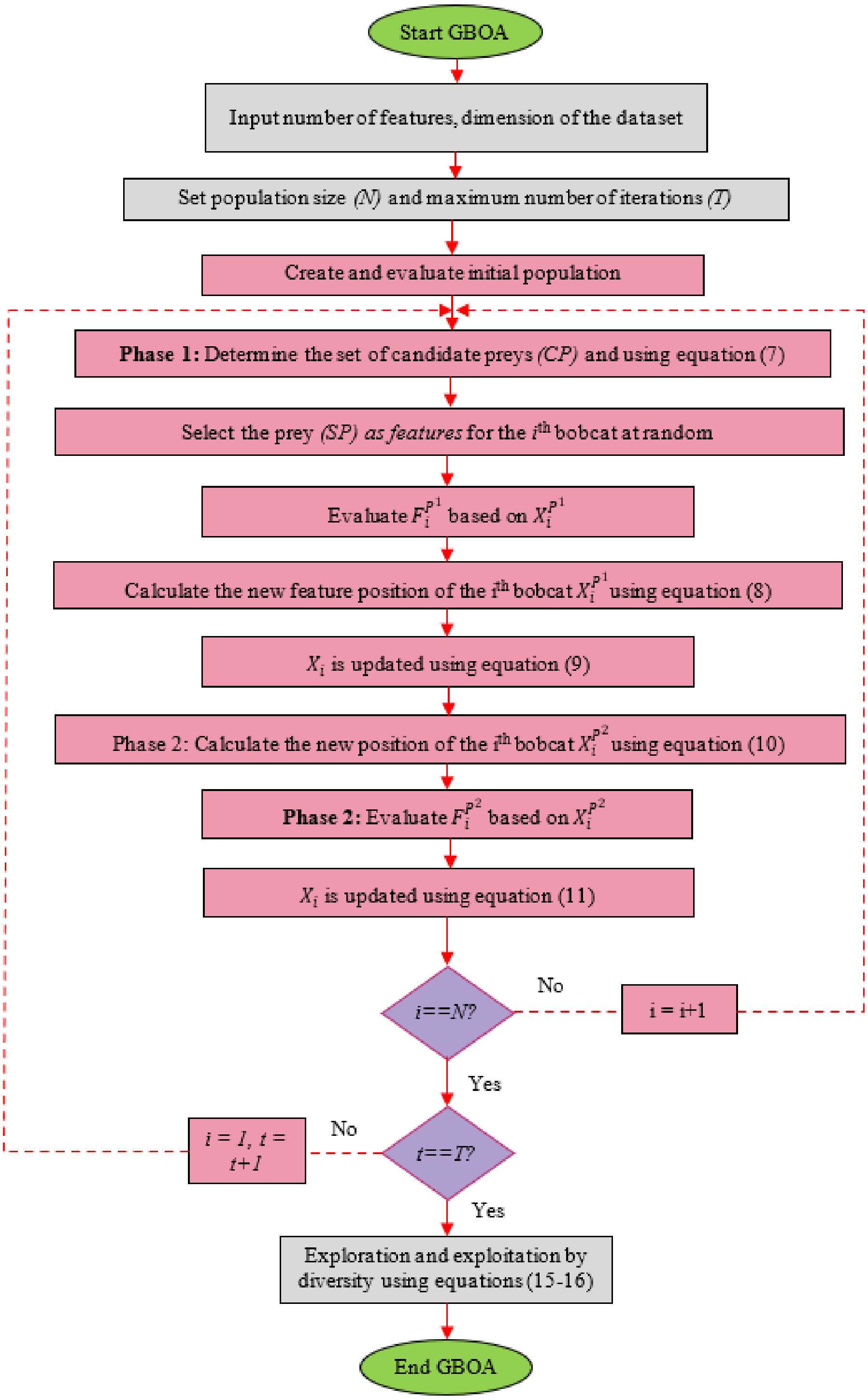
Start GBOA
Input number of features, dimension of the dataset
Set population size (N) and maximum number of iterations (T)
Create and evaluate initial population
Phase 1: Determine the set of candidate preys (CP) and using equation (7)
Select the prey (SP) as features for the i^{th} bobcat at random
Evaluate F_i^{P1} based on X_i^{P1}
Calculate the new feature position of the i^{th} bobcat X_i^{P1} using equation (8)
X_i is updated using equation (9)
Phase 2: Calculate the new position of the i^{th} bobcat X_i^{P2} using equation (10)
Phase 2: Evaluate F_i^{P2} based on X_i^{P2}
X_i is updated using equation (11)
i==N?
No
i = i+1
Yes
t==T?
No
i = 1, t = t+1
Yes
Exploration and exploitation by diversity using equations (15-16)
End GBOA

