ENHANCING RECURRENT NEURAL NETWORK PERFORMANCE FOR LATENT AUTOIMMUNE DIABETES DETECTION (LADA) USING EXOCOETIDAE OPTIMIZATION

B. SUCHITRA[1], R. KARTHIKEYAN[2], J. RAMKUMAR[3], V. VALARMATHI4

[1]Assistant Professor, Department of IT & Cognitive Systems, Sri Krishna Arts and Science College, India
[2]Assisant Professor, Department of Computer Technology, Sri Krishna Adhithya College of Arts and Science, India
[3]Assistant Professor, Department of IT & Cognitive Systems, Sri Krishna Arts and Science College, India
[4]Assistant Professor, Department of IT & Cognitive Systems, Sri Krishna Arts and Science College, India

E-mail: [1]suchipradeep@gmail.com, [2]karthikeyanrkrish@gmail.com, [3]jramkumar1986@gmail.com, [4]valarmathiv@skasc.ac.in

ABSTRACT

Latent Autoimmune Diabetes in Adults (LADA) is frequently misclassified as type 2 diabetes, a significant issue highlighted in current diagnostic research. This misclassification, due to overlapping clinical features with other diabetes types, leads to suboptimal patient management. Existing models lack the precision to discern the subtle, time-dependent variations characteristic of LADA, underscoring a critical gap in diabetes diagnostics. This study introduces a groundbreaking approach, the Exocoetidae Optimization-Inspired Recurrent Neural Network (EO-RNN), which innovatively incorporates the adaptive flight patterns of Exocoetidae (flying fish) into optimizing RNNs. This bio-inspired algorithm enhances parameter tuning and model adaptability, significantly improving the detection accuracy of complex disease patterns. Employing the EO-RNN on the Gestational Diabetes Mellitus (GDM) dataset resulted in a classification accuracy of 95.36% and an F-measure of 95.47%, surpassing traditional models by over 30%. These results not only bridge the identified literature gap by providing a more effective diagnostic tool but also contribute substantial new knowledge about the integration of nature-inspired algorithms in medical diagnostics. The success of EO-RNN in enhancing diagnostic accuracy demonstrates its potential as a transformative tool in healthcare, promising for broader applications in detecting complex diseases where similar misclassification issues exist.

Keywords: *LADA Classification, Exocoetidae Optimization, Recurrent Neural Networks, Bio-Inspired Optimization, Diabetes Prediction, Machine Learning In Healthcare, Temporal Dependency Modeling*

1. INTRODUCTION

Latent Autoimmune Diabetes in Adults (LADA) stands as a unique subtype within the diabetes spectrum, combining autoimmune features typical of type 1 diabetes with the age of onset and some characteristics associated with type 2 diabetes. LADA emerges from an autoimmune attack on pancreatic beta cells, impairing insulin production[1]. Unlike type 1 diabetes, where beta-cell destruction occurs swiftly, LADA progresses more gradually, leading to a slower insulin dependence timeline and typically affecting adults over 30. Misclassifying LADA as type 2 diabetes remains common due to the slow onset and age factor, complicating treatment and often delaying optimal therapy. The identification of specific autoantibodies, particularly glutamic acid decarboxylase (GAD) antibodies, aids in accurately diagnosing LADA[2]. By recognizing these markers, medical practitioners can differentiate LADA from other forms of diabetes, allowing for more suitable therapeutic strategies. A LADA diagnosis often necessitates a shift in treatment toward insulin-based therapies rather than standard oral medications for type 2 diabetes. This shift aims to control blood glucose levels while addressing the underlying autoimmune mechanisms, potentially preserving remaining beta-cell function. Improved awareness and diagnostic clarity regarding LADA allow for more targeted treatment plans, enhancing long-term outcomes and quality of life for individuals with this form of diabetes[3].

Classification algorithms have transformed the approach to diabetes prediction by effectively analyzing medical data and determining risk levels, facilitating proactive healthcare. By evaluating multiple health factors, such as insulin levels, genetic markers, and patient lifestyle choices, classification algorithms reveal patterns indicating a higher probability of diabetes development or categorize existing conditions into specific types[4]. Such precision enables early identification of at-risk individuals, aiding healthcare professionals in applying preventive measures. In practical applications, classification models strengthen the predictive accuracy of medical diagnostics, guiding clinical decisions that can prevent the progression of diabetes complications. By accurately classifying cases, these algorithms allow for targeted treatment plans, which may include specific lifestyle changes or medications tailored to each patient's profile[5], [6]. This personalized approach supports improved outcomes, as treatments can be more effective when matched to individual risk factors and diabetes types. Beyond individual patient care, classification algorithms aid in population health management, identifying trends in diabetes risk and supporting public health strategies aimed at reducing the prevalence of the disease. Through accurate prediction.

Classification algorithms have transformed the approach to diabetes prediction by effectively analyzing medical data and determining risk levels, facilitating proactive healthcare. By evaluating multiple health factors, such as insulin levels, genetic markers, and patient lifestyle choices, classification algorithms reveal patterns indicating a higher probability of diabetes development or categorize existing conditions into specific types[4]. Such precision enables early identification of at-risk individuals, aiding healthcare professionals in applying preventive measures. In practical applications, classification models strengthen the predictive accuracy of medical diagnostics, guiding clinical decisions that can prevent the progression of diabetes complications. By accurately classifying cases, these algorithms allow for targeted treatment plans, which may include specific lifestyle changes or medications tailored to each patient's profile[5], [6]. This personalized approach supports improved outcomes, as treatments can be more effective when matched to individual risk factors and diabetes types. Beyond individual patient care, classification algorithms aid in population health management, identifying trends in diabetes risk and supporting public health strategies aimed at reducing the prevalence of the disease. Through accurate prediction and early intervention, classification algorithms play a vital role in managing and mitigating diabetes on a broader scale[7].

Bio-inspired optimization plays a critical role in refining classification algorithms for diabetes prediction by using techniques modeled on biological systems, including natural selection, swarm behavior, and evolutionary strategies[8]. In diabetes classification, these techniques improve the selection and weighting of features, such as blood pressure, lifestyle, insulin levels, and genetic markers, streamlining the model to prioritize the most informative variables. This streamlined approach increases classification accuracy and efficiency, making it possible to predict diabetes risk levels or identify specific types with reduced computational resources. Handling high-dimensional, often imbalanced datasets is another area where bio-inspired optimization proves beneficial[9]. These methods enable classification algorithms to efficiently process vast amounts of patient data, balancing and adapting the model to avoid overfitting, a common issue in medical diagnostics. Through natural-inspired algorithms, such as those mimicking swarm intelligence or evolutionary selection, classification models become capable of quickly adapting to changes and learning from new data, thus ensuring consistent performance over time. Bio-inspired optimization ultimately enhances diabetes prediction accuracy and resource efficiency, supporting more tailored treatments and early interventions. This approach has transformed classification algorithms, making them highly adaptable and suitable for real-time, personalized healthcare solutions in diabetes management[10].

1.1. Problem Statement

Accurately distinguishing diabetes types has posed a persistent challenge in clinical diagnostics due to the subtle differences and shared characteristics between them. Traditional diagnostic approaches often rely on limited biomarkers, lacking comprehensive insight into patient data complexities, which include age, lifestyle, and genetic factors. These approaches lead to inconsistencies in diagnosis, particularly with conditions like Latent Autoimmune Diabetes in Adults (LADA), which shares characteristics of both type 1 and type 2 diabetes. Misclassifications result in inappropriate treatment plans, reducing their effectiveness and potentially worsening patient outcomes. Furthermore, existing algorithms lack robust methods for handling large-scale, multidimensional datasets, limiting their ability to capture nuanced patterns essential for accurate classification. The gap in reliable, adaptable classification tools has underscored a need for advanced solutions that can accommodate vast, diverse datasets, improve diagnostic precision, and support timely intervention for individuals at various stages of diabetes development.

1.2. Motivation

The need for improved diabetes classification has grown in light of the health and economic burden diabetes imposes worldwide, requiring more accurate, personalized diagnostic approaches. Current algorithms often struggle to differentiate between diabetes types, leading to delayed or inappropriate treatment, which can exacerbate health risks and complications. With diabetes involving multifaceted factors like genetic predispositions, lifestyle, and environmental influences, conventional classification tools lack the adaptability to navigate this complexity effectively. Recent advancements in bio-inspired optimization present an opportunity to revolutionize diabetes classification by offering robust, flexible models that improve accuracy, even with high-dimensional data. This motivation stems from the potential to shift from one-size-fits-all approaches to precision medicine, enabling tailored interventions. A focus on refining classification algorithms to address this diversity directly impacts patient outcomes, offering a pathway to timely, customized care that enhances overall health and well-being for individuals with diabetes.

1.3. Objective

The main objective of this research involves creating an Exocoetidae Optimization-inspired Recurrent Neural Network (EO-RNN) that accurately captures temporal patterns in patient data to improve classification of Latent Autoimmune Diabetes in Adults (LADA). Drawing inspiration from the gliding and adaptive behaviors of Exocoetidae (flying fish), this objective optimizes the RNN's parameters to handle complex, time-dependent data patterns essential for identifying LADA. The EO-RNN framework aims to improve the model's adaptability and precision in recognizing subtle diabetes progression indicators, which are critical for distinguishing LADA cases. This advanced classification model seeks to provide detailed insights into patient health trajectories, supporting healthcare providers with a responsive, high-performance tool for accurate LADA diagnosis and facilitating early, tailored treatment options.

2. LITERATURE REVIEW

Hybrid Artificial Neural Network (HANN) is proposed for diagnosing diabetes mellitus using a systematic and data-driven approach. The study evaluates the effectiveness of neural networks by analyzing medical attributes associated with diabetes, such as physiological and biochemical factors. Through rigorous testing with k-fold cross-validation, the model ensures robust and reliable predictions. The performance optimization, influenced by varying neuron counts in the hidden layers, highlights the importance of architecture design in neural networks. This research contributes to the growing field of AI in healthcare by providing a scalable and accurate method for predicting chronic diseases, fostering advancements in preventive medicine [11]. Efficient Deep Learning Technique (EDLT) is proposed for diabetes classification and prediction using the Indian Diabetes Dataset. The study focuses on leveraging deep learning models to analyze complex patient data and provide accurate classification of diabetic and non-diabetic individuals. Through effective preprocessing and feature selection, the technique optimizes model performance. The research emphasizes precision and reliability, showcasing the system's ability to achieve high accuracy across various evaluation metrics. This approach highlights the potential of deep learning in enhancing healthcare diagnostics and offering scalable solutions for early detection and management of diabetes[12].

