Journal of Theoretical and Applied Information Technology
30th June 2025. Vol.103. No.12

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

RESILIENCE GREY WOLF OPTIMIZATION-BASED CONVOLUTIONAL NEURAL NETWORK (RGWO-CNN) FOR CLASSIFICATION OF HEART DISEASE AND DIABETES (HDD)

S.USHA[1], R.AMEEN SHERIFF[2], S.KANCHANA[3]
[1]Department of Computer Applications, Faculty of Science and Humanities, SRM Institute of Science and Technology, Kattankulathur, Chennai, India.
[2]Department of Computer Applications, Ganadipathy Tulsi's Jain Engineering College, Vellore
[3]Department of Computer Science, Faculty of Science and Humanities, SRM Institute of Science and Technology, Kattankulathur, Chennai, India.
E-mail: [1]ushas6@srmist.edu.in, [2]ameensheriff_mca@gtec.ac.in, [3] kanchans@srmist.edu.in

ABSTRACT

Heart Disease and Diabetes (HDD) has reached epidemic proportions worldwide. Long-term issues affecting human organs are more likely due to HDD. If this critical medical condition is diagnosed at an early stage, people may be able to live healthier and longer lives by taking proper medications. The properly trained machine learning models on relevant datasets will help to diagnose HDD, but the significant issue is the lack of classification accuracy. Even more importantly, most current machine learning algorithms focus on predicting specific diseases. A classifier that accurately predicts the incidence of several diseases might be helpful in this context. This paper proposes a Resilience Grey Wolf Optimization-based Convolutional Neural Network (RGWO-CNN) to classify HDD. In the proposed RGWO-CNN method, the hyperparameters of the CNN model are represented as individual wolves in the RGWO algorithm. The position of each wolf represents a specific set of hyperparameters. The RGWO algorithm iteratively updates the position of each wolf based on their fitness (performance) and the wolves' social hierarchy (dominance). The updated position of each wolf corresponds to a new set of hyperparameters used to train and evaluate a new CNN model. This work uses RGWO-CNN to the Cardiovascular Disease Dataset and the PIMA Indian Diabetes Database to evaluate its performance. The assessment results reveal that the proposed classifier offers superior classification accuracy compared to the state-of-the-art classifiers.

Keywords: *Diabetes, Grey Wolf, Heart Disease, Neural Network, Optimization, Particle Swarm*

1. INTRODUCTION

Machine learning (ML) is an umbrella term for statistical approaches that let machines learn through experience without explicitly being programmed. Algorithm modifications are a common manifestation of this type of learning. ML software may learn to identify people in photos by analyzing thousands of images. There are two main categories of machine learning: unsupervised and supervised learning [1]. Healthcare is one of the most important fields to benefit from this breakthrough. Thanks to the progress in medicine and science over the previous century, individuals may hope to live for a very long time. New advances in areas such as AI and ML have the potential to bring about a medical revolution, even though technology has gone a long way. Naturally, computers allow for the simplification of even the minutest and least significant aspects of any activity. The healthcare business has been an early adopter of artificial intelligence (AI) and has potential future uses [2]. The healthcare sector has a long history of enthusiastically embracing innovative technology. Artificial intelligence and machine learning have found beneficial uses in the medical field, similar to those in other sectors, such as commerce and the online economy. The applications of such technology are practically endless. Innovative uses of ML are helping to revolutionize the healthcare sector. Due to required processes such as electronic medical records, medical practices have already begun utilizing big data technologies for next-generation predictive analytics. ML technologies are expected to improve this procedure even further[2,3]. These enhance the efficiency of intelligent and automated decision-

making in primary, secondary, and tertiary public healthcare settings. Since this can potentially enhance the lives of billions of people worldwide, it may be the most critical effect of ML technologies[].

The applications of ML technology in clinical trial research are many. Medical personnel might save time and money by evaluating a larger volume of data with the help of powerful predictive analytics applied to clinical trial candidates. Many ML applications, such as those that help determine the most effective sample sizes and those that make it possible to use electronic health records (EHRs) with a minimum of data entry mistakes, can further enhance the productivity of clinical trials. As skilled radiologists become scarcer worldwide, this strategy solves a significant challenge facing the healthcare industry[6,7]. Healthcare practitioners may better accommodate patients' needs and improve the efficacy of their treatment plans by combining personal health data with predictive analytics. The potential applications of ML in research and medical testing are vast. Researchers employing ML-based predictive research to find hidden clinical trial participants may move with a supply from many data sources, such as previous medical visits, social media, etc.[8]. It also takes care of the people taking part in the trial. It ensures that the data is available in real-time so that researchers may study an adequate number of people and make less statistically significant mistakes thanks to the efficiency of computerization. Different algorithms can be employed to analyze the growing digital archive of medical imaging data for trends and anomalies. ML systems may analyze imaging data as a highly-skilled radiologist would to detect skin lesions, cancers, and brain hemorrhages. Consequently, it is reasonable to expect a substantial uptick in radiologists using such services[9, 10].

1.1. Problem Statement

The National Patient Safety Foundation found that 42% of hospitalized patients had delayed diagnoses or medical mistakes. Ignorant approaches to patients' medicines, medical tests, and procedures pave the way for misdiagnosis. The massive amounts of data may be used to learn more about the patterns or rules underlying machine learning techniques. Because of the importance of disease diagnosis in data processing, many different types of patient testing are required leading to more expenses. Experiments may be conducted more quickly and efficiently with the help of data mining tools. When diagnosing HDD, data mining is essential because it allows doctors to consider the traits and features—such as age, weight, etc. The most relevant to the problem at hand may have errors. Even if data mining is used in the healthcare industry, the accuracy of the different ways of putting things into groups must be considered.

1.2. Objective

To better diagnose and predict heart disease and Diabetes (HDD), this research work has set the primary objective to propose Tremendous Hybrid Optimization-based Machine Learning Algorithm (THO-MLA), which utilizes the natural characteristics inspired by particles (i.e., birds) and wolves.

2. LITERATURE REVIEW

This section categorizes the relevant material as either (1) algorithms for classifying heart disease or (2) algorithms for classifying diabetes.

2.1 Heart Disease Classifiers

The “Phonocardiogram Prediction Model” [11] is the prospect of predicting ischemic heart disease. Savitzky-Golay filtering is used alongside deep learning and ensemble learning algorithms. Its ultimate goal is to enable fully automated prediction. Both (a) ten-fold cross-validation and (b) no testing is used to evaluate the classification accuracy. “Early Prediction of Intensive Care Unit (EPICU)” [12] is proposed to predict the readmission of patients in the ICU accurately. The expected outcome of an EPICU procedure is one of two possible outcomes. ForecastingICU readmission employs eight distinct categorization techniques. According to the results, classification after feature selection performed better than straight classification. “Effective Heart Disease Prediction” [13] is proposed to assist the clinician in predicting cardiovascular disease. It provides a roadmap to make decisions and predictions with a large amount of data produced in the healthcare industry. A hybrid random forest is an ensemble with linear regression to perform the classification. “Automated Prediction” [14] is proposed to predict heart disease by utilizing sparse discriminant analysis (SPA). It aims to minimize the time consumption by the optimum linear discriminant analysis (LDA) scoring methodology, resulting in deep execution. If classes or sub-classes are non-linear, deep execution continues until the best result is achieved. “Real-Time Heart Disease Prediction” [15] is proposed to utilize Apache Spark to send the predicted heart diseases to the clinician at the earliest. To avoid delayed streaming in the

distributed computing environment, apache-park is utilized. It involves streaming processing, storing data, and visualizing the obtained results.

"Cloud-based Internet of Medical Things Framework" [16]is proposed to analyze large datasets produced by medical sensing equipment to diagnose and predict cardiovascular disease. Cloud computing is utilized to effectively utilize resources to serve individuals better and make them take care of when they are at risk of cardiovascular disease. "Heart Sound Classification" [17] is proposed to predict heart-related diseases. The signal energy is utilized for the analysis. It has deployed a stacked encoder network for the classification task and evaluated its efficacy using PASCAL B training data. The results are compared across many datasets to demonstrate the suggested study's effectiveness. "Deep Learning based Disease Prediction" [18] is proposed to predict the diseases related to Alzheimer's. Data extraction is performed with the available ambulatory EMR data, which is heterogeneous. Naïve Bayes methodology is integrated with deep learning to enhance the prediction ratio. "Classification Using Link Prediction" [19] was proposed to detect the unlabeled data in the class. Link predictors were developed and used to categorize various datasets. Multiple link predictors are identified using a majority vote approach that takes advantage of low-level properties. "Hierarchical Neighborhood Component-Based Learning and Adaptive Multi-Layer Networks" [20]predicts future risks of heart failure is offered. Learning occurs at the attribute level, and the neighborhood component module is used to construct the core features. Finding the potential danger helps global weight vectors.