FIGURE 3. FLOWCHART OF GBOA

ALGORITHM 1. GAUSSIAN BOBCAT OPTIMIZATION ALGORITHM (GBOA)

```
Start GBOA
    Input Cleveland heart disease dataset, number of features, number of samples, fitness function, and constraints
    Start GBOA with population size (N) and iterations (T)
    Initial population matrix is generated at random using Equation (4)
    Evaluate the fitness function as classification accuracy
    For t = 1 to T
      For i = 1 to N
        Phase 1: Tracking and moving towards prey (exploration phase)
          Determine the preys set of no. of features for the i^th member by Equation (7)
            Select the termite features for the i^th BOA member at random
              Calculate the new feature position of i^th GBOA member using equation (8)
                best features are selected as member is updated using equation (9)
        Phase 2: Chasing to catch prey (exploitation phase)
            Calculate new position i^th GBOA member by Equation (10)
              Best features are selected as the i^th member using Equation (11)
      End for i
      Save the best candidate and found the Exploration and exploitation by diversity using equations (15-16) solution so far
    End for t
    Output the best optimal feature selection with GBOA
End GBOA
```

3.5 CLASSIFICATION USING MEMORY-AUGMENTED DEEP AUTOENCODER (MADAE)

Autoencoder (AE) is a type of Neural Network (NN) model used to learn efficient heart disease dataset. As shown in figure 4, the AE is composed of an encoder network and a decoder network. In general, the encoder network and the decoder network have a regular structure from the center bottleneck [34]. The encoder network in equation (17) maps the original Cleveland dataset x onto a low-dimensional feature space, while the decoder network in equation (18) attempts to recover $\hat{x}$ from the projected low-dimensional space [35]. The AE aims to learn to encode that preserves as much important Cleveland as possible for input reconstruction.

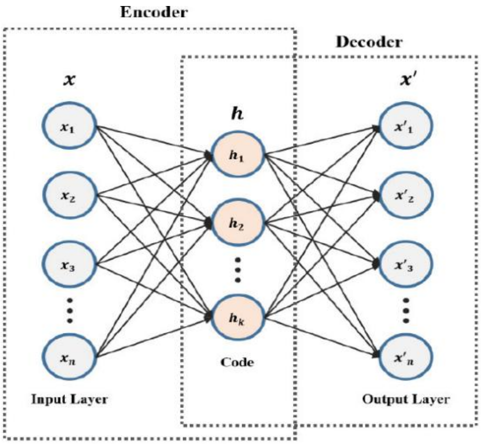

FIGURE 4. AUTOENCODER STRUCTURE

The parameters of these two networks are learned with a reconstruction loss function,

$$z = \sigma(We_e x + bi_e) \tag{17}$$

$$\hat{x} = \sigma(We_d x + bi_d) \tag{18}$$

An AE with a single hidden layer has an encoder and decoder as in equation (17) and equation (18), We and bi are the weight and bias and σ is the activation function. The activation function σ can use nonlinear functions and linear functions. z is the hidden presentation called the latent vector or code. The decoder network, z is used as the input to output the reconstructed $\hat{x}$ by equation (18). A Stacked autoencoder (SAE) is a model in which the autoencoders have multiple hidden layers. Memory-Augmented Deep Autoencoder (MADAE) consists of an encoder network, a decoder network, and a memory module as shown in Figure 5 [36].

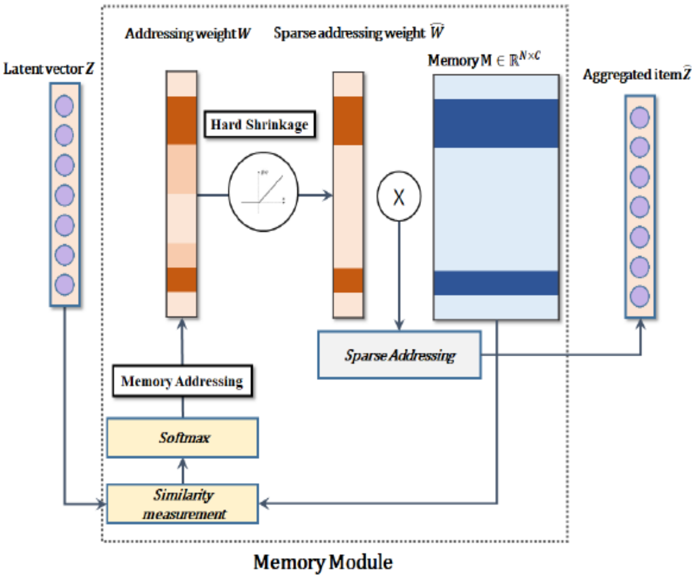

FIGURE 5. MEMORY MODULE IN MADAE

The purpose of the memory module is to learn and record a finite number of prototypical patterns of the input dataset. The memory module is located between the encoder network and the decoder network, and it receives the output z of the encoder as an input and outputs the dataset $\hat{x}$ for delivery to the decoder. So, MADAE does not feed the output z of the encoder directly to the decoder. Internally, the latent vector z is used as a query to retrieve the most relevant data in memory. Then those features are aggregated and passed to the decoder. As a result, this internal process of the memory module makes it possible to induce an output close to a positive sample because even if the latent vector of negative input becomes an input, it is aggregated and reconstructed from the normal prototypical patterns learned in memory. Therefore, it is possible to solve the problem of over-generalization in which negative samples are well reconstructed [36],