Efficient Ensemble-based Model by Chhabra et al. (2024) applies an ensemble machine learning approach for chronic kidney disease prediction, combining multiple models to enhance diagnostic precision. By integrating techniques such as Random Forest and Gradient Boosting, this model effectively addresses data variance, providing consistent predictions and improving accuracy in medical imaging for nephrology applications [13]. En-RfRsK Ensemble Model by Amma (2024) utilizes an ensemble of random forests and rule-based classifiers for diabetes prognosis. This approach enhances model interpretability while retaining high predictive accuracy by blending rule extraction and ensemble techniques, enabling healthcare professionals to gain clearer insights into disease progression [14]. Multi-Layered Hybrid Algorithm (MLHA) by Jannoud et al. (2024) for Type II diabetes classification combines multiple classifiers in a layered approach. By structuring algorithms in hierarchical layers, the model enhances robustness and classification precision, facilitating improved diagnostic accuracy and support for managing diabetes care [15].

Atherosclerosis Biomarker Identification Model by Zhang et al. (2024) employs bioinformatics and machine learning for identifying atherosclerosis markers in PCOS. The model processes genomic and proteomic data, allowing it to pinpoint biomarkers associated with disease progression, presenting a useful tool for personalized healthcare in complex syndromes [16]. Big Data and IoT Health Prediction Model by Zamani et al. (2024) integrates machine learning with IoT and big data to enhance prediction models in health informatics. This framework leverages large-scale data from IoT devices, optimizing model accuracy and enabling proactive healthcare interventions by anticipating health trends and risks [17]. Metformin Response Prediction Model by Villikudathil et al. (2024) analyzes clinical, genomic, and proteomic data to predict response to metformin monotherapy in Type-2 diabetes. The model uses diverse machine learning algorithms to reveal insights into individual therapeutic responses, guiding personalized treatment options for diabetic patients [18]. Hematologic Biomarker Model for Infection Prognosis by Upadhyaya et al. (2024) uses interpretability methods to rank hematologic biomarkers in infection prognosis. This model helps clinicians understand which biomarkers most impact patient outcomes, enhancing decision-making in infection management by providing interpretable predictions [19]. Survival Prediction Model for Kidney Transplant Recipients by Díez-Sanmartín et al. (2024) applies ensemble learning to predict kidney transplant survival rates. By combining multiple machine learning models, this algorithm addresses survival-related complexities, improving prognostic accuracy and aiding in personalized patient care planning [20].

Pre-eclampsia Prediction Model by Tiruneh et al. (2024) uses routinely collected data and machine learning techniques to predict pre-eclampsia risk. The model prioritizes feature selection to optimize predictive accuracy, making it an essential tool for early intervention in prenatal care, potentially reducing adverse outcomes [21]. Personalized Nutrition Model with Continuous Glucose Monitoring **by**Zignoli et al. (2024) explores machine learning applications in nutrition, focusing on glucose monitoring in non-controlled settings. By analyzing continuous glucose data, the model provides tailored nutritional insights, enhancing individualized dietary recommendations for improved health management [22]. Preeclampsia Prediction Model by Wesson and Smith (2024) employs machine learning to predict preeclampsia risk in pregnant women. Using patient health metrics, including blood pressure and protein levels, the model identifies high-risk indicators that often precede preeclampsia. Feature selection reduces irrelevant data, improving prediction accuracy and interpretability, allowing clinicians to act preemptively. Such early identification supports timely intervention strategies to minimize pregnancy-related complications, enhancing both maternal and fetal outcomes. This model stands out in prenatal healthcare, offering a non-invasive and data-driven method to tackle a major pregnancy risk factor [23]. IP-GCN for Insulin Prediction by Ali et al. (2024) utilizes graph convolutional networks (GCNs) for modeling biochemical interactions relevant to insulin response prediction in diabetes drug development. IP-GCN captures complex biochemical interactions and molecular structures within diabetic biomarkers, optimizing feature extraction to predict patient insulin responses effectively. This model supports the development of tailored diabetes drugs by mapping biological interactions, a key challenge in pharmacogenomics. By employing GCNs, IP-GCN enhances the accuracy of drug response predictions, potentially accelerating drug development processes and improving patient-specific treatment plans for diabetes management [24].

Coronary Artery Disease Prediction by Hassan et al. (2024) integrates feature selection with machine learning for predicting coronary artery disease (CAD). The model uses algorithms like Support Vector Machine and Random Forest to identify risk indicators by analyzing large patient datasets, including clinical variables and biomarkers. By emphasizing dimensionality reduction, the approach boosts accuracy while minimizing computational load. This innovation supports early CAD diagnosis, empowering healthcare providers with an efficient tool for managing high-risk patients, potentially improving intervention strategies and patient outcomes by identifying cardiovascular risks before they manifest severely [25]. Metabolic Syndrome Prediction Models by Pawade et al. (2024) evaluate multiple machine learning algorithms—such as Decision Trees, Neural Networks, and K-Nearest Neighbors—for assessing metabolic syndrome risk.

Through comparative performance analysis, the study highlights which models yield the most accurate predictions across large datasets. Each algorithm is tested on a range of features, including BMI, blood glucose levels, and cholesterol, to establish a robust, data-driven risk assessment framework. Findings underscore the advantages of ensemble methods in metabolic syndrome prediction, presenting healthcare practitioners with optimized models to assess and manage metabolic risks effectively [26]. Network Performance Prediction Model by Su (2024) leverages machine learning algorithms to forecast computer network performance issues, integrating data on bandwidth, latency, and packet loss. The model employs predictive algorithms, including Neural Networks and Decision Trees, to process vast amounts of network data for performance optimization. Through analyzing trends and anomalies, the model anticipates bottlenecks and other issues, assisting network administrators in preemptively managing resources. This approach enables real-time monitoring and optimized network functionality, a critical factor in environments where network stability directly impacts organizational efficiency and service delivery [27].

Parkinson's Disease Detection Model by Al Mudawi (2024) applies machine learning for early detection of Parkinson's disease, using algorithms like SVM, Decision Trees, and k-NN to analyze symptoms and neurological markers. By combining clinical features such as tremor intensity, rigidity, and speech patterns, the model identifies early-stage disease indicators, providing a non-invasive diagnostic option. The emphasis on feature selection optimizes accuracy by isolating critical variables, enhancing early intervention potential. This predictive model stands out in neurology, offering clinicians a valuable tool to detect and address Parkinson's progression before it significantly impairs quality of life [28]. Fuzzy Logic Model for Hormonal Imbalance by Khushal and Fatima (2024) employs a fuzzy machine learning approach to manage uncertainty in hormonal data analysis, applying it to hormonal imbalance prediction. The fuzzy logic system integrates linguistic variables and machine learning to improve predictions for endocrine disorders. By accommodating data imprecision, the model strengthens diagnostic reliability in assessing hormonal levels, which can fluctuate widely. This integration supports endocrinologists by providing more consistent insights, especially for diseases where exact hormone measurements are difficult, enhancing early diagnosis and targeted treatment for patients with hormonal disorders [29]. Binge-Eating Disorder Prediction Model by Procopio et al. (2025) utilizes a physiological machine learning model based on OGTT curve data to classify patients with binge-eating disorder (BED). By processing blood glucose levels over time, the model detects glucose fluctuation patterns characteristic of BED, allowing for early diagnosis. This physiological approach combines traditional clinical data with machine learning to distinguish BED patients from controls, aiding psychologists in developing personalized treatment plans. The study's non-invasive diagnostic alternative demonstrates the potential of physiological modeling in mental health, particularly for conditions with significant metabolic impacts like BED [30].

Meal Detection in Diabetes Using Ensemble Models by Ibrahim et al. (2024) improves postprandial glucose control through an ensemble model designed to detect unannounced meals. The model integrates several algorithms, such as Random Forest and Neural Networks, to interpret glucose fluctuations indicative of meal intake. This ensemble approach strengthens prediction accuracy, supporting real-time monitoring and intervention for diabetics who struggle with post-meal glucose spikes. By providing timely insights, this model offers a more dynamic solution to glucose management, which is crucial for maintaining optimal blood sugar levels and reducing diabetes complications [31]. Viral Pneumonia Prediction Model by Wang et al. (2024) uses pulmonary inflammation scores and machine learning to predict viral pneumonia risk. This model processes scores related to lung inflammation, respiratory rate, and oxygen levels to assess infection likelihood. By identifying critical features, the model enhances early pneumonia detection, providing timely guidance for antiviral treatments. Its high accuracy in distinguishing pneumonia cases from other respiratory conditions offers clinicians a reliable diagnostic tool, contributing to faster and more effective patient management, particularly in acute care settings where respiratory infections are prevalent [32]. Bio-inspired optimization are crucial is all types of researches to enrich the expected outcomes [33]-[71].

3. EXOCOETIDAE OPTIMIZATION-INSPIRED RECURRENT NEURAL NETWORK (EO-RNN)

LADA represents a critical diagnostic challenge within the spectrum of diabetes care. Characterized by its autoimmune nature yet often presenting with clinical features similar to type 2 diabetes, LADA requires precise diagnostic strategies to differentiate it effectively from other diabetes types. Current diagnostic models often fall short, leading to frequent misclassification and suboptimal treatment approaches that may exacerbate the patient's condition over time. This study introduces a novel approach to enhancing the diagnostic accuracy of LADA through the application of Exocoetidae Optimization-Inspired Recurrent Neural Networks (EO-RNN). Drawing inspiration from the dynamic and adaptive behaviors of flying fish, this bio-inspired optimization technique aims to refine the capabilities of RNNs by improving their parameter tuning and adaptability to complex data patterns prevalent in medical diagnostics. The need for this study is underscored by the potential to significantly improve patient outcomes through more accurate and timely diagnosis, enabling appropriate therapeutic strategies that are tailored to the specific needs of LADA patients. By advancing the field of machine learning in medical diagnostics, this research not only fills a crucial gap but also sets a foundation for further innovations in the application of bio-inspired algorithms for complex disease detection. Through the successful implementation and validation of EO-RNN, this study promises to offer a transformative approach to LADA detection, with broader implications for enhancing precision medicine in diabetes care and beyond.

Exocoetidae Optimization-inspired Recurrent Neural Network (EO-RNN) merges the principles of Exocoetidae optimization—a bio-inspired algorithm reflecting the unique flight behaviors of flying fish—with the powerful sequence modeling capabilities of Recurrent Neural Networks (RNNs). This innovative framework leverages the local and global search strategies inherent in the Exocoetidae optimization process to enhance the parameter optimization of RNNs, effectively improving their performance in tasks involving temporal dependencies, such as time-series prediction and classification. By simulating the movement of flying fish, EO-RNN explores diverse regions of the parameter space, maintaining population diversity while avoiding premature convergence. The integration of EO into RNN training allows for more effective learning and generalization across varying datasets. Consequently, EO-RNN presents a robust approach to optimizing neural network architectures, leading to enhanced predictive accuracy and efficiency in real-world applications.