2.2 Diabetes Classifiers

"Type 2 Diabetic Mellitus detection framework" [21] is proposed with the help of a Genome-Wide Association Study ($GWAS$) along with the Phenome-Wide Association Study ($PheWAS$). To measure the effectiveness of the structure, we turn to machine learning techniques. A greater control identification rate is demonstrated using a training and test set of 300 samples. "Structured Framework" [22] is proposed to classify and extract blood vessels affected by diabetes. The framework removes the source image's background and improves the blood vessel pixel's background for adequate classification. Feature vectors are generated and extracted using deep learning-based classification strategies. Features were trained using firefly optimization. "Automatic detection with Extreme Learning" [23] is proposed for detecting diabetic retinal image neovascularization, and filter banks' sequence is used to acquire its features. The framework was evaluated with the annotated images from the ophthalmologist's expert. The result shows that the framework clearly shows the alleged neovascularization regions and can sustain decision-making.

"Efficient Ensemble Strategy" [24] is proposed to predict Type 2 diabetics, which applies SVM for classification. The data selection method uses two phases to select the data points. Ensemble SVM is constructed by selecting the data points. It analyses the system's performance using real-world datasets and finds that it outperforms state-of-the-art models with significantly less training time. "Polynomial– Kernel SVM-based Method" [25]is created to categorize persons with diabetes at a high risk of complications. Age, sex, body mass index, and waist size are all considered. To address the nuances between poor glucose tolerance and the other markers of Type 2 diabetes, we turned to a logistic Regression-based method.

A "Smart Health Monitoring System" [26]idea is to check the patient's blood pressure and glucose level and report back to the doctor if there are any discrepancies. Predictions of hypertension and diabetes risk are made using machine learning and decision-making techniques. The intuitive technology instantly notifies and alerts medical professionals to any changes. "Neural Network-based Model" [27] was proposed to analyze diabetic patients to recognize the challenges in the existing methods. An advanced technique was implemented to execute an analytical performance on diabetic-affected people, their age group, working style, and food habits. The life span of diabetic patients can also be detected. This model aid as a guide in resolving other diseases like cardiovascular problems, retinopathy, and nephropathy, which are caused due to diabetics. Pima Indian Diabetic datasets were used for the evaluation. "Model for Finding Diabetics" [28] is proposed to classify diabetes among pregnant ladies. It applies machine learning methodologies, namely SVM, Naïve Bayes, and Decision Tree. The model is tested with Pima Indian Diabetes Dataset. The results show that the Naive Bayes performs with increased accuracy and is examined with ROC curves.

"Identifying blood-based Gene Expression Score" [29] is proposed for predicting diabetes and coronary artery disease. The dataset of 110 patients was used to identify the genes bioinformatically and are evaluated statistically with the SVM method. The GES performance was experimentally tested with suspected CAD. "Diabetic Mellitus Detection" [30] is proposed by extracting features of photoplethysmography with PPG signal consideration. The hybrid feature selection approach was used to extract 37 features and reduce noisy data. System performance is enhanced together with ten hand-picked feature sets. Diabetic patients can be treated and protected from end-organ damage using a screening tool validated by a random population test.

"Modified Support Vector Machine" (MSVM)"[31]is proposed to aid in the diagnosis of diabetes. To do this first processing, MSVM employs a variant of the principal component analysis (PCA) method to extract features from the provided medical data. After computing the covariance matrix's eigenvectors using MPCA, data mapping into a subspace with the same dimensionality is done. MSVM's goal is to choose the best characteristics for further categorization. Classifications and predictions are based on the most useful recoverable characteristics. A tailored method is used to choose the best hyperplane, yielding the most significant achievable separation margin that still satisfies the criteria for each categorization. This classification may reduce the experimental categorization error in linear and non-linear data analysis. Finding the hyperplane of greatest distancing allows the input vector to be projected onto a higher-dimensional space. "Deep Learning for Predicting Diabetes (DLPD)"[32] is a strategy for making predictions about diabetes by tapping into the capacity of deep neural networks' hidden layers. To prevent overfitting, dropout regularisation is employed. DLPD does this via a binary cross-entropy loss function whose parameters are optimized for the given task. The model's prior learnings can be preserved, and its adaptability to new data may be improved using normalization layers. After each training update, dropout randomly returns some proportion of units to zero. A constant scaling factor is applied to the hyperparameters to account for past gradients. We can write the whole gradient through recursion as the mean damping of all the square gradients that came before it. Time spent classifying data has little effect on anything other than the mean and gradient at the moment.

The proposed work [33]uses hybrid techniques to combines two different algorithms to enhance the accuracy of heart disease and diabetes. It begins with selection of most significant features from given dataset. It used Ant Colony optimization techniques. This algorithm finds the shortest path between two sites by imitating the behaviour of ants, which is an inspiration from nature. The factors that have the most effects on prediction accuracy are found using the Ant Colony Optimisation (ACO) approach. The approach use Adaptive Boosting (AdaBoost) to construct a classification model based on these features after identifying the most important components. In AdaBoost, multiple weak classifiers are combined to create a robust classifier. An AdaBoost model is trained based on selected features to predict whether a patient has heart disease or diabetes. ACO is used again to select the essential elements and Adaboost is used to train a model until the desired prediction accuracy is reached. A comparison of the prediction accuracy and feature selection of RACOAA with other state-of-the-art algorithms has shown that RACOAA is superior to other state-of-the-art algorithms.

"Self-Adaptive Particle Swarm Optimization-based Random Forest Algorithm (SAPSORF)" [34] designed to improve the diagnosis of heart disease and diabetes (HDD). This classifier aims to address the limitations of current machine learning models in accurately predicting these diseases, which are often designed to focus on specific health conditions and thus lack versatility. By integrating a bio-inspired algorithm, Self-Adaptive Particle Swarm Optimization (SAPSO), with a modified Random Forest Algorithm, the SAPSORF enhances both the sampling and dimensionality reduction phases of the Random Forest model, leading to improved performance.The effectiveness of SAPSORF is evaluated using two distinct datasets: the Cardiovascular Disease Dataset and the PIMA Indian Diabetes Dataset. The results from these evaluations demonstrate that SAPSORF outperforms existing classifiers in terms of accuracy for these disease classification tasks. This indicates that SAPSORF is a promising tool for the early and accurate risk assessment of cardiovascular disease and diabetes, potentially contributing to better prevention and treatment strategies for these lethal conditions. Optimization algorithms are best in achieving the expected outcomes [32]-[70].

3. RESILIENCE GREY WOLF OPTIMIZATION-BASED CONVOLUTIONAL NEURAL NETWORK (RGWO-CNN)

3.1. Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) begins with the Input layer. The task of the input layer is to transform the unprocessed data into a form that the neural network can understand. Data embeddings commonly encode the input text, where a fixed-length vector denotes each word.

3.1. Layer of Embedding

At the embedding layer, a dense vector of a predetermined length represents the meaning of each word in the input phrase. Let's say we have an input sequence S of length L and a vocabulary size of V. The input sequence is first transformed into a sequence of one-hot vectors of size V, where each vector has a value of 1 at the index corresponding to the word in the vocabulary and 0 elsewhere. The one-hot vectors are then multiplied with a weight matrix W_{emb} of size $V\ x\ E$, wherein E is the dimensionality of the embedding space. To achieve optimal performance, the weight matrix is learned during training.

Let $S = (S_1, S_2, \dots, S_L)$ be an input sequence of length L, where each s_i is an index into the vocabulary of size V. Let W_{emb} be the weight matrix of size $V\ x\ E$, wherein E is the dimensionality of the embedding space. Then the matrix of word embeddings W_{emb}can be obtained using Eq.(1).

$$W_{emb} = S * W_{emb} \quad (1)$$

where * denotes matrix multiplication.

3.1.2 Convolutional layer

Convolutional layers are the backbone of CNN when collecting relevant features from input data. Convolutional layers produce feature maps highlighting basic patterns or structures in the data by applying filters to the input data learned during training.

3.1.2.1 Convolution Operation

To execute a convolution, a filter is moved over the input data. In contrast, the threshold value among the filter and a localized piece of the inputs is calculated at each location. The mathematical definition of the convolution operation is mathematically expressed as Eq.(2).

$$y_{s,w} = \sum_m \sum_n p_s + m, j + n. w_{m,n} + b \quad (2)$$

The input data is x, the filter is w, the bias term is b, and the final feature map is y. Output feature map coordinates are indicated by indices i and j, and filter coordinates by indices m and n.

3.1.2.2 Multiple Filters

Convolutional layers typically use multiple filters to learn multiple feature maps simultaneously. Each filter is initialized randomly and updated during training to learn useful patterns or structures in the input data. By convolving with the input data, each filter produces a separate feature map, and the sum of all feature maps produced scales linearly with the number of filters employed.

3.1.3 Stride and Padding

To modify the output feature map's spatial dimensions, convolutional layers let the setting of stride and padding parameters. Stride, which specifies how many pixels the filter moves by at each step to lower the output feature maps' spatial dimensions, can be adjusted to a value higher than 1. Padding involves adding extra pixels around the input data to preserve its spatial dimensions after convolution.