$$z_i = wM = \sum_{i=1}^{N} w_i m_i \tag{19}$$

$$w_i = \frac{\exp\big(d(z, m_i)\big)}{\sum_{j=1}^{N} \exp\big(d(z, m_j)\big)} \tag{20}$$

The memory is matrix $M \in \mathbb{R}^{N\times C}$ containing N real valued vectors of fixed dimension C, as shown in Figure 5. As shown in equation (19), aggregated dataset $\hat{x}$ is obtained through soft addressing of vector w and memory. Memory M reflects features to record various prototypical patterns of positive dataset. The weight vector w is obtained from z and it is computed through a softmax operation by equation (20), where d is a cosine similarity. In addition, as shown in equation (21), memory modules can increase sparsity through hard shrinkage operations on soft addressing vector w. The sparse addressing encourages the model to represent an example using smaller amount, but more relevant memory dataset, leading to learning more useful representations in memory. ReLU activation function is considering that not all input values are negative by equation (22). The value of λ is the shrinkage threshold, which uses the value of interval [1/N, 3/N] [36],

$$\widehat{w}_i = h(w_i, \lambda) = \begin{cases} w_i, if\ w_i > \lambda \\ 0, else \end{cases} \tag{21}$$

$$\widehat{w}_i = \frac{\max(w_i - \lambda, 0) \cdot w_i}{|w_i - \lambda| + \varepsilon} \tag{22}$$

After the hard shrinkage operation, re-normalization is performed with $\widehat{w} = \widehat{w}\ /\|\widehat{w}\|_1$: sparse memory (SM) and non-sparse memory. Samples reconstructed through memory modules tend to be close to the normal samples used for learning. Therefore, the abnormal sample is reconstructed close to the positive sample, which means that the reconstruction error value for the negative inputs increases.

4. EXPERIMENTAL RESULTS AND DISCUSSIONS

In this section, briefly describe the dataset incorporated in the experiment of this study. As mentioned in previous sections, make use of a state-of-the-art dataset. This set of attributes is a subset of a dataset compiled by medical practitioners in African countries. It incorporates only 14 attributes from this dataset to predict the presence of a CHD. Table 1 depicts a list of features in the algorithm with their short description and the possible range of values wherever applicable. Cleveland database is collected from UCI repository available at (https://archive.ics.uci.edu/dataset/45/heart+disease).This database contains 13 attributes and 303 instances and in which 70.00% is used for training and 30.00% is used for testing.

TABLE 1. ATTRIBUTES IN CLEVELAND DATABASE

ATTRIBUTE	DESCRIPTION	POSSIBLE VALUES
Age	Age (in years) when admitted to the hospital	Valid numbers
Sex	Sex of person	1: male, 0: female
Cp	Type of chest pain (anginal pain)	1: typical angina, 2: atypical angina, 3: non-anginal pain, 4: asymptomatic (Nominal)
trestbps	Blood pressure while resting (when admitted to hospital) in mm/Hg	90–200
Chol	Level cholesterol in mg/dl	125–565
Fbs	Blood sugar (fasting) is fbs>120 mg/dl?	1: true, 0: false
restecg	Results of ECG while resting	0: normal, 1: ST-T wave abnormality, 2: left ventricular hypertrophy (with Estes' criteria)
thalach	Maximum heart rate	71–202
exang	Angina induced by exercise	1 =true, 0 =false
old peak	Exercise induced ST-depression in relative with the state of rest	0–7
Slope	ST segment measured in terms of slope during peak exercise	1: upsloping, 2: flat, 3: down-sloping
Ca	Number of major vessels by fluoroscopy	0-3
Thal	A blood disorder called thalassemia	3= normal, 6 = fixed defect, and 7=reversable Defect
num	diagnosis of heart disease	0: less than 50% narrowing diameter, 1: greater than 50% narrowing diameter

4.1 Experimental Setup

Experimental workstation are equipped with an Intel Quad-Core i7 4th generation processor, working at a clock rate of 2.3 GHz with an L1 cache of 32 KB, L2 of 256 KB and L3 cache memory with 4 MB of size. This work incorporates Microsoft Windows 10 Pro as the base operating system with MATrix LABoratory R2018a (MATLABR2018a).