3.1. Define the Problem and Dataset for RNN Input

First step establishes the classification task and prepares the dataset for Recurrent Neural Network (RNN) processing. Consider a classification problem with a dataset $D = \{X, Y\}$, where X represents the input sequences, and Y denotes the corresponding labels. The task is to classify sequences based on temporal dependencies captured by the RNN.The dataset D consists of multiple sequences $X_i \in R^{T\times d}$, where $i \in \{1,2,\dots,N\}, N$ is the total number of sequences, T is the length of each sequence, and d is the dimensionality of the feature space. Each sequence $X_i = \{x_{i1}, x_{i2}, \dots, x_{iT}\}$, where $x_{it} \in R^d$, corresponds to a time-ordered set of feature vectors, and Y_i represents the class label for the sequence X_i, with $Y_i \in \{1,2,\dots,C\}$ for a classification task involving C classes.

The goal is to design the RNN to capture temporal dependencies across the input sequences.X_i and predict the corresponding label Y_i. The objective function to be minimized is the cross-entropy loss, defined as:

$$L(\theta) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} y_{ic}\, log\hat{y}_{ic} \tag{1}$$

where θ represents the parameters of the RNN (weights and biases), y_{ic} is the true label indicator for class c of sequence i, and $\hat{y}_{ic}$ is the predicted probability for class c.

The input to the RNN consists of sequences X_i, which need to be normalized or scaled to ensure consistent training. Normalization transforms each feature x_{it} in sequence X_i using:

$$x'_{it} = \frac{x_{it} - \mu}{\sigma} \tag{2}$$

where μ is the mean of the feature values across all sequences, and σ is the standard deviation. Normalization ensures that each feature is centered and scaled, improving training stability.

Each sequence X_i is passed into the RNN, which processes it step by step. The hidden state h_t at time step t is updated based on the current input x_{it} and the previous hidden state h_{t-1}. This is computed as:

$$h_t = f(W_h h_{t-1} + W_x x_{it} + b_h) \tag{3}$$

where $W_h \in R^{d_h \times d_h}$ is the recurrent weight matrix, $W_x \in R^{d_h \times d}$ is the input weight matrix, $b_h \in R^{d_h}$ is the bias vector, and f is a non-linear activation function such as $tanh$.

The final output of the RNN at time step T, denoted as h_T, is passed through a softmax layer to produce the predicted class probabilities $\hat{y}_i$ for sequence X_i. The softmax function is applied as:

$$\hat{y}_i = softmax(W_o h_T + b_o) \tag{4}$$

where $W_o \in R^{C \times d_h}$ is the output weight matrix, and $b_o \in R^C$ is the output bias.

EO algorithm optimizes the RNN parameters$\theta = \{W_h, W_x, W_o, b_h, b_o\}$ to minimize the cross-entropy loss $L(\theta)$ through an iterative process of local (swimming) and global (flying) parameter updates.

3.2. Initialize RNN Architecture

Initialization defines the architecture for processing input sequences X_i, previously normalized in Step 1, to capture temporal patterns effectively. Weights and biases parameterize the architecture, denoted as W_h, W_x, W_o, b_h, b_o, which will be optimized later using EO.The RNN comprises an input layer, a recurrent hidden layer, and an output layer. The input to the RNN is the sequence $X_i = \{x_{i1}, x_{i2}, .., x_{iT}\}$, where each $x_{id} \in R^d$ is the feature vector at time step t. Each feature vector x_{it} is passed into the input layer of the RNN, where the weight matrix $W_x \in R^{d_h \times d}$ maps the input into the hidden state space. At time step t, the hidden state $h_t \in R^{d_h}$ is updated by combining the previous hidden state h_{t-1} and the current input x_{it}. The update equation for the hidden state is defined as:

$$h_t = f(W_h h_{t-1} + W_x x_{it} + b_h) \tag{5}$$

where $W_h \in R^{d_h \times d_h}$ is the weight matrix for the recurrent connection, and $b_h \in R^{d_h}$ is the bias term applied to the hidden layer. The activation function f, typically a non-linear function like tanh or ReLU, controls the output of the hidden state.

The hidden state h_t maintains a memory of previous time steps, enabling the RNN to capture temporal dependencies in the input sequence. The hidden state is updated iteratively for all $t \in \{1,2, \ldots, T\}$, with the initial hidden state h_0 often initialized as a zero vector, $h_0 = 0$. At the final time step T, the hidden state h_T represents the aggregated information from the entire sequence X_i. This hidden state is passed through the output layer to compute the predicted class probabilities $\hat{y}_i$ for sequence X_i. The output layer applies a softmax function to the final hidden state, as follows:

$$\hat{y}_i = softmax(W_o h_T + b_o) \tag{6}$$

where $W_o \in R^{C \times d_h}$ is the output weight matrix, and $b_o \in R^C$ is the bias vector for the output layer. The softmax function converts the output into a probability distribution over the C classes, where $\hat{y}_{ic}$ represents the predicted probability of class c for sequence X_i.

The architecture design involves selecting the number of hidden units d_h, which determines the dimensionality of the hidden state h_t, and the number of layers in the RNN. A single-layer RNN is expressed as:

$$h_t = f\left(W_h^{(1)} h_{t-1}^{(1)} + W_x^{(1)} x_{it} + b_h^{(1)}\right) \tag{7}$$

For multi-layer RNNs, additional hidden layers are stacked, and each hidden layer computes its hidden state based on the output of the previous layer. For example, in a two-layer RNN:

$$h_t^{(2)} = f\left(W_h^{(2)} h_{t-1}^{(2)} + W_x^{(2)} h_t^{(1)} + b_h^{(2)}\right) \tag{8}$$

The parameters $\theta = \{W_h, W_x, W_o, b_h, b_o\}$initialized in this step define the architecture's capacity to learn temporal dependencies in the data. These parameters are subsequently optimized using Exocoetida Optimization (EO) to minimize the loss function defined in Section 3.1.

$$L(\theta) = -\frac{1}{N} \sum_{i=1}^{N} \sum_{c=1}^{C} y_{ic} log \hat{y}_{ic} \tag{9}$$

3.3. Initialize EO Population

This step focuses on initializing the population of EO to optimize the parameters of the RNN. This step involves creating a population of candidate solutions, where each solution represents a set of RNN parameters that include the weights and biases initialized in Step 2. The goal is to explore the parameter space through the EO framework, enhancing the performance of the RNN in terms of classification. The population consists of M flying fish, where each fish F_j for $j \in \{1,2, \ldots., M\}$ represents ta candidate solution for the RNN parameter set $\theta_j = \{W_h^j, W_x^j, W_o^j, b_h^j, b_o^j\}$. These parameters are randomly initialized within a predefined range, ensuring that the initial population explores diverse regions of the parameter space.

The position of each fish in the search space corresponds to the specific values of the RNN parameters. The dimensionality of the search space is defined by the total number of parameters in the RNN. For a given fishF_j, its position is represented as a vector:

$$\theta_j = \left[W_h^j, W_x^j, W_o^j, b_h^j, b_o^j\right] \in R^n \tag{10}$$

where n represents the total number of parameters in the RNN, including all weight matrices and bias terms. The initial values of the weights$W_h^j, W_x^j, W_o^j$and biases b_h^j, b_o^j are drawn from a uniform or Gaussian distribution:

$$W_h^j, W_x^j, W_o^j \sim u(-\epsilon, \epsilon) \tag{11}$$

$$b_h^j, b_o^j \sim (-\epsilon, \epsilon) \tag{12}$$

where ϵ is a small constant that controls the range of initialization. This ensures that the initial fish population explores different areas of the parameter space before optimization begins.

Each fish F_j is evaluated based on the fitness function derived from the RNN's performance on the classification task. The fitness of each fish is measured by training the RNN with the current parameter setθ_j and calculating the cross-entropy loss $L(\theta_j)$, as defined in Step 1:

$$L(\theta_j) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} y_{ic} log\hat{y}_{ic} \tag{13}$$

The fitness of fish F_j is then given by:

$$Fitness(F_j) = -L(\theta_j) \tag{14}$$

The population of fish is initialized with a diversity of parameter sets to ensure that the Exocoetida Optimization algorithm has a broad search space to explore. The goal of EO is to evolve these initial parameter sets through the swimming (local search) and flying (global search) behaviors, ultimately finding the optimal parameter set θ^* that minimizes the loss function. The positions of the fish are updated based on their fitness, and the EO algorithm governs the movement of each fish. Fish with better fitness values guide the exploration, while others adjust their positions based on both local and global exploration mechanisms. The movement equations for the fish during swimming and flying behaviors will be introduced in subsequent steps, where local and global exploration of the parameter space takes place.

3.4. RNN Training with Initial Parameters

This step process focuses on optimizing the RNN's performance to minimize the loss function defined in previous steps.Each fish F_j in the population has a corresponding set of parameters $\theta_j = \{W_h^j, W_x^j, W_o^j, b_h^j, b_o^j\}$ that represent the weights and biases for the RNN. The training process begins by feeding the input sequences X_i into the RNN, which are structureds time-series data. For each sequence X_i, the RNN processes the input in a stepwise manner, updating the hidden state h_t at each time step t using the previously defined equations:

$$h_t = f\left(W_h^j h_{t-1} + W_x^j x_{it} + b_h^j\right) \tag{15}$$

The hidden state h_t aggregates information from both the current input x_{it} and the previous hidden state h_{t-1}. This allows the RNN to capture temporal dependencies within the sequence. After processing the entire sequence X_i, the final hidden state h_T is obtained and utilized to generate the output probabilities $\hat{y}_i$ through the softmax layer:

$$\hat{y}_i = softmax\left(W_o^j h_T + b_o^j\right) \tag{16}$$

The output $\hat{y}_{ic}$ indicates the predicted probability for each class c for sequence X_i. The objective is to minimize the difference between the predicted probabilities and the true labels Y_i. The loss function, represented as cross-entropy, is computed to quantify the model's performance:

$$L(\theta_j) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} y_{ic} log\hat{y}_{ic} \tag{17}$$

where $L(\theta_j)$ measures how well the predicted probabilities $\hat{y}_{ic}$ align with the true labels y_{ic}. Lower values of the loss indicate better performance.