3.1.2. Softmax Activation Function

Convolutional neural networks often use the softmax activation function at the result layer for multi-class classification issues. A vector of real-valued inputs is utilized to compute a probability distribution across the classes. The definition of the same is given in Eq.(3).

$$Softmax(p_s) = exp(p_s)/sum_j \left(exp(p_j)\right) \quad (3)$$

where $sum_j \left(exp(p_j)\right)$ is the sum of all the input values' exponential values and p_s is the input for the i-th class. The softmax function is appropriate for classification problems based on probability since it guarantees that the output values are around 0 and 1 and add up to 1.

To simplify the feature maps produced by the convolutional layers, pooling layers are utilized in neural networks. By teaching the network to be translationally invariant, pooling allows it to recognize the same pattern no matter where it appears in the input.

Let's consider a convolutional layer with input feature map X of size $W\ x\ H\ x\ C$, where W is the width, H is the height, and C is the number of channels. We apply a pooling operation with kernel

size k and stride s, which outputs a new feature map Y of size $(W-k)/s+1 \times (H-k/s+1 \times c)$.

Max pooling is often utilized in CNN. The greatest value in a k-by-k square region of the input feature map is extracted using max pooling. Eq. (4) is a mathematical representation of this idea.

$$Y(i,j,c) = max\big(X(is,js,c)\big)\ for\ is = is\ to\ (is)+k-1\ and\ js = js\ to\ (js)+k-1 \quad (4)$$

where $Y(i,j,c)$ is the output value at position (i,j) in channel c, and $X(is,js,c)$ is the input value at position (is,js) in channel c.

3.1.4. Layering Pools

Neural networks use pooling layers to make the feature maps generated by the convolutional layers more spatially manageable. The goal of pooling is to introduce a degree of translation invariance into the network, meaning that the network can recognize the same pattern regardless of its position in the input. A pooling layer typically operates on a 3D tensor of shape (W,H,C), where W is the width, H is the height, and C is the number of channels. Let X be the input tensor to the pooling layer, and let Y be the output tensor. The pooling operation is typically defined by a pooling function and two hyperparameters: the pooling size k and the stride s.

The pooling size k determines the size of the pooling window. Maximum pooling (which returns the largest value) and average pooling (which returns the average values in the pooling window) are the most common choices for the pooling function. The stride s determines the step size for moving the pooling window across the input tensor.

More formally, let $X(i,j,c)$ be the element of the input tensor at position (i,j) in channel c. Let $Y(i',j',c)$ be the element of the output tensor at position (i',j') in channel c. Hence, this research work characterizes the pooling process as follows:

Max pooling:

$$Y(i',j',c) = max_{\{p=0\}}^{\{k-1\}} max_{\{q=0\}}^{\{k-1\}} X(i'*s+p, j'*s+q, c) \quad (5)$$

Average pooling:

$$Y(i',j',c) = (1/k^2) * sum_{\{p=0\}}^{\{k-1\}} sum_{\{q=0\}}^{\{k-1\}} X(i'*s+p, j'*s+q, c) \quad (6)$$

where the indices p and q iterate over the pooling window. Note that in the case of average pooling, we divide by k^2 to ensure that the output values are normalized.

Pooling layers can also be applied globally, meaning that the pooling window covers the entire spatial dimensions of the input tensor. The last stage of a convolutional neural network often employs global pooling to provide a fixed-length output suitable for feeding into a fully connected layer. This is how we may define the maximum global pooling and global average pooling processes:

Global max pooling:

$$Y(c) = max_{\{i=0\}}^{\{W-1\}} max_{\{j=0\}}^{\{H-1\}} X(i,j,c) \quad (7)$$

Global average pooling:

$$Y(c) = \big(1/(W*H)\big) * sum_{\{i=0\}}^{\{W-1\}} sum_{\{j=0\}}^{\{H-1\}} X(i,j,c) \quad (8)$$

where the indices i and j iterate over the spatial dimensions of the input tensor.

3.1.5. Backpropagation

During training, the fully linked layer's weights and biases are learned via backpropagation. Given a loss function L, the loss gradient concerning the layer's output can be computed as Eq.(9).

$$dL/dy = grad \quad (9)$$

where grad describes the loss's gradient at the layer's output.

The weight and biases loss gradient can be computed using the chain rule as Eq.(10).

$$dL/dW = dL/dy * dy/dW \quad dL/db = dL/dy * dy/db \quad (10)$$

where dy/dW and dy/db are the derivatives of the layer output concerning the weights and biases, respectively.

After then, gradient descent or another optimization technique may be used to adjust the weights and biases.

3.1.6. Binary Cross-Entropy Loss

The binary cross-entropy loss function is frequently employed for issues involving only two classes. The desired label for the current input is y.

Therefore, we'll treat this as a binary classification issue. y_{ha} is the positive class probability we expect. The definition of the binary cross-entropy loss function is as follows. (11).

$$L = -y * log(y_{hat}) - (1 - y) * log(1 - y_{hat}) \quad (11)$$

where y is the actual label andy_{hat} is the expected proportion of successes.

The training seeks to achieve a minimum of this loss function. Backpropagation may be used to calculate the loss gradient concerning the model parameters.

3.1.7. Categorical Cross-Entropy Loss

For issues with several classes, the categorical cross-entropy loss function is applied. Let's consider a multi-class classification problem where the target label for a given input is a one-hot encoded vector y of length k. The predicted probabilities of the k classes are given by the vector $y_{hat}$of length k. We may characterize the loss function of the cross-entropy category as Eq.(12).

$$L = -sum(y * log(y_{hat})) \quad (12)$$

where y is the true label and $y_{hat}$is the predicted probabilities of the k classes.

The training seeks to achieve a minimum of this loss function. Backpropagation may be used to calculate the loss gradient concerning the model parameters.

3.1.8. Stochastic Gradient Descent

By sampling from the training data at random, stochastic gradient descent updates the parameters based on the gradient of the reduction function. This method is more computationally efficient when calculating the gradient across the complete training set. Stochastic gradient descent's updating rule is defined as Eq.(13).

$$theta = theta - learning\_rate * gradient\_minibatch \quad (13)$$

where theta represents the model's parameters, the learning rate is a hyperparameter that controls the size of the updates, and gradient minibatch is the gradient of the loss with the parameters in mind, computed over a sample of the training data at random.

3.1.9. Hyperparameter Tuning

Hyperparameters are the configuration parameters that determine the behavior and performance of a neural network. Examples of hyperparameters in a CNN for HDD classification include the learning rate, batch size, number of epochs, number of filters, kernel size, etc. Tuning these hyperparameters is essential for achieving the best performance of the model. Hyperparameter tuning can be done manually, by trial and error, or by using automated methods such as grid search, random search, or Bayesian optimization. In manual tuning, the hyperparameters are varied individually, and the model's performance is evaluated each time. In automated tuning, a search space of hyperparameters is defined, and the algorithm tries to find the best combination of hyperparameters within that space.

Hyperparameter tuning can be represented as an optimization problem, where the objective is to find the set of hyperparameters that maximizes the performance metric of the model. Let's denote the hyperparameters as θ and the performance metric as $J(\theta)$. Then, we can represent the hyperparameter tuning process as follows:

Algorithm 1. Hyperparameter Tuning

Step 1: Define a search space for the hyperparameters θ
Step 2: Initialize a set of hyperparameters θ_0
Step 3: Train the model with the hyperparameters θ_0
Step 4: Evaluate the performance of the model using the performance metric $J(\theta)$
Step 5: Update the set of hyperparameters $\theta_1$based on the evaluation result
Step 6: Repeat steps 3-5 until the performance metric converges or a maximum number of iterations is reached

The optimization problem can be solved using various optimization algorithms, such as gradient descent, Adam, or RMSprop. The choice of optimization algorithm also affects the model's performance; thus, it is also a hyperparameter that needs to be tuned. Hyperparameter tuning is a crucial step in developing a CNN for HDD classification. It helps to find the best set of hyperparameters that maximizes the model's performance and improves its generalization ability.

3.2. Resilience Grey Wolf Optimization

In CNN, Stochastic Gradient Descent is an optimization strategy to attain the results. But, Stochastic Gradient Descent's Optimization is insufficient to achieve better classification accuracy

towards HDD classification and prediction. Hence, this research utilizes bio-inspired optimization, namely Resilience Grey Wolf Optimization.

3.2.1. Initialization of Population

The initialization of the population in the GWO algorithm involves randomly initializing a population of N wolves within the search space, where each wolf is represented as a D-dimensional vector. The initialization process should ensure wide search space coverage to promote solution space exploration.

Let the population size be N, and the search space be represented as D-dimensional vectors $x = [x1, x2, \ldots, xD]$. Then, the population of wolves can be represented as a matrix X of size NxD, where each row represents a candidate solution or wolf. The same is expressed in Eq.(14).

$$X = [x1;\ x2;\ \ldots;\ xN] \quad (14)$$

The wolves are first placed in the search space at random. Let's say lb and ub are the bottoms and upper limits of the search space. Next, we generate an initial wolf location xi according to the equation Eq (15).

$$xi = lb + (ub - lb) * rand(D, 1) \quad (15)$$

where $rand(D, 1)$ generates a random D-dimensional vector between 0 and 1.