4.2 Performance Metrics

This section evaluates the performance of the classifiers (SVM, IENN, ACNN-LSTM, HCNNSVM, and MADAE) by using some performance metrics. In machine learning, various criteria are available for evaluating the classifier's performance. Some of these criteria are explained as follows,

Precision: Precision is the measure of exactness for evaluating the performance of a classifier. If the precision is high, it means there are fewer false positives. In a model with lower precision means, there are more false positives. It can be calculated by the following equation (23),

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{23}$$

Recall: Recall is a measure to determine the completeness of the classifier. Higher the recall, lesser the false negatives, and lower the recall, higher the false negatives. Improvement in recall often results in a decrease in precision. It can be calculated by the following equation (24),

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{24}$$

F-Measure: The combination of precision and recall is called F-Measure. It can be calculated by the following equation (25),

$$F - Measure = \frac{2 * Precision * Recall}{Precision + Recall} \tag{25}$$

As depicted in equation (25), the F-measure can be obtained by dividing the product of recall and precision with the sum of recall and precision.

Accuracy: Accuracy is an evaluation metric that measures the proportion of correct model predictions out of all predictions made.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{26}$$

Table 2 shows the metrics has been evaluated using classifiers and compared against feature selection methods like DFCSS, AOA, EO-SSA, ICSO, and GBOA.

TABLE 2. PERFORMANCE METRICS RESULTS COMPARISON OF CLASSIFIERS

FS& Classifiers	**Precision (%)**				
	SVM	**IENN**	**ACNN-LSTM**	**HCNNSVM**	**MADAE**
DFCSS	76.78	78.03	80.31	84.09	86.22
AOA	80.56	81.67	83.75	86.21	88.26
EO-SSA	82.27	83.54	86.62	87.82	89.10
ICSO	84.17	85.44	86.25	88.05	91.19
GBOA	85.98	88.53	90.38	91.61	93.54
FS& Classifiers	**Recall (%)**				
	SVM	**IENN**	**ACNN-LSTM**	**HCNNSVM**	**MADAE**
DFCSS	81.21	82.07	83.61	85.19	87.04
AOA	83.73	85.16	86.37	88.55	90.41
EO-SSA	85.52	86.84	88.23	90.13	92.06
ICSO	86.92	88.23	91.39	92.71	93.55
GBOA	87.66	89.10	92.15	92.81	94.16
FS& Classifiers	**F-Measure (%)**				

	SVM	IENN	ACNN-LSTM	HCNNSVM	MADAE
DFCSS	78.93	79.99	81.93	84.64	86.63
AOA	82.12	83.37	85.04	87.36	89.32
EO-SSA	83.87	85.16	87.37	88.96	90.56
ICSO	85.53	86.81	88.74	90.32	92.35
GBOA	86.81	88.82	91.26	92.21	93.85
FS& Classifiers	**Accuracy (%)**				
	SVM	**IENN**	**ACNN-LSTM**	**HCNNSVM**	**MADAE**
DFCSS	80.21	81.07	83.79	85.19	86.35
AOA	82.32	83.75	86.11	87.31	89.26
EO-SSA	83.49	84.82	87.13	88.78	90.43
ICSO	85.15	86.47	88.49	90.09	92.07
GBOA	86.47	88.45	91.09	92.08	93.73