Backpropagation through time (BPTT) is employed to update the RNN parameters. Gradients of the loss function concerning the weights and biases are calculated, allowing for efficient updates. The gradients for the weights W_h^j, W_x^j, and W_o^j can be computed as follows:

$$\frac{\partial L}{\partial W_o^j} = \sum_{i=1}^{N} (\hat{y}_i - Y_i) h_T^T \qquad (18)$$

$$\frac{\partial L}{\partial W_h^j} = \sum_{t=1}^{T} \frac{\partial L}{\partial h_t} h_{t-1}^T \qquad (19)$$

$$\frac{\partial L}{\partial W_x^j} = \sum_{t=1}^{T} \frac{\partial L}{\partial h_t} x_{it}^T \qquad (20)$$

where the gradients are accumulated over all sequences in the dataset to perform a batch update. Once the gradients are calculated, the parameters are updated using an optimization algorithm, typically stochastic gradient descent or its variants, such as Adam:

$$W_h^j \leftarrow W_h^j - \eta \frac{\partial L}{\partial W_h^j} \qquad (21)$$

$$W_x^j \leftarrow W_x^j - \eta \frac{\partial L}{\partial W_x^j} \qquad (22)$$

$$W_o^j \leftarrow W_o^j - \eta \frac{\partial L}{\partial b_h^j} \qquad (23)$$

$$b_h^j \leftarrow b_h^j - \eta \frac{\partial L}{\partial b_h^j} \qquad (24)$$

$$b_o^j \leftarrow b_o^j - \eta \frac{\partial L}{\partial b_o^j} \qquad (25)$$

whereη represents the learning rate, which controls the size of the updates. This process is repeated for a predetermined number of epochs or until convergence is reached.

Upon completion of training for each fish, the fitness function for the parametersθ_j is calculated based on the achieved loss $L(\theta_j)$,which will later guide the Exocoetida Optimization process to refine the parameters further. This training process is crucial for ensuring that the RNN learns to classify the sequences within the dataset effectively, setting the stage for subsequent optimization steps.

3.5. Swimming Behavior (RNN Parameter Fine-Tuning)

This behavior allows each fish, representing a candidate solution, to make local adjustments to its parameters, enhancing the exploration of the parameter space to improve RNN performance. In the swimming phase, each fish F_j updates its parameter set $\theta_j = \{W_h^j, W_x^j, W_o^j, b_h^j, b_o^j\}$using a local search mechanism. The aim is to explore nearby regions of the parameter space while maintaining the quality of the solution. The swimming behavior introduces perturbations to the parameters based on a defined step size δ, which determines the magnitude of the changes applied to each parameter.The updated parameters for each fish can be expressed as:

$$W_h^{j'} \leftarrow W_h^j + \delta.rand_h \qquad (26)$$

$$W_x^{j'} \leftarrow W_x^j + \delta.rand_x \qquad (27)$$

$$W_o^{j'} \leftarrow W_o^j + \delta.rand_o \qquad (28)$$

$$b_h^{j'} \leftarrow b_h^j + \delta.rand_{b_h} \qquad (29)$$

$$b_o^{j'} \leftarrow b_o^j + \delta.rand_{b_o} \qquad (30)$$

where $rand_h, rand_x, rand_o, rand_{b_h}, rand_{b_o}$are random variables uniformly distributed in the range $[-1,1]$, allowing the parameters to be perturbed in both positive and negative directions. This ensures a comprehensive exploration of the parameter space around the current values.

Once the parameters are updated, the RNN is retrained using these new parameter setsθ_j'. The fitness of each fish is recalculated by evaluating the performance of the RNN with the perturbed parameters. The fitness function remains defined as the negative of the cross-entropy loss:

$$Fitness(F_j) = -L(\theta_j') \qquad (31)$$

where the loss $L(\theta_j')$ is calculated as follows:

$$L(\theta_j') = -\frac{1}{N} \sum_{i=1}^{N} \sum_{c=1}^{C} y_{ic} log \hat{y}_{ic}' \qquad (32)$$

The updated output probabilities$\hat{y}_{ic}'$are computed based on the new parameters:

$$\hat{y}_{ic}' = softmax\left(W_o^{j'} h_T' + b_o^{j'}\right) \qquad (33)$$

where $h_T'$is the hidden state obtained after processing the sequence with the updated parameters. The fitness evaluation is crucial for determining whether the local perturbations have improved classification performance.

Swimming behavior also incorporates a mechanism to maintain diversity within the fish population. A new random perturbation can be applied to its parameters if a fish's performance does not improve after a defined number of swimming iterations. This ensures that the search process does not converge prematurely to local optima.After updating the fitness values for all fish, the swimming behavior concludes with the selection of the best-performing fish. The best fish represent the most promising parameter sets, which can serve as a foundation for subsequent flying behavior, where more significant parameter updates are made to explore the global search space.

3.6. Backpropagation in RNN

This step focuses on the backpropagation process within the Recurrent Neural Network (RNN), which is essential for updating the parameters after evaluating the network's performance. This process occurs following the swimming behavior, where local adjustments are made to the RNN parameters. Backpropagation through time (BPTT) is utilized to calculate the gradients of the loss function concerning the RNN parameters, enabling effective parameter optimization.The loss function for the RNN, defined in previous steps, is expressed as:

$$L(\theta_j') = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} y_{ic} log \hat{y}_{ic}' \tag{34}$$

where $\hat{y}_{ic}'$is the predicted output probability for sequenceX_i. Backpropagation aims to compute the gradients $\frac{\partial L}{\partial W_h^{j'}}, \frac{\partial L}{\partial W_x^{j'}}, \frac{\partial L}{\partial W_o^{j'}}, \frac{\partial L}{\partial b_h^{j'}}, \frac{\partial L}{\partial b_o^{j'}}$necessary for updating the weights and biases of the RNN.

The backpropagation algorithm starts from the final time step T and moves backward through the sequence. The gradient of the loss concerning the output probabilities can be computed as follows:

$$\delta_T = \hat{y}_i - Y_i \tag{35}$$

where δ_T represents the error at the output layer for sequence X_i. The gradients concerning the output layer weights are calculated:

$$\frac{\partial L}{\partial W_o^{j'}} = \delta_T h_T^T \tag{36}$$

$$\frac{\partial L}{\partial b_o^{j'}} = \delta_T \tag{37}$$

Moving backward in time, the gradients of the hidden state h_t need to be propagated. The error at time step t can be computed using the chain rule:

$$\delta_t = \left(W_o^{j'}\right)^T \delta_{t+1} . f'(h_t) \tag{38}$$

where δ_{t+1} is the error from the subsequent time step, and $f'(h_t)$ is the derivative of the activation function applied at the hidden state h_t.

Next, gradients for the recurrent and input weights are calculated:

$$\frac{\partial L}{\partial W_h^{j'}} = \delta_t h_{t-1}^T \tag{39}$$

$$\frac{\partial L}{\partial W_x^{j'}} = \delta_t x_{it}^T \tag{40}$$

The gradient with respect to the hidden bias is similarly computed:

$$\frac{\partial L}{\partial W_h^{j'}} = \delta_t \tag{41}$$

After calculating the gradients for all time steps, the total gradients for each parameter across all sequences are summed to facilitate batch updates:

$$\frac{\partial L}{\partial W_h^{j'}} = \sum_{t=1}^{T} \frac{\partial L}{\partial W_h^{j'}}\bigg|_t \tag{42}$$

$$\frac{\partial L}{\partial W_x^{j'}} = \sum_{t=1}^{T} \frac{\partial L}{\partial W_x^{j'}}\bigg|_t \tag{43}$$

$$\frac{\partial L}{\partial W_o^{j'}} = \sum_{t=1}^{T} \frac{\partial L}{\partial W_o^{j'}}\bigg|_t \tag{44}$$

$$\frac{\partial L}{\partial b_h^{j'}} = \sum_{t=1}^{T} \frac{\partial L}{\partial b_h^{j'}}\bigg|_t \tag{45}$$

$$\frac{\partial L}{\partial b_o^{j'}} = \sum_{t=1}^{T} \frac{\partial L}{\partial b_o^{j'}}\bigg|_t \tag{46}$$

These accumulated gradients provide the necessary information to update the parameters using an optimization algorithm, such as stochastic gradient descent or Adam. The parameter updates are performed as follows:

$$W_h^j \leftarrow W_h^j - \eta \frac{\partial L}{\partial W_h^j} \quad (47)$$

$$W_x^j \leftarrow W_x^j - \eta \frac{\partial L}{\partial W_x^j} \quad (48)$$

$$W_o^j \leftarrow W_o^j - \eta \frac{\partial L}{\partial b_h^j} \quad (49)$$

$$b_h^j \leftarrow b_h^j - \eta \frac{\partial L}{\partial b_h^j} \quad (50)$$

$$b_o^j \leftarrow b_o^j - \eta \frac{\partial L}{\partial b_o^j} \quad (51)$$

whereη represents the learning rate, which governs the magnitude of the updates. This backpropagation process is repeated iteratively for all training sequences until convergence is achieved, refining the RNN's parameters for improved classification performance.

3.7. Flight Behavior (Global Search in Parameter Space)

This step focuses on the flight behavior, a critical phase in the Exocoetida Optimization (EO) process that facilitates global search within the Recurrent Neural Network (RNN) parameter space. This step allows each fish F_j to make significant jumps in the parameter space, explore distant regions that might yield better solutions than local swimming adjustments made previously.In this phase, the parameters of each fish are updated based on the following equations, which account for both the current position and a random perturbation that guides the exploration:

$$W_h^{j''} = W_h^j + \alpha.\left(W_h^{best} - W_h^j\right) + \delta.rand_h \quad (52)$$

$$W_x^{j''} = W_x^j + \alpha.\left(W_x^{best} - W_x^j\right) + \delta.rand_x \quad (53)$$

$$W_o^{j''} = W_o^j + \alpha.\left(W_o^{best} - W_o^j\right)\delta.rand_o \quad (54)$$

$$b_h^{j''} = b_h^j + \alpha.\left(b_h^{best} - b_h^j\right)\delta.rand_{b_h} \quad (55)$$

$$b_o^{j''} = b_o^j + \alpha.\left(b_o^{best} - b_o^j\right)\delta.rand_{b_o} \quad (56)$$

where $W_h^{best}, W_x^{best}, W_o^{best}, b_h^{best}, b_o^{best}$represent the best parameters obtained from the top-performing fish in the population, reflecting the most effective solution identified thus far. The coefficient$\alpha (where\ 0 < \alpha < 1)$controls the influence of the best-found parameters on the updates, promoting exploration toward better solutions.