The initial population should cover a wide range of the search space to ensure a good exploration of the solution space. The initialization process should also ensure that the wolves are not initialized too close to each other, as this can lead to premature convergence to local optima. Therefore, a random initialization strategy that covers a wide range of the search space is preferred.

3.2.2. Fitness Function Evaluation

Each solution in the population is given a fitness value and ranked according to the GWO algorithm's fitness function assessment step. ALP, BTA, and DTA wolves are chosen according to their fitness scores, which direct the search for optimal solutions. The fitness function measures how well a solution meets optimization requirements; it should be tailored to the situation at hand.

Consider the fitness function to be $f(x)$, which ranks the effectiveness of a set of potential answers. Next, the fitness value $f(xi)$ is calculated for each wolf in the population using Eq.(16).

$$fit_i = f(xi) \quad (16)$$

ALP, BTA, and DTA wolves are chosen based on their fitness ratings, which indicate the relative quality of each solution in the population. The wolf with the highest fitness value (the ALP wolf) is chosen as the leader, while the wolf with the 2nd-highest fitness value (the BTA wolf) and the wolf with the 3rd-highest fitness value (the DTA wolf) are given secondary roles in the pack.

Let fit_{ALP}, fit_{BTA}, and fit_{DTA} be the fitness values of the ALP, BTA, and DTA wolves, respectively. The following selection rule provided in Algorithm 2 is used.

Algorithm 2. Fitness Evaluation-based Selection

Step 1:	if $fit_i < fit_{ALP}$
Step 2:	$elif fit_{ALP} < fit_i < fit_{BTA}$
Step 3:	$elif fit_{BTA} < fit_i < fit_{DTA}$

where ALP, BTA, and DTA are the indices of the alpha, beta and delta wolves, respectively.

The fitness function determines how well a solution meets optimization requirements; it should be tailored to the situation. Reducing or maximizing the fitness function is the objective of optimization issues. The fitness function's return value should be a single numeric value that indicates the solution's quality. It is also possible for the fitness function to be multi-objective, intending to optimize several criteria simultaneously.

3.2.3. Identification of ALP, BTA, and DTA Wolves

The RGWO algorithm determines the ALP, BTA, and DTA wolves by choosing the three wolves with the highest fitness scores. These pack leaders steer the pack in its quest for improved solutions, and the pack's collective knowledge is updated based on their latest observations. The fitness levels of the wolves in the population serve as the basis for the selection rule used to determine the ALP, BTA, and DTA wolves.

Let the fitness levels of the ALP, BTA, and DTA be fit_{ALP}, fit_{BTA}, and fit_{DTA}, respectively. The fitness of the ALP wolf is the highest. Hence its location is labeled x_{ALP}. Similarly, x_{BTA} and x_{DTA} represent the BTA and

delta wolves, who rank second and third in fitness. Wolves in a population are ranked as " ALP," " BTA," and "DTA" according to their fitness levels. ALP, BTA, and DTA wolves are determined using the selection procedure presented in Algorithm 3.

Let fit_{ALP}, fit_{BTA}, and fit_{DTA} be the fitness values of the ALP, BTA, and DTA wolves, respectively. The ALP wolf has the best fitness value, and its position is denoted as x_{alpha}. Similarly, the BTA and DTA wolves have the second-best and third-best fitness values, respectively, and their positions are denoted as x_{beta} and x_{delta}, respectively. The identification of the ALP, BTA, and DTA wolves is based on the fitness values of the wolves in the population. The selection rule provided in Algorithm 3 is used to identify the ALP, BTA, and DTA wolves.

Algorithm 3. Selection Rule Used to Identify Wolves

Step 1: for $i = 1$ to N:
Step 2: if $fit_i < fit_{ALP}$:
Step 3: DTA = BTA
Step 4: BTA = ALP
Step 5:
Step 6: $elif fit_{ALP} < fit_i < fit_{BTA}$
Step 7: DTA = BTA
Step 8:
Step 9: $elif fit_{BTA} < fit_i < fit_{DTA}$
Step 10:

where ALP, BTA, and DTA are the indices of the Alpha, Beta, and Delta wolves, respectively.

Leaders, such as the ALP, BTA, and DTA wolves, direct the pack in its quest for improved strategies. The position of the alpha wolf is used to revise the locations of the other wolves in the pack, as it is the best answer so far.

3.2.4. Updating the Position of Candidate Solutions

Each potential solution (wolf) location is updated in GWO based on the locations of ALP, BTA, and DTA wolves. Let x_i be the location of the ith wolf while x_{ALP}, x_{DTA}, and x_{DTA} denote the locations of the ALP, BTA, and DTA wolves.

3.2.5. Exploration and Exploitation of Search Space

The GWO algorithm's exploration-exploitation trade-off is modulated by two parameters: exploration rate and exploitation rate. As the pace of exploration increases, the BTA, and DTA wolves become more influential in the position update of each possible solution. In contrast, the exploitation rate increases the alpha wolf's impact. Overhead for RGWO-CNN is provided by Algorithm 4.

Algorithm 4. RGWO-CNN

Step 1: Initialize the population of grey wolves with random solutions.
Step 2: Find each grey wolf's fitness level depending on how well its matching CNN model performed.
Step 3: Identification of Alpha, Beta, and Delta Wolves:
Step 3.1: Rank the grey wolves according to their fitness and select the alpha, beta, and delta grey wolves, the three best solutions.
Step 3.2: Updating the Position of Candidate Solutions:
Step 3.3: Update the position of each grey wolf using the RGWO algorithm, which involves computing the new position based on the current position, the position of the alpha, beta, and delta wolves, and a step size parameter.
Step 3.4: Perform the Exploration phase and Exploitation Phase in the available Search Space:
Step 3.5: Generate a new CNN model for each grey wolf based on its updated position.
Step 4: Convert the input data into a matrix of embedding vectors, representing the input features in a lower dimensional space.
Step 5: Using a convolution operation and a filter set, extract useful features from the input data by multiplying the matrix of embeddings by the filter set.
Step 6: To normalize the data and translate them into a probability distribution over multiple classes, the output of the convolutional layer is given a softmax activation function.
Step 7: Convolutional layer outputs can be downsampled, and their dimensionality reduced by applying a pooling procedure.
Step 8: The backpropagation technique will compute the loss function gradients as a function of the CNN's parameters.
Step 9: When the output has two categories, use the binary cross-entropy loss function to determine the loss.
Step 10: The loss should be calculated using the categorical cross-entropy loss function

when the output has more than two categories.

Step 11: Tune the hyperparameters of the CNN and RGWO algorithms to achieve the best performance on the validation set.

Step 12: Repeat steps 2-14 for a predetermined number of iterations or until a convergence criterion is met.

Step 13: Select the grey wolf with the best fitness as the final solution.

Step 14: Return the final CNN model corresponding to the best grey wolf as the trained model.

4. ABOUT DATASET AND PERFORMANCE METRICS

4.1 Dataset

The Cardiovascular Disease dataset and the PIMA Indians Diabetes dataset, found on the Kaggle website and downloaded for free, are used to evaluate the suggested classifier's performance. The cardiovascular Disease dataset holds 70000 records, and the PIMA Indians Diabetes dataset holds 768 records. Datasets are described in Table 1 and Table 2.

Table 1. Cardiovascular Disease Dataset

Feature	Feature Type	Description
Age	Objective	Age of Patient
Height	Objective	Height of Patient in cm
Weight	Objective	Weight of Patient in kilogram (kg)
Gender	Objective	Gender of Patient. 1 indicates Male, 2 indicates Female
Systolic Blood Pressure	Examination	The sudden influx of blood brings on pressure into the arteries. It is measured in millimetres of mercury (mmHg).
Diastolic Blood Pressure	Examination	Arterial blood pressure refers to the force exerted by the blood on the walls of the arteries in between heartbeats. It is measured in millimetres of mercury (mmHg).
Cholesterol	Examination	Cholesterol present in patient blood is indicated in 3 different levels, which are: 1 indicating normal level, 2 indicating above the normal level, 3 indicating well above the normal level
Glucose	Examination	Glucose present in patient blood is indicated in 3 different levels, which are: 1 indicating normal level, 2 indicating above the normal level, 3 indicating well above the normal level
Smoking	Subjective	Whether the patient has the habit of smoking or not.
Alcohol Intake	Subjective	Whether the patient consumes alcohol or not.
Physical Activity	Subjective	Whether the patient does any physical activity or not.
Cardiovascular Disease Presence/Absence	Subjective	Whether the patient has the disease of Cardiovascular or not

Table 2. PIMA Indians Diabetes Dataset

Feature	Feature Type	Description
Preg	Subjective	Indicates the number of times the patient got pregnant.
Gluc	Examination	Glucose is present in the patient blood.
Diastolic Blood Pressure	Examination	Arterial blood pressure refers to the force exerted by the blood on the walls of the arteries in between heartbeats.
Triceps Skin Fold Thickness	Objective	Indicates tricep thickness of skin fold.
Insulin	Examination	Insulin serum for 2 hours.
Body Mass Index	Examination	A medical screening tool that compares the height and weight of the patient to identify the presence of fat.
Diabetes Pedigree Function	Subjective	Determines whether or not a patient has a high risk of developing diabetes depending on their family.
Age	Objective	Age of Patient.
Diabetes Presence/Absence	Subjective	Whether the patient has the disease of diabetes or not

4.2 PERFORMANCE METRICS

- **Classification Accuracy:** It's a statistical evaluation of the classifier's efficiency in making predictions about unknown classes of unknown vectors.
- **F-Measure:** It's a way to evaluate the efficacy of an algorithm as a whole by considering both its precision and recall.
- **Matthews Correlation Coefficient:** It helps evaluate the accuracy of binary classifications.