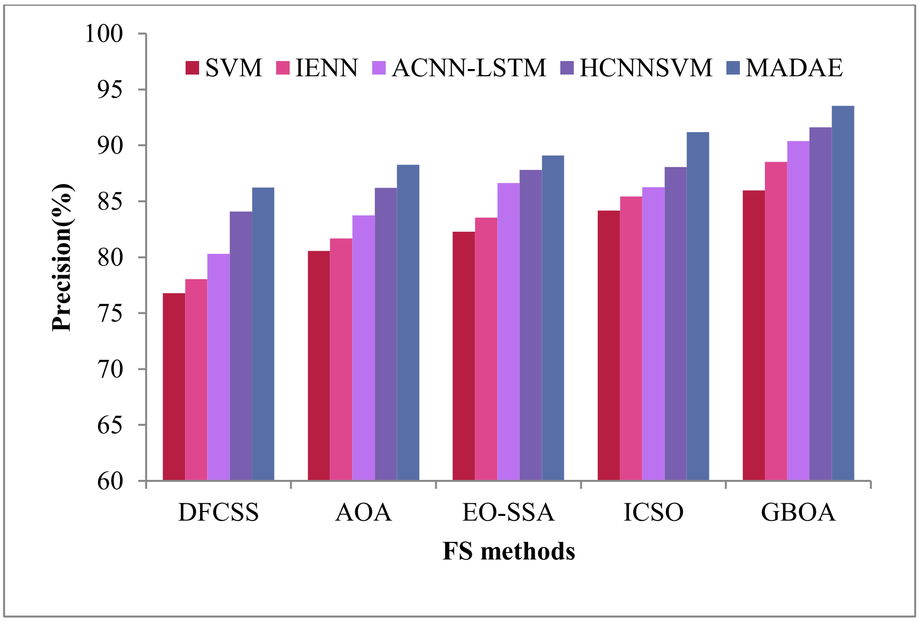

FIGURE 6. PRECISION RESULTS COMPARISON OF CLASSIFIERS

Figure 6 shows the precision results comparison of classifiers like SVM, IENN, ACNN-LSTM, HCNNSM, and MADAE with feature selection methods (DFCSS, AOA, EO-SSA, ICSO, and GBOA). The proposed system has highest precision results of 86.22%, 88.26%, 89.10%, 91.19%, and 93.54% for DFCSS, AOA, EO-SSA, ICSO, and GBOA. SVM, IENN, ACNN-LSTM, and HCNNSM have precision results of 85.98%, 88.53%, 90.38%, and 91.61% against GBOA.

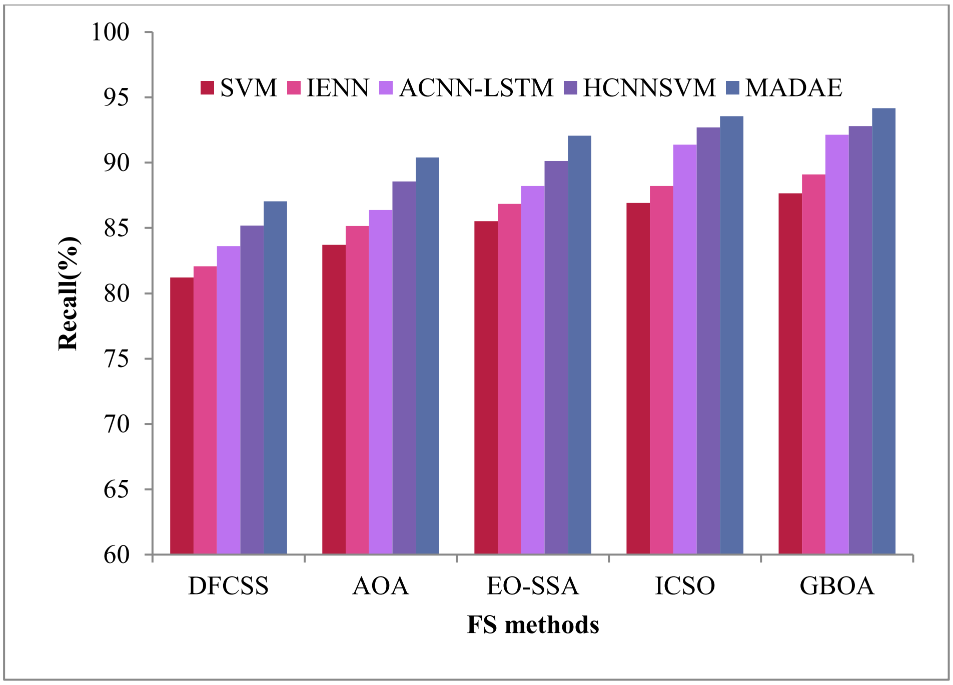

FIGURE 7. RECALL RESULTS COMPARISON OF CLASSIFIERS

Recall results comparing classifiers like SVM, IENN, ACNN-LSTM, HCNNSM, and MADAE with feature selection methods (DFCSS, AOA, EO-SSA, ICSO, and GBOA) are illustrated in figure 7. The proposed system has highest recall results of 87.04%, 90.41%, 92.06%, 93.55%, and 94.16% for DFCSS, AOA, EO-SSA, ICSO, and GBOA. Other classifiers like SVM, IENN, ACNN-LSTM, and HCNNSM have recall results of 87.66%, 89.10%, 92.15%, and 92.81% against GBOA.