The term $\delta \cdot rand$ introduces a random component to the updates, ensuring that the search is not deterministic and that exploration occurs in various directions. The step size δ can be adjusted dynamically based on the iteration count, allowing for more significant jumps in earlier iterations and smaller adjustments as convergence approaches.After updating the parameters, the RNN is retrained using the newly adjusted parameter set $\theta_j'' = \left\{W_h^{j''}, W_x^{j''}, W_o^{j''}, b_h^{j''}, b_o^{j''}\right\}$.The fitness of each fish is re-evaluated by calculating the cross-entropy loss:

$$L\left(\theta_j''\right) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} y_{ic} log \hat{y}_{ic}'' \quad (57)$$

This new loss value will guide the optimization process. The fitness evaluation is crucial for determining whether the global search improved the RNN's classification capabilities.The best fish is updated based on the new fitness scores from the retrained RNN. If the fitness of the current fish is better than the previously recorded best fish, the new best parameters are recorded:

$$Fitness\left(F_j\right) = -LL\left(\theta_j''\right)$$

if $Fitness\left(F_j\right) > Fitness\left(F_{best}\right)$ then $\left\{W_h^{j''}, W_x^{j''}, W_o^{j''}, b_h^{j''}, b_o^{j''}\right\}$ (58)

The successful candidates become the focal point for further exploration in subsequent iterations of the optimization process. The flight behavior enhances the exploration of the parameter space and ensures that promising regions identified by previous iterations are further exploited. This combination of exploration (flight) and exploitation (local swimming) creates a balanced search strategy, enhancing the overall optimization of the RNN parameters.

3.8. RNN Performance Evaluation After Global Search

This step involves evaluating the performance of the Recurrent Neural Network (RNN) after the global search conducted during the flight behavior. This step assesses whether the updated parameters obtained from the flight behavior have improved the RNN's classification capabilities. After the flight behaviour, each fish F_j has an updated set of parameters $\theta_j'' =$

$\left\{W_h^{j''}, W_x^{j''}, W_o^{j''}, b_h^{j''}, b_o^{j''}\right\}$. The RNN is retrained using these new parameters. The input sequences are processed through the RNN to compute the hidden states and outputs. The hidden state at time step t is calculated using the updated recurrent and input weights:

$$h''_t = f\left(W_h^{j''} h''_{t-1} + W_x^{j''} x_{it} + b_h^{j''}\right) \tag{59}$$

The initial hidden state is set to zero:

$$h''_t = 0 \tag{60}$$

The final hidden state h''_T encapsulates the information from the entire sequence X_i. The outout layer uses this hidden state to compute the predicted probabilities for each class:

$$\hat{y}''_i = softmax\left(W_o^{j''} h''_T + b_o^{j''}\right) \tag{61}$$

The softmax function converts the logits into probabilities:

$$\hat{y}''_{ic} = \frac{exp(z''_{ic})}{\sum_{k=1}^{C} exp(z''_{ik})} \tag{62}$$

where $z''_{ic} = \left(W_o^{j''} h''_T + b_o^{j''}\right)_c$ is the logit for class c.

The performance of the RNN is evaluated using the cross-entropy loss function, computed with the updated predictions:

$$L(\theta''_j) = \frac{1}{N}\sum_{i=1}^{N} \delta\left(arg\max_c \hat{y}''_{ic}, \arg\max_c y_{ic}\right) \tag{63}$$

where $\delta(a,b)$ is the kronecker delta function, returning 1 if $a = b$ and 0 otherwise.

The fitness of each fish is updated based on the new loss value:

$$Fitness(F_j) = -L(\theta''_j) + \beta A(\theta''_j) \tag{66}$$

where β is a weighting factor balancing the contribution of accuracy.

The improvement in fitness is calculated by comparing the new fitness with the previous fitness:

$$\Delta Fitness(F_j) = Fitness(F_j) - Fitness_{prev}(F_j) \tag{67}$$

If $\Delta Fitness(F_j) > 0$,the updated parameters are considered superior. The best-performing fishF_{best} is identified:

$$F_{best} = \arg\max_{F_j}\{Fitness(F_j)\} \tag{68}$$

The corresponding parameters $\theta_{best} = \theta''_j$ are noted for guiding future iterations. Statistical measures, such as the confusion matrix C, can be computed to analyze the classification performance in detail. The elements of the confusion matrix are defined as:

$$C_{pq} = \sum_{i=1}^{N} \delta(\hat{y}''_i = p, y_i = q) \tag{69}$$

where p and q range over the class labels.

Receiver Operating Characteristic (ROC) curves and the Area Under the Curve (AUC) can also be calculated to assess the model's ability to distinguish between classes. The True Positive Rate (TPR) and False Positive Rate (FPR) for class c are computed as:

$$TPR_c = \frac{TP_c}{TP_c + FN_c} \tag{70}$$

$$FPR_c = \frac{FP_c}{FP_c + TN_c} \tag{71}$$

where TP_c, FP_c, FN_c and $TN_c$are true positives, false positives, false negatives, and true negatives for class c.

The evaluation results inform whether the global search has effectively enhanced the RNN's performance. If significant improvements are observed, the updated parameters are retained for subsequent optimization steps. Otherwise, alternative strategies, such as adjusting the exploration parameters in the flight behavior, may be considered.

3.9. Collision Avoidance for Maintaining Parameter Diversity

This step focuses on collision avoidance to maintain parameter diversity within the EO-RNN framework. After the global search in Step 7 and performance evaluation in Step 8, the fish population may start converging towards similar parameter sets, potentially leading to premature convergence and loss of diversity. To address this issue, collision avoidance mechanisms are applied to ensure that the fish (candidate solutions) remain sufficiently diverse in the parameter space. For each pair of fish F_j and F_k in the population (where $j \neq k$ and $j, k \in \{1,2,\dots,M\}$), the Euclidean distance between their parameter sets θ_j'' and θ_k'' is calculated:

$$D_{jk} = \left\|\theta_j'' - \theta_k''\right\|_2 \tag{72}$$

where θ_j'' and θ_k'' are the updated parameter vectors of fish F_j and F_k respectively, after the flight behaviour.

A predefined collision threshold ε is set to determine whether two fish are too close in the parameter space. If$D_{jk} < \epsilon$, corrective actions are taken to increase the instance between the fish. When a collision is detected, the position of fish F_j is adjusted using a repulsion mechanism:

$$\theta_j''' = \theta_j'' + \gamma.\left(\theta_j'' - \theta_k''\right).\frac{\epsilon - D_{jk}}{\epsilon} \tag{73}$$

where γ is a scaling factor controlling the magnitude of the adjustment. The term $\left(\theta_j'' - \theta_k''\right)$ represents the direction away from fish F_k, and $\frac{\epsilon - D_{jk}}{\epsilon}$ scales the adjustment based on how close the fish are.

After adjusting the parameters, the new position θ_j''' is checked to ensure it remains within the allowable parameter bounds θ_{min} and θ_{max}, and it ensures that the updated parameters are valid for the RNN model.

$$\theta_j''' = min\left(max\left(\theta_j''', \theta_{min}\right), \theta_{max}\right) \tag{74}$$

The RNN is then retrained using the adjusted parametersθ_j''', and the performance is evaluated:

$$L\left(\theta_j'''\right) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} y_{ic} log\left(\hat{y}_{ic}'''\right) \tag{75}$$

where $\hat{y}_{ic}'''$ are the predicted probabilities obtained using the adjusted parameters.

The fitness of fish F_j is updated accordingly:

$$Fitness\left(F_j\right) = -L\left(\theta_j'''\right) \tag{76}$$

If the fitness improves after collision avoidance, the adjusted parameters are retained; otherwise, the original parametersθ_j'' are restored. To maintain diversity across the entire population, this collision avoidance process is repeated for all pairs of fish. Additionally, a diversity measure Δ can be computed to assess the overall diversity of the population:

$$\Delta = \frac{2}{M(M-1)}\sum_{j=1}^{M}\sum_{k=k+1}^{M} D_{jk} \tag{77}$$

A minimum diversity threshold Δ_{min} can be established to ensure that the population does not become too homogeneous. If $\Delta < \Delta_{min}$, further diversity-enhancing mechanisms, such as random perturbations, can be applied to the fish:

$$\theta_j''' = \theta_j''' + \eta.rand_\theta \tag{78}$$

where η is a small perturbation factor, and $rand_\theta$is a random vector with elements uniformly distributed in $[-1,1]$.

By implementing collision avoidance, the EO-RNN algorithm maintains diverse candidate solutions, enhancing the ability to explore the parameter space effectively and avoid premature convergence to suboptimal solutions.

3.10. Refinement of RNN Parameters Through Multiple Epochs

This step focuses on refining the parameters of the Recurrent Neural Network (RNN) through multiple epochs, building upon the diverse candidate solutions obtained after collision avoidance in section 3.9. The goal is to update the parameters iteratively $\theta_j = \left\{W_h^j, W_x^j, W_o^j, b_h^j, b_o^j\right\}$ to enhance the model's performance on the classification task.The training process involves passing the input sequencesX_i through the RNN for each fish F_j over several epochs $e = 1,2,\dots.,E$. At each epoch, the hidden state $h_t^{(e)}$ at time step t computed using the current parameters:

$$h_t^{(e)} = f\left(W_h^j h_{t-1}^{(e)} + W_x^j x_{it} + b_h^j\right) \tag{79}$$

The initial hidden state is set as:

$$h_t^{(e)} = 0 \tag{80}$$

The output at the final time step T is calculated:

$$\hat{y}_i^{(e)} = softmax\left(W_o^j h_T^{(e)} + b_o^j\right) \tag{81}$$

The loss function for epoch e is defined as:

$$L^{(e)}(\theta_j) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} y_{ic}\, log\left(\hat{y}_{ic}^{(e)}\right) \qquad (82)$$

To prevent overfitting, a regularization term is added:

$$L_{reg}^{(e)}(\theta_j) = L^{(e)}(\theta_j) + \frac{\lambda}{2}\left(\left\|W_h^j\right\|_F^2 + \left\|W_x^j\right\|_F^2 + \left\|W_o^j\right\|_F^2\right) \qquad 83)$$

where λ is the regularization coefficient and $\|.\|_F$ denotes the Frobenius norm.

The gradients of the regularized loss concerning the parameters are computed:

$$\frac{\partial L_{reg}^{(e)}}{\partial W_h^j} = \frac{\partial L^{(e)}}{\partial W_h^j} + \lambda W_h^j \qquad 84)$$

$$\frac{\partial L_{reg}^{(e)}}{\partial W_x^j} = \frac{\partial L^{(e)}}{\partial W_x^j} + \lambda W_x^j \qquad 85)$$

$$\frac{\partial L_{reg}^{(e)}}{\partial W_o^j} = \frac{\partial L^{(e)}}{\partial W_o^j} + \lambda W_o^j \qquad 86)$$

Parameter updates are performed using an optimization algorithm like Adam:

$$m_{\theta_j}^{(e)} = \beta_1 m_{\theta_j}^{(e-1)} + (1-\beta_1)\frac{\partial L_{reg}^{(e)}}{\partial \theta_j} \qquad 87)$$

$$v_{\theta_j}^{(e)} = \beta_2 m_{\theta_j}^{(e-1)} + (1-\beta_2)\left(\frac{\partial L_{reg}^{(e)}}{\partial \theta_j}\right)^2 \qquad 88)$$

$$\widehat{m}_{\theta_j}^{(e)} = \frac{m_{\theta_j}^{(e)}}{1-\beta_1^e} \qquad 89)$$

$$\hat{v}_{\theta_j}^{(e)} = \frac{v_{\theta_j}^{(e)}}{1-\beta_2^e} \qquad 90)$$

$$\theta_j^{(e+1)} = \theta_j^{(e)} - \eta\frac{\widehat{m}_{\theta_j}^{(e)}}{\sqrt{\hat{v}_{\theta_j}^{(e)}} + \epsilon} \qquad 91)$$

where $m_{\theta_j}^{(e)}$ and $v_{\theta_j}^{(e)}$ are the first and second moment estimates, β_1 and $\beta_2$are hyperparameters for the decay rates, η is the learning rate, and ϵ is a small constant to prevent division by zero.