- **Fowlkes-Mallows Index:** It is a measure to identify how closely two clusters of points are related. It is the bilateral precision and recall average expressed as a geometric mean.

The four variables involved in the metrics mentioned above calculation, which are True Positive ($TrPst$), False Positive ($FlPst$), True Negative ($TrNgt$), False Negative ($FlNgt$):

- $\boldsymbol{TrPst}$ = Count of medical cases exactly diagnosed as the patient.
- $\boldsymbol{FlPst}$ = Count of medical cases inexactly diagnosed as the patient.
- $\boldsymbol{TrNgt}$ = Count of medical cases exactly diagnosed as healthy.
- $\boldsymbol{FlNgt}$ = Count of medical cases inexactly diagnosed as healthy.

5. RESULTS AND DISCUSSION

5.1Analysis of TP, TN, FP, FN

Each model's accuracy in making predictions is shown by its True Positive, False Positive, True Negative, and False Negative numbers, respectively. These numbers may judge the accuracy of binary classification models.

Table 3. TP, FP, TN, and FN Result Values

Algorithms	True Positive	False Positive	True Negative	False Negative
BSVM	30.813	20.812	26.395	21.980
S-ANN	34.593	17.328	29.667	18.411
RGWO-CNN	49.151	2.860	44.899	3.090

The BSVM model has the lowest TP value of 30.813, indicating that it is ineffective at correctly identifying positive instances. It has a higher FN value of 21.980, meaning it incorrectly classifies positive instances as negative. The model also has a relatively high FP value of 20.812, indicating a higher tendency to classify negative instances as positive, reducing its specificity. However, the TN value of 26.395 indicates that the model correctly identifies negative instances.

The S-ANN model performs better than the BSVM model, with a higher TP value of 34.593 and a lower FN value of 18.411. It is better at correctly identifying positive instances than the BSVM model. The S-ANN model also has a lower FP value of 17.328, indicating that it is better at correctly identifying negative instances than the BSVM model. Additionally, the TN value of 29.667 indicates that the model effectively identifies negative instances.

The RGWO-CNN model performs the best out of the three models, with the highest TP value of 49.151 and the lowest FN value of 3.090. This indicates that it is very effective at correctly identifying positive instances and has a low tendency to classify positive instances as negative incorrectly. The model also has a very low FP value of 2.860, indicating that it is very effective at correctly identifying negative instances and has a very low tendency to classify negative instances as positive. The TN value of 44.899 indicates that the model is also effective at correctly identifying negative instances.

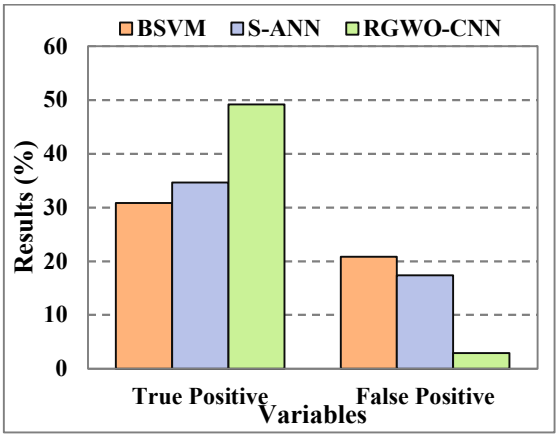

Figure 1a. TP and FP Analysis

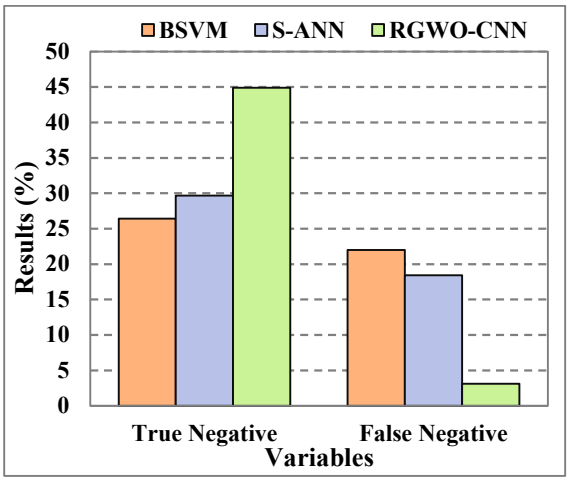

Figure 1b. TN and FN Analysis

Overall, the RGWO-CNN model outperforms the BSVM and S-ANN models regarding its TP, FP, TN, and FN values, indicating that it is the most effective at correctly identifying

both positive and negative instances. This suggests that the RGWO-CNN model is better suited for the given binary classification problem than the other two. The pictorial representation of Table 3 is as Figure 1.

5.2 Analysis of Classification Accuracy and F-Measure

The classification accuracy and F-measure scores in Table 4 indicate that the RGWO-CNN model performs best, followed by the S-ANN and BSVM models. The RGWO-CNN model outperforms the other two models regarding classification accuracy and F-measure. Both the classification accuracy and the F-measure of the RGWO-CNN model are much greater than those of the other two models, as shown in Figure 2. Concerning accuracy and F-measure scores, the S-ANN model ranks in the middle. Of the three models, BSVM yields the smallest estimates.

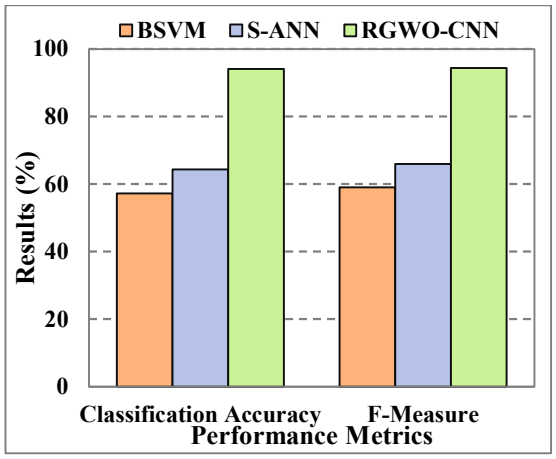

Figure 2. Classification Accuracy – F-Measure

Table 4. Result Values of Classification Accuracy-F-Measure

Algorithms	Classification Accuracy	F-Measure
BSVM	57.208	59.019
S-ANN	64.261	65.939
RGWO-CNN	94.050	94.292

The results of the three models are differentiated in Figure 2. The RGWO-CNN model outperforms the other two models in properly recognizing the class labels of the examples, as evidenced by its high classification accuracy and F-measure. The RGWO-CNN model performs best, whereas the S-ANN model performs well but not as well as the BSVM model. When evaluating the efficacy of various machine learning models, it is crucial to use reliable assessment measures, as shown in Fig 1. The graph shows that the RGWO-CNN model is superior to the other 2 models regarding classification accuracy and F-measure.

5.2 Analysis of Matthews Correlation Coefficient and Fowlkes-Mallows Index

In a binary classification issue, the Fowlkes-Mallows Index (FMI) calculates the degree of agreement between the predicted and true class labels. It is calculated as the average of the model's accuracy and recall scores, reflecting how well it recognises positive and negative examples and avoids making mistakes. An FMI of 1 implies a full agreement between the expected and true labels, whereas a value of 0 indicates no agreement.

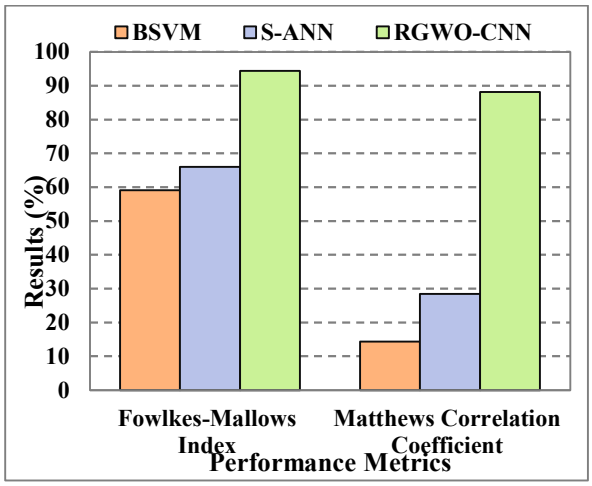

Fig 3. Matthews Correlation Coefficient and Fowlkes-Mallows

The accuracy of binary classification predictions may be evaluated using the Matthews Correlation Coefficient (MCC). It considers all possible outcomes (correct predictions, incorrect predictions, random predictions, and complete disagreement between predicted and true labels) and can take on values between -1 and 1. When comparing unbalanced and skewed datasets, MCC is often preferred to accuracy.