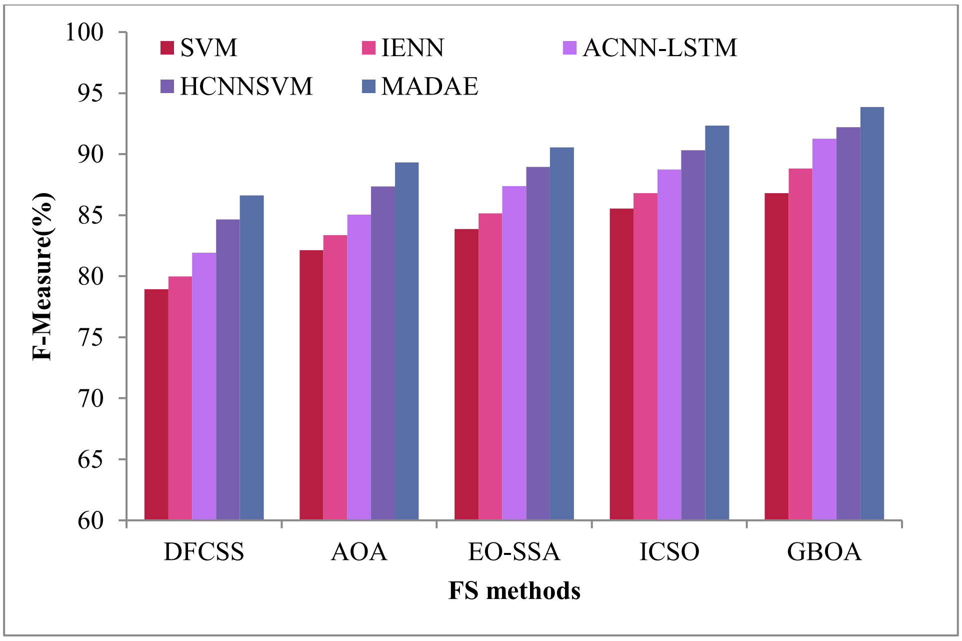

FIGURE 8. F-MEASURE RESULTS COMPARISON OF CLASSIFIERS

F-measure results comparison of classifiers like SVM, IENN, ACNN-LSTM, HCNNSM, and MADAE with feature selection methods (DFCSS, AOA, EO-SSA, ICSO, and GBOA) are illustrated in figure 8. The proposed system has highest f-measure results of 86.63%, 89.32%, 90.56%, 92.35%, and 93.85 % for DFCSS, AOA, EO-SSA, ICSO, and GBOA. SVM, IENN, ACNN-LSTM, and HCNNSM have f-measure results of 86.81%, 88.82%, 91.26%, and 92.21% against GBOA.

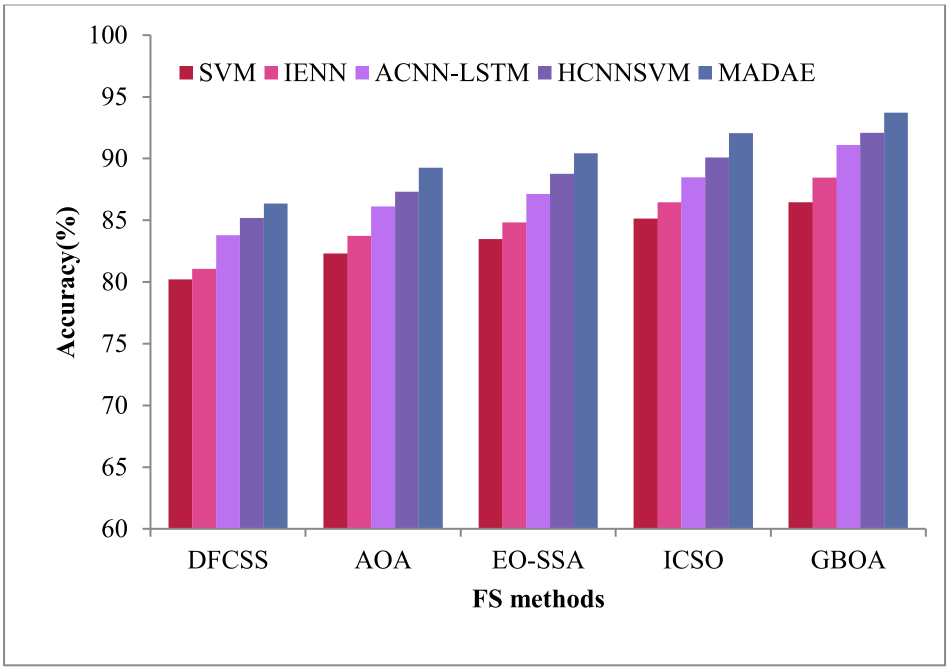

FIGURE 9. ACCURACY RESULTS COMPARISON OF CLASSIFIERS

Classifiers like SVM, IENN, ACNN-LSTM, HCNNSM, and MADAE with feature selection methods (DFCSS, AOA, EO-SSA, ICSO, and GBOA) among accuracy are illustrated in figure 9. The proposed system has highest accuracy results of 86.35%, 89.26%, 90.43%, 92.07%, and 93.73% for DFCSS, AOA, EO-SSA, ICSO, and GBOA. SVM, IENN, ACNN-LSTM, and HCNNSM have accuracy results of 86.47%, 88.45%, 91.09%, and 92.08% against GBOA.