A learning rate schedule may be applied to adjust η over epochs:

$$\eta^{(e)} = \eta_0 . exp(-ke) \qquad 92)$$

where $\eta_0$is the initial learning rate and κ controls the rate of decay.

Early stopping criteria are implemented based on validation loss$L_{val}^{(e)}$: If $L_{val}^{(e)} > L_{val}^{(e-1)}$for p consecutive epochs, stop trainingwhere p is a predefined patience parameter.

The refinement process continues until convergence is achieved, ensuring that the RNN parameters$\theta_j$are optimized for better classification accuracy. This iterative training over multiple epochs allows the model to learn complex temporal patterns and improve generalization to unseen data.

3.11. Selection of Optimal RNN Parameters

This step involves selecting the optimal RNN parameters θ^* from the refined candidate solutions obtained after multiple epochs of training in Step 10. The objective is to identify the parameter set that achieves the best performance on validation data, ensuring robust generalization. First, evaluate each fish F_j in the population $\{F_1, F_2, .., F_M\}$ using the validation dataset. Compute the validation loss $L_{val}(\theta_j)$ for each parameter set $\theta_j = \{W_h^j, W_x^j, W_o^j, b_h^j, b_o^j\}$:

$$L_{val}(\theta_j) = -\frac{1}{N_{val}}\sum_{i=1}^{N_{val}}\sum_{c=1}^{C} y_{ic}\, log\hat{y}_{ic}(\theta_j) \qquad 93)$$

where N_{val} is the number of validation samples, y_{ic} is the true label indicator, and $\hat{y}_{ic}(\theta_j)$ is the predicted probability for class c using parameters θ_j.Next, calculate the validation accuracy$A_{val}(\theta_j)$:

$$A_{val}(\theta_j) = -\frac{1}{N_{val}}\sum_{i=1}^{N_{val}} \delta\left(\arg\max_c \hat{y}_{ic}(\theta_j), \arg\max_c y_{ic}\right) \qquad 94)$$

The function $\delta(a, b)$ equals 1 if $a = b$ and 0 otherwise. This measures the proportion of correctly classified samples. Define a composite fitness function $\Phi(\theta_j) = \alpha L_{val}(\theta_j) - \beta A_{val}(\theta_j)$. A weights$\alpha$ and β balance the importance of minimizing loss and maximizing accuracy. Rank all parameter sets θ_j based on their fitness $\Phi(\theta_j)$. Select the optimal parameters. θ^* that minimize the fitness function:

$$\theta^* = \arg\min_{\theta_j} \Phi(\theta_j) \qquad 95)$$

Cross-validation is performed to ensure the selected parameters are generalized well. Split the training data into K folds and compute the average cross-validated loss $\bar{L}_{cv}(\theta_j)$ and accuracy $\bar{A}_{cv}(\theta_j)$:

$$\bar{L}_{cv}(\theta_j) = \frac{1}{K}\sum_{k=1}^{K} L_{k=1}^{(k)}(\theta_j) \qquad 96)$$

$$\bar{A}_{cv}(\theta_j) = \frac{1}{K}\sum_{k=1}^{K} A_{k=1}^{(k)}(\theta_j) \qquad 97)$$

Update the composite fitness function accordingly:

$$\bar{\Phi}(\theta_j) = \alpha\bar{L}_{cv}(\theta_j) - \beta\bar{A}_{cv}(\theta_j) \qquad (98)$$

Select the optimal parameters based on cross-validated fitness:

$$\theta^* = arg\min_{\theta_j}\bar{\Phi}(\theta_j) \qquad 99)$$

Assess the statistical significance of performance differences using a paired t-test on validation losses between top candidatesθ_j and θ_k:

$$t = \frac{\bar{d}}{s_d/\sqrt{K}} \qquad 100)$$

where $\bar{d}$ is the mean difference in losses $d_k = L_{val}^{(k)}(\theta_j) - L_{val}^{(k)}(\theta_k)$, and s_d is the standard deviation of these differences.

Compute the p-value associated with the t-statistic to determine if the performance improvement is statistically significant. If so, confirmθ^* as the optimal parameter set. Finally, retain the RNN using θ^*on the combined training and validation data to fully utilize available information. The final model is then ready for testing on unseen data, completing the selection of optimal RNN parameters.

Algorithm 1: EO-RNN

Input:

- Dataset with input sequences and corresponding labels
- Number of flying fish (population size)
- Maximum number of iterations (epochs)
- Collision avoidance threshold
- Learning rate and other optimization parameters

Output:

- Optimal RNN parameters
- Trained RNN model

Procedure:

1. Preprocess the dataset to create input sequences and corresponding labels.
2. Initialize the architecture of the RNN with defined layers, units, and activation functions.
3. Generate an initial population of flying fish with random RNN parameters.
4. Evaluate the fitness of each fish by training the RNN with current parameters on the training set.
5. Apply local search (swimming behavior) to fine-tune each fish's parameters.
6. Perform backpropagation to update RNN weights based on the computed gradients.
7. Implement global search (flight behavior) to explore new regions in the parameter space.
8. Evaluate the RNN performance after the global search using validation data.
9. Implement collision avoidance to maintain diversity within the population.
10. Refine the RNN parameters through multiple epochs of training and evaluation.
11. Select the optimal RNN parameters based on validation performance.

Retrain the RNN using the optimal parameters on the combined training and validation data.

4. DATASET

The Gestational Diabetes Mellitus (GDM Data Set)comprises 3,525 rows and 17 columns, offering a comprehensive foundation for gestational diabetes analysis. It captures patient demographics, such as case number, age, and pregnancy details, alongside clinical variables like BMI, HDL, and blood pressure readings. Risk factors, including PCOS, family history, sedentary lifestyle, and prenatal complications, such as large child or birth defect, are incorporated. Additionally, diagnostic attributes like OGTT and hemoglobin levels enable insights into diabetes progression and prediabetic states. This dataset highlights the interrelation of blurred vision and autoimmune disorders with gestational diabetes. The inclusion of diverse attributes facilitates research in disease prediction, risk assessment, and tailored healthcare solutions, particularly in understanding diabetic pregnancy complexities.

TABLE 1: Gdm Dataset Features Description

Feature Name	Description
Case Number	Unique identifier for the patient in the dataset.
Age	Patient's age recorded at the time of medical assessment.
Number of	Count of pregnancies experienced by the

Pregnancies	individual.
Gestation	Length of pregnancy measured in weeks.
BMI	Body Mass Index calculated from weight and height.
HDL	Level of High-Density Lipoprotein, a key cholesterol marker.
Family History	Indicator of genetic predisposition to diabetes.
Unexplained Prenatal Loss	Record of past miscarriages without identified medical reasons.
Large Child or Birth Defect	Presence of birth complications like macrosomia or defects.
PCOS	Diagnosis of Polycystic Ovary Syndrome, linked to insulin resistance.
Systolic BP	Blood pressure's systolic measurement, indicating heart activity.
Diastolic BP	Blood pressure's diastolic measurement, indicating heart rest.
OGTT	Test results from the Oral Glucose Tolerance Test for diabetes screening.
Hemoglobin	Hemoglobin concentration, reflecting oxygen-carrying capacity of the blood.
Sedentary Lifestyle	Assessment of the individual's physical activity levels.
Prediabetics	Identification of individuals in the prediabetic stage.

5. RESULTS AND DISCUSSION

5.1. Classification Accuracy and F-Measure Analysis

Classification accuracy evaluates the proportion of correctly classified instances among the total, reflecting the model's overall performance in making accurate predictions. The F-measure provides a harmonic mean of precision and recall, ensuring a balance between false positives and false negatives. These metrics are critical for assessing the effectiveness of models in LADA prediction, where precision and reliability are essential to avoid misdiagnoses and missed cases.

Figure 1 compares the performance of three models—HANN, EDLT, and EO-RNN—using classification accuracy and F-measure as evaluation metrics. For classification accuracy, HANN exhibits the lowest value of 49.833%, indicating its limited effectiveness in correctly classifying instances. EDLT improves upon this with an accuracy of 62.003%, demonstrating better predictive performance. EO-RNN surpasses both models significantly, achieving an accuracy of 95.364%, showcasing its robust architecture and optimization strategies that enhance classification precision.A similar trend is observed for the F-measure. HANN records the lowest F-measure of 49.828%, reflecting challenges in balancing precision and recall. EDLT demonstrates improvement, achieving an F-measure of 62.087%, highlighting its better handling of false positives and false negatives. EO-RNN achieves the highest F-measure of 95.467%, indicating its capability to maintain a high balance between precision and recall, making it the most reliable model among the three.The results highlight EO-RNN's superior performance in both classification accuracy and F-measure, demonstrating its effectiveness in LADA prediction. The findings emphasize the critical role of advanced optimization techniques and model architectures in improving predictive reliability for chronic diseases, providing a reliable framework for clinical applications.

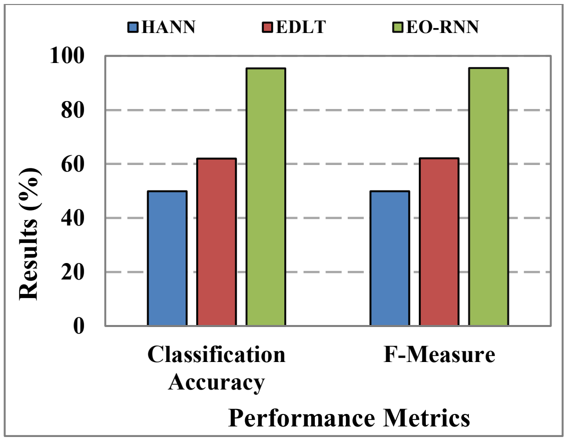

Figure 1. Classification Accuracy And F-Measure

5.2. False Discovery and Omission Analysis

False discovery rate (FDR) evaluates the proportion of false positive classifications among all positive predictions, reflecting a model's precision. A higher FDR indicates an increased likelihood of incorrect classifications, leading to unnecessary interventions or misdiagnoses. False omission rate (FOR) quantifies the proportion of false negative classifications among all negative predictions, assessing the extent of missed actual cases. A lower FOR is essential in healthcare contexts, where undetected conditions like Latent Autoimmune Diabetes in Adults (LADA) can delay treatment. Both metrics are crucial for validating the reliability and robustness of predictive models in clinical diagnostics.