Table 4. Result Values of Matthews Correlation Coefficient and Fowlkes-Mallows

Algorithms	Fowlkes-Mallows Index	Matthews Correlation Coefficient
BSVM	59.023	14.264
S-ANN	65.942	28.363
RGWO-CNN	94.292	88.079

Table 4 shows that the RGWO-CNN model has the best FMI and MCC scores, suggesting that it has successfully predicted accurate binary labels with a high degree of resemblance to genuine labels. BSVM has the lowest FMI and MCC scores compared to other models, while S-ANN is in second place. The FMI and MCC values can be shown on a graph with the model names along the x-axis to see how they compare. As the graph shows, the RGWO-CNN model has the highest performance across both measures, followed by the S-ANN model and the BSVM model in that order. The RGWO-CNN model would perform far better than the other models, as depicted by the graph, demonstrating the model's dominance in this binary classification task. Table 4 is depicted graphically in Figure 3.

6. CONCLUSION

Compared to state-of-the-art approaches, the proposed Resilience Grey Wolf Optimization-based Convolutional Neural Network (RGWO-CNN) achieves notable improvements in categorizing Heart Disease and Diabetes (HDD). The experimental findings reveal that the RGWO-CNN approach outperforms other methods, such as BSVM and S-ANN, in classification accuracy, achieving an astonishing 94.050% accuracy. These findings demonstrate that the suggested approach is successful and efficient for categorizing medical data. The RGWO-CNN model's capacity to deal with noise and imbalance in the input data proves its robustness and resilience. This provides more evidence that the suggested technique for medical data categorization is trustworthy and accurate. In conclusion, the suggested RGWO-CNN method provides a promising strategy to categorize HDD data effectively and quickly. The additional study might examine whether the proposed approach holds for additional medical data sets and whether adding other optimization approaches would improve the model's performance.

REFERENCES

[1] S. Sandhiya and U. Palani, "An effective disease prediction system using incremental feature selection and temporal convolutional neural network," *J. Ambient Intell. Humaniz. Comput.*, vol. 11, no. 11, pp. 5547–5560, Nov. 2020, doi: 10.1007/s12652-020-01910-6.

[2] M. S. Nawaz, B. Shoaib, and M. A. Ashraf, "Intelligent Cardiovascular Disease Prediction Empowered with Gradient Descent Optimization," *Heliyon*, vol. 7, no. 5, p. e06948, 2021, doi: 10.1016/j.heliyon.2021.e06948.

[3] S. Asif, Y. Wenhui, Y. Tao, S. Jinhai, and H. Jin, "An Ensemble Machine Learning Method for the Prediction of Heart Disease," in *2021 4th International Conference on Artificial Intelligence and Big Data, ICAIBD 2021*, 2021, pp. 98–103. doi: 10.1109/ICAIBD51990.2021.9459010.

[4] M. Aslam and Jaisharma, "Hierarchical Random Forest Formation with Nonlinear Regression Model for Cardiovascular Diseases Prediction," in *2021 International Conference on Computer Communication and Informatics, ICCCI 2021*, 2021, pp. 1–5. doi: 10.1109/ICCCI50826.2021.9402571.

[5] V. Kedia, S. R. Regmi, K. Jha, A. Bhatia, S. Dugar, and B. K. Shah, "Time Efficient IOS Application for CardioVascular Disease Prediction Using Machine Learning," in *Proceedings - 5th International Conference on Computing Methodologies and Communication, ICCMC 2021*, 2021, pp. 869–874. doi: 10.1109/ICCMC51019.2021.9418453.

[6] W. Chang, Y. Liu, X. Wu, Y. Xiao, S. Zhou, and W. Cao, "A New Hybrid XGBSVM Model: Application for Hypertensive Heart Disease," *IEEE Access*, vol. 7, pp. 175248–175258, 2019, doi: 10.1109/ACCESS.2019.2957367.

[7] S. Goyal, J. M. Waghmare, and M. A. Ranjanikar, "Heart disease classification models from optical device-based electrocardiogram signals using machine learning algorithms," *Optik (Stuttg).*, vol. 271, p. 170176, 2022, doi: 10.1016/j.ijleo.2022.170176.

[8] L. Riyaz, M. A. Butt, and M. Zaman, "Ensemble Learning for Coronary Heart Disease Prediction," in *2022 2nd International Conference on Intelligent Technologies, CONIT 2022*, 2022, pp. 1–9. doi: 10.1109/CONIT55038.2022.9848292.

[9] B. Shiva Shanta Mani and V. M. Manikandan, "Heart disease prediction using machine learning," in *Handbook of Research on Disease Prediction Through Data Analytics and Machine Learning*, 2020, pp. 373–381. doi: 10.4018/978-1-7998-2742-9.ch018.

[10] J. M. T. Wu *et al.*, "Applying an ensemble convolutional neural network with Savitzky–

Golay filter to construct a phonocardiogram prediction model," *Appl. Soft Comput. J.*, vol. 78, pp. 29–40, 2019, doi: 10.1016/j.asoc.2019.01.019.

[11] M. Loreto, T. Lisboa, and V. P. Moreira, "Early prediction of ICU readmissions using classification algorithms," *Comput. Biol. Med.*, vol. 118, p. 103636, 2020, doi: 10.1016/j.compbiomed.2020.103636.

[12] S. Mohan, C. Thirumalai, and G. Srivastava, "Effective heart disease prediction using hybrid machine learning techniques," *IEEE Access*, vol. 7, pp. 81542–81554, 2019, doi: 10.1109/ACCESS.2019.2923707.

[13] K. M. Z. Hasan, S. Datta, M. Z. Hasan, and N. Zahan, "Automated Prediction of Heart Disease Patients using Sparse Discriminant Analysis," in *2nd International Conference on Electrical, Computer and Communication Engineering, ECCE 2019*, 2019, pp. 1–6. doi: 10.1109/ECACE.2019.8679279.

[14] A. Ed-Daoudy and K. Maalmi, "Real-time machine learning for early detection of heart disease using big data approach," in *2019 International Conference on Wireless Technologies, Embedded and Intelligent Systems, WITS 2019*, 2019, pp. 1–5. doi: 10.1109/WITS.2019.8723839.

[15] K. S. Adewole *et al.*, "Cloud-based IoMT framework for cardiovascular disease prediction and diagnosis in personalized E-health care," in *Intelligent IoT Systems in Personalized Health Care*, A. K. Sangaiah and S. B. T.-I. I. S. in P. H. C. Mukhopadhyay, Eds., Academic Press, 2020, pp. 105–145. doi: 10.1016/B978-0-12-821187-8.00005-8.

[16] O. Deperlioglu, "Heart sound classification with signal instant energy and stacked autoencoder network," *Biomed. Signal Process. Control*, vol. 64, p. 102211, 2021, doi: 10.1016/j.bspc.2020.102211.

[17] B. Ljubic *et al.*, "Influence of medical domain knowledge on deep learning for Alzheimer's disease prediction," *Comput. Methods Programs Biomed.*, vol. 197, p. 105765, 2020, doi: 10.1016/j.cmpb.2020.105765.

[18] S. A. Fadaee and M. Amir Haeri, "Classification using link prediction," *Neurocomputing*, vol. 359, pp. 395–407, 2019, doi: 10.1016/j.neucom.2019.06.026.

[19] O. W. Samuel *et al.*, "A new technique for the prediction of heart failure risk driven by hierarchical neighborhood component-based learning and adaptive multi-layer networks," *Futur. Gener. Comput. Syst.*, vol. 110, pp. 781–794, 2020, doi: 10.1016/j.future.2019.10.034.

[20] T. Zheng *et al.*, "A machine learning-based framework to identify type 2 diabetes through electronic health records," *Int. J. Med. Inform.*, vol. 97, pp. 120–127, Jan. 2017, doi: 10.1016/j.ijmedinf.2016.09.014.

[21] T. J. Jebaseeli, C. A. Deva Durai, and J. D. Peter, "Retinal blood vessel segmentation from diabetic retinopathy images using tandem PCNN model and deep learning based SVM," *Optik (Stuttg).*, vol. 199, p. 163328, Dec. 2019, doi: 10.1016/j.ijleo.2019.163328.