5. CONCLUSION AND FUTURE WORK

In this paper, prediction of heart diseases at earlier stages may prevent possible deaths due to heart attacks. A good classification algorithm and balanced samples may help the physician predict the presence of cardiovascular disease before its actual occurrence. Initially, samples were collected which included 164 samples indicating the absence of heart disease and 139 samples indicate the presence of heart disease. Secondly, Conditional Generative Adversarial Network (CGAN) is a type of neural network to generate new data samples similar to a given training sample. CGAN has been used to consider the representation of the minority and majority classes equally. Thirdly, the Gaussian Bobcat Optimization Algorithm (GBOA) approach is introduced to the feature selection process and it is inspired by the hunting strategy of bobcats in the wild. Hunting process of feature selection can be considered in two parts: (i) tracking and moving towards the prey and (ii) chasing and catching the prey. Gaussian distribution is introduced for random number generation in these two parts. Diversity is introduced for updating the position of all bobcats in the FS process based on exploration and exploitation phases. Finally, Memory-Augmented Deep Autoencoder (MADAE) is introduced with memory module to learn and record a finite number of prototypical patterns of the input dataset. In the future, the same techniques can also be applied to other disease prediction and also some other intelligence techniques will be applied to select the most important features for predicting the coronary artery heart disease.

REFERENCES

[1] Indrakumari, R., Poongodi, T. and Jena, S.R., 2020. Heart disease prediction using exploratory data analysis. Procedia Computer Science, 173, pp.130-139.

[2] Fitriyani, N.L., Syafrudin, M., Alfian, G. and Rhee, J., 2020. HDPM: an effective heart disease prediction model for a clinical decision support system. IEEE Access, 8, pp.133034-133050.

[3] Pathan, M.S., Nag, A., Pathan, M.M. and Dev, S., 2022. Analyzing the impact of feature selection on the accuracy of heart disease prediction. Healthcare Analytics, 2, pp.1-9.

[4] Mohan, S., Thirumalai, C. and Srivastava, G., 2019. Effective heart disease prediction using hybrid machine learning techniques. IEEE access, 7, pp.81542-81554.

[5] Bhatt, C.M., Patel, P., Ghetia, T. and Mazzeo, P.L., 2023. Effective heart disease prediction using machine learning techniques. Algorithms, 16(2), pp.1-14.

[6] Yahaya, L., Oye, N.D. and Garba, E.J., 2020. A comprehensive review on heart disease prediction using data mining and machine learning techniques. American Journal of Artificial Intelligence, 4(1), pp.20-29.

[7] Aremu, O.O., Hyland-Wood, D. and McAree, P.R., 2020. A machine learning approach to circumventing the curse of dimensionality in discontinuous time series machine data. Reliability Engineering & System Safety, 195, p.106706.

[8] Pavithra, V. and Jayalakshmi, V., 2020, Review of feature selection techniques for predicting diseases. In 2020 5th International Conference on Communication and Electronics Systems (ICCES), pp. 1213-1217.

[9] Cai, J., Luo, J., Wang, S. and Yang, S., 2018. Feature selection in machine learning: A new perspective. Neurocomputing, 300, pp.70-79.

[10] Wang, G., Lauri, F. and El Hassani, A.H., 2021, A study of dimensionality reduction's influence on heart disease prediction. In 2021 12th International Conference on Information, Intelligence, Systems & Applications (IISA), pp. 1-6.

[11] Rostami, M., Berahmand, K., Nasiri, E. and Forouzandeh, S., 2021. Review of swarm intelligence-based feature selection methods. Engineering Applications of Artificial Intelligence, 100, pp.1-31.

[12] Nguyen, B.H., Xue, B. and Zhang, M., 2020. A survey on swarm intelligence approaches to feature selection in data mining. Swarm and Evolutionary Computation, 54, pp.1-27.

[13] Saw, T. and Myint, P.H., 2019. Swarm intelligence based feature selection for high dimensional classification: a literature survey. Int. J. Comput, 33(1), pp.69-83.

[14] Alrefai, N. and Ibrahim, O., 2022. Optimized feature selection method using particle swarm intelligence with ensemble learning for cancer classification based on microarray datasets. Neural Computing and Applications, 34(16), pp.13513-13528.

[15] Maldonado, S. and López, J., 2018. Dealing with high-dimensional class-imbalanced datasets: Embedded feature selection for SVM classification. Applied Soft Computing, 67, pp.94-105.

[16] Liu, H., Zhou, M. and Liu, Q., 2019. An embedded feature selection method for imbalanced data classification. IEEE/CAA Journal of Automatica Sinica, 6(3), pp.703-715.