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

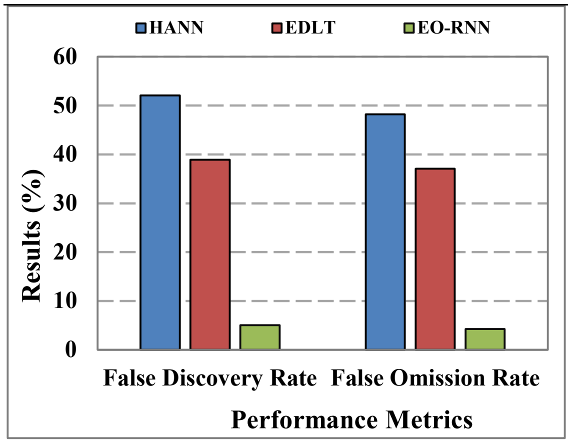

Figure 2. False Discovery And False Omission

Figure 2 depicts the FDR and FOR performance of three models: HANN, EDLT, and EO-RNN. HANN records the highest FDR at 52.012%, highlighting its limited precision and susceptibility to false positives. EDLT achieves a reduction in FDR to 38.901%, demonstrating moderate improvement in precision over HANN. EO-RNN achieves a significant reduction, with the lowest FDR at 4.998%, indicating superior performance in minimizing false positive classifications. This result reflects EO-RNN's capacity to deliver precise predictions and reduce unnecessary diagnostic errors.The FOR analysis aligns with these observations. HANN exhibits the highest FOR at 48.175%, indicating significant challenges in detecting true negative cases. EDLT reduces the FOR to 37.059%, showing improved reliability in minimizing missed actual cases. EO-RNN outperforms both, achieving the lowest FOR at 4.254%, demonstrating its efficiency in capturing true positives while maintaining low false negative rates. These outcomes underscore the effectiveness of EO-RNN in balancing prediction errors, establishing it as the most reliable model among the three. By minimizing both FDR and FOR, EO-RNN contributes to enhanced diagnostic accuracy and reliability, making it suitable for predictive applications in LADA and similar healthcare challenges.

5.3. Positive and Negative Likelihood Analysis

Positive likelihood ratio (PLR) quantifies the ratio between the probability of a positive test result in individuals with the condition and the probability of the same result in individuals without the condition. A higher PLR value indicates that the model is effective in correctly identifying true positives, which is crucial for accurate diagnostics. Negative likelihood ratio (NLR) measures the ratio between the probability of a negative result in individuals with the condition and the probability of a negative result in individuals without it. Lower NLR values suggest a reduced chance of missing true positive cases, which is vital for comprehensive detection.

Table 2. Positive And Negative Likelihood

Algorithms	Positive Likelihood Ratio (%)	Negative Likelihood Ratio (%)
HANN	0.996	1.004
EDLT	1.615	0.605
EO-RNN	18.348	0.043

Table 2 presents a comparison of PLR and NLR across three algorithms: HANN, EDLT, and EO-RNN. HANN records a PLR of 0.996, reflecting a suboptimal performance in distinguishing true positives from false positives. The corresponding NLR is 1.004, indicating considerable difficulty in minimizing false negatives, thus impacting its reliability in LADA prediction. EDLT shows improvement with a PLR of 1.615, signifying enhanced capability in identifying positive instances accurately. The NLR value of 0.605 also demonstrates EDLT's moderate improvement in reducing missed diagnoses, suggesting better reliability than HANN.EO-RNN outperforms both models significantly, achieving a PLR of 18.348, indicating a strong ability to correctly classify true positives, thereby enhancing diagnostic accuracy. The NLR for EO-RNN is 0.043, which is considerably lower than both HANN and EDLT, demonstrating its exceptional capacity to minimize false negatives. This low NLR value underscores EO-RNN's effectiveness in providing comprehensive diagnostic coverage by reducing the likelihood of missed LADA cases.The comparison indicates that EO-RNN excels in both metrics, showcasing its reliability for accurate detection and prediction of LADA, making it the preferred model for healthcare applications requiring high diagnostic precision and reliability.

5.3. Critique

In reflecting upon the findings of this study, several limitations must be considered. The application of Exocoetidae Optimization in RNNs, while innovative, was tested within a controlled dataset which may not fully capture the diversity of clinical scenarios in which LADA is diagnosed. This could affect the generalizability of the results to broader clinical applications. Additionally, the computational intensity of the optimization process may limit its feasibility in real-time diagnostic settings. Future research should explore the scalability of this approach in larger, more heterogeneous datasets to validate its effectiveness across varied clinical conditions. Further, comparative studies with other bio-inspired algorithms could illuminate areas for enhancement in optimization efficiency and accuracy. Such investigations are crucial for advancing the application of machine learning in medical diagnostics and ensuring that these technologies deliver robust, reliable tools for healthcare providers.

5.4. Distinctions from Prior Work

As established in the initial phases of this study, the accurate classification and diagnosis of Latent Autoimmune Diabetes in Adults (LADA) have remained a significant challenge due to the subtle and overlapping clinical features of the disease with type 2 diabetes. Traditional diagnostic models often rely on standard machine learning algorithms which do not account for the complex, temporal dynamics of autoimmune responses in LADA patients. This study has introduced a pioneering approach by implementing the Exocoetidae Optimization-Inspired Recurrent Neural Network (EO-RNN). Unlike previous methods, the EO-RNN capitalizes on the dynamic adaptation strategies of Exocoetidae (flying fish), enhancing the RNN's parameter optimization process for better handling of temporal and sequential data in clinical diagnostics. This bio-inspired enhancement significantly improves the model's ability to discern nuanced patterns in patient data that are indicative of LADA. Comparatively, the EO-RNN demonstrated a classification accuracy of 95.36% and an F-measure of 95.47%, figures that significantly surpass those achieved by conventional RNNs without Exocoetidae Optimization. These metrics not only showcase substantial improvements over existing models but also highlight the efficacy of incorporating bio-inspired algorithms in medical diagnostics. The achievements of this study are particularly notable in their potential to transform the diagnostic landscape for autoimmune diabetes. By providing a more accurate, reliable, and efficient tool for detecting LADA, the EO-RNN model supports the advancement of personalized medicine and enhances the quality of care for diabetes patients. These contributions not only fulfill the identified need for better diagnostic tools but also pave the way for future research into bio-inspired machine learning applications in healthcare.

6. CONCLUSION

This study addressed the critical challenge of accurately diagnosing Latent Autoimmune Diabetes in Adults (LADA) by developing an Exocoetidae Optimization-Inspired Recurrent Neural Network (EO-RNN). The novel application of bio-inspired optimization strategies has not only enhanced the RNN's capability to interpret complex, temporal medical data but has also significantly improved diagnostic accuracy, achieving a classification accuracy of 95.36% and an F-measure of 95.47%. The scientific contributions of this research are manifold. Primarily, it introduces a pioneering optimization technique that can be leveraged in various medical diagnostic applications, underscoring the versatility and potential of bio-inspired algorithms in machine learning. Furthermore, by significantly boosting the accuracy of LADA detection, this study contributes to the field of precision medicine, offering pathways for more tailored and effective patient treatment plans. The implications of these advancements are profound, promising to enhance the quality of life for patients through more accurate and timely diagnoses. Looking forward, this work lays a solid foundation for further research into the application of bio-inspired optimization in other complex disease areas, potentially transforming diagnostic methodologies across the medical spectrum. By pushing the boundaries of how machine learning can be harnessed to address healthcare challenges, this research not only contributes to the scientific community but also paves the way for future innovations that could redefine medical diagnostics.

REFERENCES:

[1] R. M. Wasserman, S. R. Patton, M. A. Clements, D. Guffey, D. D. Schwartz, and B. J. Anderson, “Risky self-management behaviors in adolescents with type 1 diabetes: Measurement validation for the Diabetes-Specific Risk-Taking Inventory,” *Pediatr Diabetes*, vol. 23, no. 7, pp. 1113–1121, 2022, doi: 10.1111/pedi.13387.

[2] A. S. Kortegaard, R. B. Rokkjær, H. M. H. Harboe, S. Lund, A. Andersen, and M. Bohl, “A group-based intervention for diabetes-related emotional distress among emerging adults with type 1 diabetes: A pilot study,”

Eur J Intern Med, 2024, doi: 10.1016/j.ejim.2024.06.002.

[3] 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.

[4] J. A. Carter, C. S. Long, B. P. Smith, T. L. Smith, and G. L. Donati, "Combining elemental analysis of toenails and machine learning techniques as a non-invasive diagnostic tool for the robust classification of type-2 diabetes," *Expert Syst Appl*, vol. 115, pp. 245–255, Jan. 2019, doi: 10.1016/j.eswa.2018.08.002.

[5] J. T. K. Cheung *et al.*, "Combined associations of family history and self-management with age at diagnosis and cardiometabolic risk in 86,931 patients with type 2 diabetes: Joint Asia Diabetes Evaluation (JADE) Register from 11 countries," *BMC Med*, vol. 20, no. 1, 2022, doi: 10.1186/s12916-022-02424-y.

[6] N. P. Tigga and S. Garg, "Prediction of Type 2 Diabetes using Machine Learning Classification Methods," *Procedia Comput Sci*, vol. 167, pp. 706–716, 2020, doi: 10.1016/j.procs.2020.03.336.

[7] N. Taya *et al.*, "Change in fatty acid composition of plasma triglyceride caused by a 2 week comprehensive risk management for diabetes: A prospective observational study of type 2 diabetes patients with supercritical fluid chromatography/mass spectrometry-based semi-targe," *J Diabetes Investig*, vol. 14, no. 1, pp. 102–110, 2023, doi: 10.1111/jdi.13924.

[8] R. J, "Meticulous Elephant Herding Optimization based Protocol for Detecting Intrusions in Cognitive Radio Ad Hoc Networks," *International Journal of Emerging Trends in Engineering Research*, vol. 8, no. 8, pp. 4548–4554, 2020, doi: 10.30534/ijeter/2020/82882020.

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

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

[11] B. Paul and B. Karn, "Diabetes Mellitus Prediction using Hybrid Artificial Neural Network," in *2021 IEEE Bombay Section Signature Conference (IBSSC)*, 2021, pp. 1–5. doi: 10.1109/IBSSC53889.2021.9673397.