[22] H. Huang, H. Ma, H. JW van Triest, Y. Wei, and W. Qian, "Automatic detection of neovascularization in retinal images using extreme learning machine," *Neurocomputing*, vol. 277, pp. 218–227, Feb. 2018, doi: 10.1016/j.neucom.2017.03.093.

[23] S. Kang, P. Kang, T. Ko, S. Cho, S. J. Rhee, and K. S. Yu, "An efficient and effective ensemble of support vector machines for anti-diabetic drug failure prediction," *Expert Syst. Appl.*, vol. 42, no. 9, pp. 4265–4273, Jun. 2015, doi: 10.1016/j.eswa.2015.01.042.

[24] E. Longato *et al.*, "Glycaemic variability-based classification of impaired glucose tolerance vs. type 2 diabetes using continuous glucose monitoring data," *Comput. Biol. Med.*, vol. 96, pp. 141–146, May 2018, doi: 10.1016/j.compbiomed.2018.03.007.

[25] S. P. Chatrati *et al.*, "Smart home health monitoring system for predicting type 2 diabetes and hypertension," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 34, no. 3, pp. 862–870, Mar. 2022, doi: 10.1016/j.jksuci.2020.01.010.

[26] N. Pradhan, G. Rani, V. S. Dhaka, and R. C. Poonia, "Diabetes prediction using artificial neural network," *Deep Learn. Tech. Biomed. Heal. Informatics*, pp. 327–339, Jan. 2020, doi: 10.1016/B978-0-12-819061-6.00014-8.

[27] D. Sisodia and D. S. Sisodia, "Prediction of Diabetes using Classification Algorithms," *Procedia Comput. Sci.*, vol. 132, pp. 1578–1585, Jan. 2018, doi: 10.1016/j.procs.2018.05.122.

[28] X. bin Wang, N. hua Cui, X. Liu, and L. Ming, "Identification of a blood-based 12-gene signature that predicts the severity of coronary artery stenosis: An integrative approach based on gene network construction, Support Vector Machine algorithm, and multi-cohort validation," *Atherosclerosis*, vol. 291, pp. 34–

43, Dec. 2019, doi: 10.1016/j.atherosclerosis.2019.10.001.

[29] N. Nirala, R. Periyasamy, B. K. Singh, and A. Kumar, "Detection of type-2 diabetes using characteristics of toe photoplethysmogram by applying support vector machine," *Biocybern. Biomed. Eng.*, vol. 39, no. 1, pp. 38–51, Jan. 2019, doi: 10.1016/j.bbe.2018.09.007.

[30] S. Thenappan, M. Valan Rajkumar, and P. S. Manoharan, "Predicting Diabetes Mellitus Using Modified Support Vector Machine with Cloud Security," *IETE J. Res.*, pp. 1–11, Jul. 2020, doi: 10.1080/03772063.2020.1782781.

[31] H. Zhou, R. Myrzashova, and R. Zheng, "Diabetes prediction model based on an enhanced deep neural network," *Eurasip J. Wirel. Commun. Netw.*, vol. 2020, no. 1, p. 148, 2020, doi: 10.1186/s13638-020-01765-7.

[32] R. Karthikeyan and R. Vadivel, "Boosted Mutated Corona Virus Optimization Routing Protocol (BMCVORP) for Reliable Data Transmission with Efficient Energy Utilization," *Wirel. Pers. Commun.*, vol. 135, no. 4, pp. 2281–2301, 2024, doi: 10.1007/s11277-024-11155-7.

[33] R. Karthikeyan and R. Vadivel, "Proficient Dazzling Crow Optimization Routing Protocol (PDCORP) for Effective Energy Administration in Wireless Sensor Networks," in *IEEE International Conference on Electrical, Electronics, Communication and Computers, ELEXCOM 2023*, Institute of Electrical and Electronics Engineers Inc., 2023. doi: 10.1109/ELEXCOM58812.2023.10370559.

[34] J. Ramkumar, B. Varun, V. Valarmathi, D. R. Medhunhashini, and R. Karthikeyan, "JAGUAR-BASED ROUTING PROTOCOL (JRP) FOR IMPROVED RELIABILITY AND REDUCED PACKET LOSS IN DRONE AD-HOC NETWORKS (DANET)," *J. Theor. Appl. Inf. Technol.*, vol. 103, no. 2, pp. 696–713, 2025, [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85217213044&partnerID=40&md5=e38a375e46cf43c95d6702a3585a7073

[35] B. Suchitra, J. Ramkumar, and R. Karthikeyan, "FROG LEAP INSPIRED OPTIMIZATION-BASED EXTREME LEARNING MACHINE FOR ACCURATE CLASSIFICATION OF LATENT AUTOIMMUNE DIABETES IN ADULTS (LADA)," *J. Theor. Appl. Inf. Technol.*, vol. 103, no. 2, pp. 472–494, 2025, [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85217140979&partnerID=40&md5=9540433c16d5ff0f6c2de4b8c43a4812

[36] J. Ramkumar, V. Valarmathi, and R. Karthikeyan, "Optimizing Quality of Service and Energy Efficiency in Hazardous Drone Ad-Hoc Networks (DANET) Using Kingfisher Routing Protocol (KRP)," *Int. J. Eng. Trends Technol.*, vol. 73, no. 1, pp. 410–430, 2025, doi: 10.14445/22315381/IJETT-V73I1P135.

[37] J. Ramkumar and R. Vadivel, "Whale optimization routing protocol for minimizing energy consumption in cognitive radio wireless sensor network," *Int. J. Comput. Networks Appl.*, vol. 8, no. 4, pp. 455–464, 2021, doi: 10.22247/ijcna/2021/209711.

[38] S. P. Geetha, N. M. S. Sundari, J. Ramkumar, and R. Karthikeyan, "ENERGY EFFICIENT ROUTING IN QUANTUM FLYING AD HOC NETWORK (Q-FANET) USING MAMDANI FUZZY INFERENCE ENHANCED DIJKSTRA'S ALGORITHM (MFI-EDA)," *J. Theor. Appl. Inf. Technol.*, vol. 102, no. 9, pp. 3708–3724, 2024, [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85197297302&partnerID=40&md5=72d51668bee6239f09a59d2694df67d6

[39] J. Ramkumar and D. Ravindran, "Machine learning and robotics in urban traffic flow optimization with graph neural networks and reinforcement learning," in *Machine Learning and Robotics in Urban Planning and Management*, 2025, pp. 83–104. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-105000106746&doi=10.4018%2F979-8-3693-9410-6.ch005&partnerID=40&md5=f647b3741afce4400893c2913f2bbf55

[40] R. Jaganathan, S. Mehta, and R. Krishan, "Preface," *Bio-Inspired Intell. Smart Decis.*, pp. xix–xx, 2024, [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85195725049&partnerID=40&md5=7a2aa7adc005662eebc12ef82e3bd19f

[41] S. P. Priyadharshini, F. Nirmala Irudayam, and J. Ramkumar, "An Unique Overture of Plithogenic Cubic Overset, Underset and Offset," in *Studies in Fuzziness and Soft Computing*, vol. 435, 2025, pp. 139–156. [Online]. Available:

https://www.scopus.com/inward/record.uri?eid=2-s2.0-105001675443&doi=10.1007%2F978-3-031-78505-4_7&partnerID=40&md5=e9def8c6a233de4fbf8f1549ad72027f

[42] R. Jaganathan, K. Rajendran, and P. S. Ponnukumar, "Peregrine Falcon Optimization Routing Protocol (PFORP) for Achieving Ultra-Low Latency and Boosted Efficiency in 6G Drone Ad-Hoc Networks (DANET)," *Int. J. Comput. Digit. Syst.*, vol. 17, no. 1, pp. 1–18, 2025, [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-105002444499&doi=10.12785%2Fijcds%2F1571111848&partnerID=40&md5=7d8374b797c2f718dcaa731fc2944439

[43] B. Suchitra, R. Karthikeyan, J. Ramkumar, and V. Valarmathi, "ENHANCING RECURRENT NEURAL NETWORK PERFORMANCE FOR LATENT AUTOIMMUNE DIABETES DETECTION (LADA) USING EXOCOETIDAE OPTIMIZATION," *J. Theor. Appl. Inf. Technol.*, vol. 103, no. 5, pp. 1645–1667, 2025, [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-105000948603&partnerID=40&md5=66c8f111b153fed68b3d0ea9c88c411e

[44] R. Jaganathan, S. Mehta, and R. Krishan, "Preface," *Intell. Decis. Mak. Through Bio-Inspired Optim.*, pp. xiii–xvi, 2024, [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85192858710&partnerID=40&md5=f8f1079e8772bd424d2cdd979e5f2710

[45] J. Ramkumar and R. Vadivel, "Multi-Adaptive Routing Protocol for Internet of Things based Ad-hoc Networks," *Wirel. Pers. Commun.*, vol. 120, no. 2, pp. 887–909, Apr. 2021, doi: 10.1007/s11277-021-08495-z.