[17] Kamalov, F., Thabtah, F. and Leung, H.H., 2023. Feature selection in imbalanced data. Annals of Data Science, 10(6), pp.1527-1541.

[18] Johnson, J.M. and Khoshgoftaar, T.M., 2019. Survey on deep learning with class imbalance. Journal of big data, 6(1), pp.1-54.

[19] Rendon, E., Alejo, R., Castorena, C., Isidro-Ortega, F.J. and Granda-Gutierrez, E.E., 2020. Data sampling methods to deal with the big data multi-class imbalance problem. Applied Sciences, 10(4), pp.1-57.

[20] Ghosh, K., Bellinger, C., Corizzo, R., Branco, P., Krawczyk, B. and Japkowicz, N., 2024. The class imbalance problem in deep learning. Machine Learning, 113(7), pp.4845-4901.

[21] Chen, Z., Duan, J., Kang, L. and Qiu, G., 2021. Class-imbalanced deep learning via a class-balanced ensemble. IEEE transactions on neural networks and learning systems, 33(10), pp.5626-5640.

[22] Sampath, V., Maurtua, I., Aguilar Martin, J.J. and Gutierrez, A., 2021. A survey on generative adversarial networks for imbalance problems in computer vision tasks. Journal of big Data, 8, pp.1-59.

[23] Douzas, G. and Bacao, F., 2018. Effective data generation for imbalanced learning using conditional generative adversarial networks. Expert Systems with applications, 91, pp.464-471.

[24] Ghosh, P., Azam, S., Jonkman, M., Karim, A., Shamrat, F.J.M., Ignatious, E., Shultana, S., Beeravolu, A.R. and De Boer, F., 2021. Efficient prediction of cardiovascular disease using machine learning algorithms with relief and LASSO feature selection techniques. IEEE Access, 9, pp.19304-19326.

[25] Li, J.P., Haq, A.U., Din, S.U., Khan, J., Khan, A. and Saboor, A., 2020. Heart disease identification method using machine learning classification in e-healthcare. IEEE access, 8, pp.107562-107582.

[26] Ishaq, A., Sadiq, S., Umer, M., Ullah, S., Mirjalili, S., Rupapara, V. and Nappi, M., 2021. Improving the prediction of heart failure patients' survival using SMOTE and effective data mining techniques. IEEE access, 9, pp.39707-39716.

[27] Wankhede, J., Kumar, M. and Sambandam, P., 2020. Efficient heart disease prediction-based on optimal feature selection using DFCSS and classification by improved Elman-SFO. IET systems biology, 14(6), pp.380-390.

[28] Saranya, G. and Pravin, A., 2023. A novel feature selection approach with integrated feature sensitivity and feature correlation for improved prediction of heart disease. Journal of Ambient Intelligence and Humanized Computing, 14(9), pp.12005-12019.

[29] Wadhawan, S. and Maini, R., 2022. ETCD: An effective machine learning based technique for cardiac disease prediction with optimal feature subset selection. Knowledge-Based Systems, 255, p.109709.

[30] Alghamdi, F.A., Almanaseer, H., Jaradat, G., Jaradat, A., Alsmadi, M.K., Jawarneh, S., Almurayh, A.S., Alqurni, J. and Alfagham, H., 2024. Multilayer perceptron neural network with arithmetic optimization algorithm-based feature selection for cardiovascular disease prediction. Machine Learning and Knowledge Extraction, 6(2), pp.987-1008.

[31] Dwarakanath, B., Latha, M., Annamalai, R., Kallimani, J.S., Walia, R. and Belete, B., 2022. A novel feature selection with hybrid deep learning based heart disease detection and classification in the e-healthcare environment. Computational Intelligence and Neuroscience, vol.2022, no.1167494, pp.1-12.

[32] Zhou, G., Fan, Y., Shi, J., Lu, Y. and Shen, J., 2022. Conditional generative adversarial networks for domain transfer: a survey. Applied Sciences, 12(16), pp.1-26.

[33] Benmamoun, Z., Khlie, K., Bektemyssova, G., Dehghani, M. and Gherabi, Y., 2024. Bobcat Optimization Algorithm: an effective bio-inspired metaheuristic algorithm for solving supply chain optimization problems. Scientific Reports, 14(1), pp.1-62.

[34] Xie, R., Wen, J., Quitadamo, A., Cheng, J. and Shi, X., 2017. A deep auto-encoder model for gene expression prediction. BMC genomics, 18, pp.39-49.

[35] Pan, Y., He, F. and Yu, H., 2020. Learning social representations with deep autoencoder for recommender system. World Wide Web, 23(4), pp.2259-2279.

[36] Min, B., Yoo, J., Kim, S., Shin, D. and Shin, D., 2021. Network anomaly detection using memory-augmented deep autoencoder. IEEE Access, 9, pp.104695-104706.