[12] P. Singh, S. Silakari, and S. Agrawal, "An Efficient Deep Learning Technique for Diabetes Classification and Prediction Based on Indian Diabetes Dataset," in *2023 3rd International Conference on Technological Advancements in Computational Sciences (ICTACS)*, 2023, pp. 487–491. doi: 10.1109/ICTACS59847.2023.10390518.

[13] D. Chhabra, M. Juneja, and G. Chutani, "An Efficient Ensemble-based Machine Learning approach for Predicting Chronic Kidney Disease," *Curr Med Imaging*, vol. 20, 2024, doi: https://doi.org/10.2174/1573405620666230508104538.

[14] B. Amma N.G., "En-RfRsK: An ensemble machine learning technique for prognostication of diabetes mellitus," *Egyptian Informatics Journal*, vol. 25, p. 100441, 2024, doi: https://doi.org/10.1016/j.eij.2024.100441.

[15] I. Jannoud, M. Z. Masoud, Y. Jaradat, A. Manaserah, and D. Zaidan, "A Multi-Layered Hybrid Machine Learning Algorithm (MLHA) for Type II Diabetes Classification," *Procedia Comput Sci*, vol. 237, pp. 445–452, 2024, doi: https://doi.org/10.1016/j.procs.2024.05.126.

[16] W. Zhang *et al.*, "Identification of key biomarkers for predicting atherosclerosis progression in polycystic ovary syndrome via bioinformatics analysis and machine learning," *Comput Biol Med*, vol. 183, p. 109239, 2024, doi: https://doi.org/10.1016/j.compbiomed.2024.109239.

[17] A. S. Zamani, A. H. A. Hashim, A. S. A. Shatat, Md. M. Akhtar, M. Rizwanullah, and S. S. I. Mohamed, "Implementation of machine learning techniques with big data and IoT to create effective prediction models for health informatics," *Biomed Signal Process Control*, vol. 94, p. 106247, 2024, doi: https://doi.org/10.1016/j.bspc.2024.106247.

[18] A. T. Villikudathil, D. H. Mc Guigan, and A. English, "Exploring metformin monotherapy response in Type-2 diabetes: Computational insights through clinical, genomic, and

proteomic markers using machine learning algorithms," *Comput Biol Med*, vol. 171, p. 108106, 2024, doi: https://doi.org/10.1016/j.compbiomed.2024.108106.

[19] D. P. Upadhyaya, Y. Tarabichi, K. Prantzalos, S. Ayub, D. C. Kaelber, and S. S. Sahoo, "Machine learning interpretability methods to characterize the importance of hematologic biomarkers in prognosticating patients with suspected infection," *Comput Biol Med*, vol. 183, p. 109251, 2024, doi: https://doi.org/10.1016/j.compbiomed.2024.109251.

[20] C. Díez-Sanmartín, A. Sarasa Cabezuelo, and A. Andrés Belmonte, "Ensemble of machine learning techniques to predict survival in kidney transplant recipients," *Comput Biol Med*, vol. 180, p. 108982, 2024, doi: https://doi.org/10.1016/j.compbiomed.2024.108982.

[21] S. A. Tiruneh, D. L. Rolnik, H. J. Teede, and J. Enticott, "Prediction of pre-eclampsia with machine learning approaches: Leveraging important information from routinely collected data," *Int J Med Inform*, vol. 192, p. 105645, 2024, doi: https://doi.org/10.1016/j.ijmedinf.2024.105645.

[22] A. Zignoli, K. Skroce, D. J. Lipman, and H. C. Zisser, "Personalized nutrition and machine-learning: Exploring the scope of continuous glucose monitoring in healthy individuals in uncontrolled settings," *Biomed Signal Process Control*, vol. 90, p. 105809, 2024, doi: https://doi.org/10.1016/j.bspc.2023.105809.

[23] J. L. Wesson and N. Smith, "A machine learning model to predict preeclampsia in pregnant women," *Procedia Comput Sci*, vol. 239, pp. 1645–1652, 2024, doi: https://doi.org/10.1016/j.procs.2024.06.341.

[24] F. Ali, M. Khalid, A. Almuhaimeed, A. Masmoudi, W. Alghamdi, and A. Yafoz, "IP-GCN: A deep learning model for prediction of insulin using graph convolutional network for diabetes drug design," *J Comput Sci*, vol. 81, p. 102388, 2024, doi: https://doi.org/10.1016/j.jocs.2024.102388.

[25] Md. M. Hassan *et al.*, "Efficient prediction of coronary artery disease using machine learning algorithms with feature selection techniques," *Computers and Electrical Engineering*, vol. 115, p. 109130, 2024, doi: https://doi.org/10.1016/j.compeleceng.2024.109130.

[26] D. Pawade, D. Bakhai, T. Admane, R. Arya, Y. Salunke, and Y. Pawade, "Evaluating the Performance of Different Machine Learning Models for Metabolic Syndrome Prediction," *Procedia Comput Sci*, vol. 235, pp. 2932–2941, 2024, doi: https://doi.org/10.1016/j.procs.2024.04.277.

[27] J. Su, "Construction of a Computer Network Performance Prediction Optimization Model Based on Machine Learning Algorithms," *Procedia Comput Sci*, vol. 243, pp. 364–371, 2024, doi: https://doi.org/10.1016/j.procs.2024.09.045.

[28] N. Al Mudawi, "Developing a Model for Parkinson's Disease Detection Using Machine Learning Algorithms," *Computers, Materials and Continua*, vol. 79, no. 3, pp. 4945–4962, 2024, doi: https://doi.org/10.32604/cmc.2024.048967.

[29] R. Khushal and U. Fatima, "Fuzzy machine learning logic utilization on hormonal imbalance dataset," *Comput Biol Med*, vol. 174, p. 108429, 2024, doi: https://doi.org/10.1016/j.compbiomed.2024.108429.

[30] A. Procopio *et al.*, "Physiological model-based machine learning for classifying patients with binge-eating disorder (BED) from the Oral Glucose Tolerance Test (OGTT) curve," *Comput Methods Programs Biomed*, vol. 258, p. 108477, 2025, doi: https://doi.org/10.1016/j.cmpb.2024.108477.

[31] M. Ibrahim, A. Beneyto, I. Contreras, and J. Vehi, "An ensemble machine learning approach for the detection of unannounced meals to enhance postprandial glucose control," *Comput Biol Med*, vol. 171, p. 108154, 2024, doi: https://doi.org/10.1016/j.compbiomed.2024.108154.

[32] Y. Wang *et al.*, "Prediction of viral pneumonia based on machine learning models analyzing pulmonary inflammation index scores," *Comput Biol Med*, vol. 169, p. 107905, 2024, doi: https://doi.org/10.1016/j.compbiomed.2023.107905.

[33] R. Jaganathan and R. Vadivel, "Intelligent Fish Swarm Inspired Protocol (IFSIP) for Dynamic Ideal Routing in Cognitive Radio Ad-Hoc Networks," *International Journal of Computing and Digital Systems*, vol. 10, no. 1, pp. 1063–1074, 2021, doi: 10.12785/ijcds/100196.

[34] P. Menakadevi and J. Ramkumar, "Robust Optimization Based Extreme Learning

Machine for Sentiment Analysis in Big Data," in *2022 International Conference on Advanced Computing Technologies and Applications, ICACTA 2022*, 2022. doi: 10.1109/ICACTA54488.2022.9753203.

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

[36] 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

[37] 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*, 2025, pp. 713–723. doi: 10.1007/978-981-97-6106-7_43.

[38] 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," *International Journal of Computer Networks and Applications*, vol. 10, no. 4, pp. 668–687, 2023, doi: 10.22247/ijcna/2023/223319.

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

[40] 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*, 2024, pp. 345–358. doi: 10.1007/978-981-97-3690-4_26.

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

[42] 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*, 2024, pp. 117–137. doi: 10.4018/979-8-3693-0724-3.ch007.

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

[44] M. P. Swapna and J. Ramkumar, "Multiple Memory Image Instances Stratagem to Detect Fileless Malware," in *Communications in Computer and Information Science*, 2024, pp. 131–140. doi: 10.1007/978-3-031-59100-6_11.

[45] 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

[46] 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*, 2022. doi: 10.1145/3590837.3590907.

[47] D. Jayaraj, J. Ramkumar, M. Lingaraj, and B. Sureshkumar, "AFSORP: Adaptive Fish Swarm Optimization-Based Routing Protocol for Mobility Enabled Wireless Sensor Network," *International Journal of Computer Networks and Applications*, vol. 10, no. 1, pp. 119–129, 2023, doi: 10.22247/ijcna/2023/218516.

[48] M. Lingaraj, T. N. Sugumar, C. S. Felix, and J. Ramkumar, "Query aware routing protocol for mobility enabled wireless sensor network," *International Journal of Computer*

Networks and Applications, vol. 8, no. 3, pp. 258–267, 2021, doi: 10.22247/ijcna/2021/209192.

[49] 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, 2021, doi: 10.1007/s11277-021-08495-z.

[50] J. Ramkumar and R. Vadivel, “Improved Wolf prey inspired protocol for routing in cognitive radio Ad Hoc networks,” *International Journal of Computer Networks and Applications*, vol. 7, no. 5, pp. 126–136, 2020, doi: 10.22247/ijcna/2020/202977.

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

[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*, 2023, pp. 17–27. doi: 10.1007/978-981-19-8353-5_2.

[53] 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*, 2023. doi: 10.1109/ELEXCOM58812.2023.10370559.

[54] 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.

[55] R. Jaganathan, S. Mehta, and R. Krishan, *Intelligent Decision Making Through Bio-Inspired Optimization*. 2024. doi: 10.4018/979-8-3693-2073-0.

[56] 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

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

[58] 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

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

[60] 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*, 2017, pp. 145–153. doi: 10.1007/978-981-10-3874-7_14.

[61] 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

[62] R. Jaganathan and V. Ramasamy, “Performance modeling of bio-inspired routing protocols in Cognitive Radio Ad Hoc Network to reduce end-to-end delay,” *International Journal of Intelligent Engineering and Systems*, vol. 12, no. 1, pp. 221–231, 2019, doi: 10.22266/IJIES2019.0228.22.

[63] 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, 2025, pp. 141–153. doi: 10.1007/978-981-97-6710-6_12.

[64] 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

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

[66] 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)," *International Journal of Engineering Trends and Technology*, vol. 73, no. 1, pp. 410–430, 2025, doi: 10.14445/22315381/IJETT-V73I1P135.

[67] 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

[68] 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)," *International Journal of Computer Networks and Applications*, vol. 11, no. 5, pp. 726–748, 2024, doi: 10.22247/ijcna/2024/45.

[69] 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*, 2022. doi: 10.1109/ICACTA54488.2022.9752899.

[70] 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*, 2019, pp. 109–121. doi: 10.4018/978-1-7998-1090-2.ch006.

[71] 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