[46] A. Senthilkumar, J. Ramkumar, M. Lingaraj, D. Jayaraj, and B. Sureshkumar, "Minimizing Energy Consumption in Vehicular Sensor Networks Using Relentless Particle Swarm Optimization Routing," *Int. J. Comput. Networks Appl.*, vol. 10, no. 2, pp. 217–230, 2023, doi: 10.22247/ijcna/2023/220737.

[47] P. Menakadevi and J. Ramkumar, "Robust Optimization Based Extreme Learning Machine for Sentiment Analysis in Big Data," *2022 Int. Conf. Adv. Comput. Technol. Appl. ICACTA 2022*, pp. 1–5, Mar. 2022, doi: 10.1109/ICACTA54488.2022.9753203.

[48] R. Vadivel and J. Ramkumar, "QoS-enabled improved cuckoo search-inspired protocol (ICSIP) for IoT-based healthcare applications," in *Incorporating the Internet of Things in Healthcare Applications and Wearable Devices*, IGI Global, 2019, pp. 109–121. doi: 10.4018/978-1-7998-1090-2.ch006.

[49] N. K. Ojha, A. Pandita, and J. Ramkumar, "Cyber security challenges and dark side of AI: Review and current status," in *Demystifying the Dark Side of AI in Business*, IGI Global, 2024, pp. 117–137. doi: 10.4018/979-8-3693-0724-3.ch007.

[50] S. P. Priyadharshini and J. Ramkumar, "Mappings Of Plithogenic Cubic Sets," *Neutrosophic Sets Syst.*, vol. 79, pp. 669–685, 2025, doi: 10.5281/zenodo.14607210.

[51] R. Jaganathan and V. Ramasamy, "Performance modeling of bio-inspired routing protocols in Cognitive Radio Ad Hoc Network to reduce end-to-end delay," *Int. J. Intell. Eng. Syst.*, vol. 12, no. 1, pp. 221–231, 2019, doi: 10.22266/IJIES2019.0228.22.

[52] J. Ramkumar, S. S. Dinakaran, M. Lingaraj, S. Boopalan, and B. Narasimhan, "IoT-Based Kalman Filtering and Particle Swarm Optimization for Detecting Skin Lesion," in *Lecture Notes in Electrical Engineering*, K. Murari, N. Prasad Padhy, and S. Kamalasadan, Eds., Singapore: Springer Nature Singapore, 2023, pp. 17–27. doi: 10.1007/978-981-19-8353-5_2.

[53] M. P. Swapna and J. Ramkumar, "Multiple Memory Image Instances Stratagem to Detect Fileless Malware," in *Communications in Computer and Information Science*, S. Rajagopal, K. Popat, D. Meva, and S. Bajeja, Eds., Cham: Springer Nature Switzerland, 2024, pp. 131–140. doi: 10.1007/978-3-031-59100-6_11.

[54] J. Ramkumar, A. Senthilkumar, M. Lingaraj, R. Karthikeyan, and L. Santhi, "OPTIMAL APPROACH FOR MINIMIZING DELAYS IN IOT-BASED QUANTUM WIRELESS SENSOR NETWORKS USING NM-LEACH ROUTING PROTOCOL," *J. Theor. Appl. Inf. Technol.*, vol. 102, no. 3, pp. 1099–1111, 2024, [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85185481011&partnerID=40&md5=bf0ff974ceabc0ad58e589b28797c684

[55] D. Jayaraj, J. Ramkumar, M. Lingaraj, and B. Sureshkumar, "AFSORP: Adaptive Fish Swarm Optimization-Based Routing Protocol

for Mobility Enabled Wireless Sensor Network," *Int. J. Comput. Networks Appl.*, vol. 10, no. 1, pp. 119–129, Jan. 2023, doi: 10.22247/ijcna/2023/218516.

[56] M. Lingaraj, T. N. Sugumar, C. S. Felix, and J. Ramkumar, "Query aware routing protocol for mobility enabled wireless sensor network," *Int. J. Comput. Networks Appl.*, vol. 8, no. 3, pp. 258–267, 2021, doi: 10.22247/ijcna/2021/209192.

[57] J. Ramkumar, R. Vadivel, and B. Narasimhan, "Constrained Cuckoo Search Optimization Based Protocol for Routing in Cloud Network," *Int. J. Comput. Networks Appl.*, vol. 8, no. 6, pp. 795–803, 2021, doi: 10.22247/ijcna/2021/210727.

[58] J. Ramkumar and R. Vadivel, "Improved Wolf prey inspired protocol for routing in cognitive radio Ad Hoc networks," *Int. J. Comput. Networks Appl.*, vol. 7, no. 5, pp. 126–136, 2020, doi: 10.22247/ijcna/2020/202977.

[59] J. Ramkumar, K. S. Jeen Marseline, and D. R. Medhunhashini, "Relentless Firefly Optimization-Based Routing Protocol (RFORP) for Securing Fintech Data in IoT-Based Ad-Hoc Networks," *Int. J. Comput. Networks Appl.*, vol. 10, no. 4, pp. 668–687, 2023, doi: 10.22247/ijcna/2023/223319.

[60] J. Ramkumar, C. Kumuthini, B. Narasimhan, and S. Boopalan, "Energy Consumption Minimization in Cognitive Radio Mobile Ad-Hoc Networks using Enriched Ad-hoc On-demand Distance Vector Protocol," in *2022 International Conference on Advanced Computing Technologies and Applications, ICACTA 2022*, Institute of Electrical and Electronics Engineers Inc., 2022. doi: 10.1109/ICACTA54488.2022.9752899.

[61] J. Ramkumar and R. Vadivel, "Improved frog leap inspired protocol (IFLIP) – for routing in cognitive radio ad hoc networks (CRAHN)," *World J. Eng.*, vol. 15, no. 2, pp. 306–311, 2018, doi: 10.1108/WJE-08-2017-0260.

[62] R. Jaganathan, S. Mehta, and R. Krishan, *Intelligent Decision Making Through Bio-Inspired Optimization*. Sri Krishna Arts and Science College, India: IGI Global, 2024. doi: 10.4018/979-8-3693-2073-0.

[63] M. P. Swapna, J. Ramkumar, and R. Karthikeyan, "Energy-Aware Reliable Routing with Blockchain Security for Heterogeneous Wireless Sensor Networks," in *Lecture Notes in Networks and Systems*, V. Goar, M. Kuri, R. Kumar, and T. Senjyu, Eds., Springer Science and Business Media Deutschland GmbH, 2025, pp. 713–723. doi: 10.1007/978-981-97-6106-7_43.

[64] J. Ramkumar, R. Karthikeyan, and V. Valarmathi, "Alpine Swift Routing Protocol (ASRP) for Strategic Adaptive Connectivity Enhancement and Boosted Quality of Service in Drone Ad Hoc Network (DANET)," *Int. J. Comput. Networks Appl.*, vol. 11, no. 5, pp. 726–748, 2024, doi: 10.22247/ijcna/2024/45.

[65] J. Ramkumar and R. Vadivel, "CSIP—cuckoo search inspired protocol for routing in cognitive radio ad hoc networks," in *Advances in Intelligent Systems and Computing*, Springer Verlag, 2017, pp. 145–153. doi: 10.1007/978-981-10-3874-7_14.

[66] L. Mani, S. Arumugam, and R. Jaganathan, "Performance Enhancement of Wireless Sensor Network Using Feisty Particle Swarm Optimization Protocol," in *ACM International Conference Proceeding Series*, Association for Computing Machinery, 2022. doi: 10.1145/3590837.3590907.

[67] K. S. J. Marseline, J. Ramkumar, and D. R. Medhunhashini, "Sophisticated Kalman Filtering-Based Neural Network for Analyzing Sentiments in Online Courses," in *Smart Innovation, Systems and Technologies*, A. K. Somani, A. Mundra, R. K. Gupta, S. Bhattacharya, and A. P. Mazumdar, Eds., Springer Science and Business Media Deutschland GmbH, 2024, pp. 345–358. doi: 10.1007/978-981-97-3690-4_26.

[68] R. Jaganathan and R. Vadivel, "Intelligent Fish Swarm Inspired Protocol (IFSIP) for Dynamic Ideal Routing in Cognitive Radio Ad-Hoc Networks," *Int. J. Comput. Digit. Syst.*, vol. 10, no. 1, pp. 1063–1074, 2021, doi: 10.12785/ijcds/100196.

[69] J. Ramkumar, R. Karthikeyan, and M. Lingaraj, "Optimizing IoT-Based Quantum Wireless Sensor Networks Using NM-TEEN Fusion of Energy Efficiency and Systematic Governance," in *Lecture Notes in Electrical Engineering*, V. Shrivastava, J. C. Bansal, and B. K. Panigrahi, Eds., Springer Science and Business Media Deutschland GmbH, 2025, pp. 141–153. doi: 10.1007/978-981-97-6710-6_12.

[70] R. Jaganathan, S. Mehta, and R. Krishan, *Bio-Inspired intelligence for smart decision-making*. IGI Global, 2024. doi: 10.4018/9798369352762.