International Journal of Engineering Trends and Technology
ISSN: 2231–5381 / https://doi.org/10.14445/22315381/IJETT-V72I12P128
Volume 72 Issue 12, 330-348, December 2024
© 2024 Seventh Sense Research Group®

Original Article

Proficient Red Deer Optimization-based Relevance Vector Machine (PRDO-RVM) for Elevated Intrusion Detection System in MANET

M. Sasikumar[1], K. Rohini[2]

[1]School of Computing Science, Vels Institute of Science, Technology &Advanced Studies (VISTAS) Chennai, Tamilnadu, India.
[2]Department of Information Technology, School of Computing Science, Vels Institute of Science, Technology & Advanced Studies (VISTAS), Chennai, Tamilnadu, India.

[1]Corresponding Author : marksasiphd@gmail.com

Received: 01 June 2024 Revised: 07 October 2024 Accepted: 14 November 2024 Published: 25 December 2024

***Abstract* - *Mobile Ad-hoc Networks (MANETs) are wireless networks composed of autonomous mobile devices that communicate with each other without relying on a centralized infrastructure. MANET security issues can compromise data confidentiality, integrity, and availability, highlighting the critical need for robust security mechanisms. Intrusion Detection Systems (I.D.S.) are crucial in identifying and mitigating security threats in MANETs. Designing effective I.D.S. for MANETs is inherently challenging due to these networks' dynamic and resource-constrained nature. To address these challenges, this research proposes the Proficient Red Deer Optimization-based Relevance Vector Machine (PRDO-RVM) for intrusion detection in MANETs. PRDO-RVM leverages the sparsity-inducing properties of Relevance Vector Machine (RVM) and the efficient optimization capabilities of Red Deer Optimization to achieve accurate and efficient intrusion detection in dynamic network environments. By effectively identifying and classifying intrusions, PRDO-RVM enhances the security posture of MANETs, mitigating the risks posed by malicious actors and ensuring the integrity and availability of network communications. Using the NSK-KDD dataset, PRDO-RVM is evaluated for its effectiveness in detecting intrusions in MANETs. The results demonstrate the superior classification accuracy and efficiency of PRDO-RVM compared to existing I.D.S. solutions, affirming its potential as a reliable and scalable security mechanism for MANETs.*

***Keywords* - *Intrusion, MANET, Optimization, Security, Krill herd, Random Forest.*

1. Introduction

A Mobile Ad-hoc Network (MANET) is a decentralized wireless network comprising mobile nodes communicating without requiring a fixed infrastructure. MANETs are, hence, highly dynamic and self-organizing and are ideal for situations where traditional networks cannot be practically employed, such as military operations and disaster recovery or collaborative mobile applications. [1]. In a MANET, each node can act as a host and a router, allowing for direct peer-to-peer communication. These networks rely on wireless communication protocols like Wi-Fi or Bluetooth, ensuring adaptability and flexibility in various environments. MANETs have significant advantages, including rapid deployment, scalability, and robustness in the face of node mobility or network topology changes [2]. Their use extends beyond military applications, including emergency response, vehicular communication, and IoT systems. Challenges within MANETs include security concerns due to their openness, limited power supply, and the potential for network partitioning. Researchers have explored solutions such as intrusion detection, energy-efficient communication, and improved protocols to address these issues [3]. Intrusion Detection Systems (I.D.S.) are integral to network security, helping organizations safeguard network resources' confidentiality, integrity, and availability. Their primary role is detecting and responding to real-time security threats [4]. By identifying unauthorized access attempts, malware infections, and denial-of-service attacks, I.D.S. contributes significantly to maintaining network security. They also assist organizations in adhering to data protection regulations by generating comprehensive logs and reports of security incidents [5]. In a constantly evolving threat landscape, I.D.S. is adapting to remain effective, incorporating artificial intelligence, machine learning, and automated threat response to be ahead of evolving challenges and improve accuracy in detecting and responding to security threats [6]. In the context of MANETs, I.D.S. is an indispensable tool for securing the network against many potential threats. MANETs are decentralized, self-organizing networks where nodes communicate directly, and as a result, they are susceptible to

This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/)

various security vulnerabilities [7]. I.D.S. for MANETs monitors and analyzes the network and its traffic to detect intrusion by acting like vigilant gatekeepers. It uses heuristic analysis, anomaly detection, and signature-based methods to detect unauthorised access and other malicious routing attacks [8]. The dynamic nature of MANETs, characterized by node mobility and changing network topologies, demands I.D.S. solutions that can adapt and operate effectively under these conditions [9].

Bio-inspired computation is the inspiration from natural systems and biological processes to support and enhance computational procedures and models. Mimicking behaviors observed in evolution, swarming, foraging, and cellular growth, this field addresses complex problem-solving in optimization, pattern recognition, and adaptive learning. Techniques such as genetic algorithms, bee colony optimization, and swarm optimizations replicate the efficiency and adaptability of natural organisms. Bio-inspired computing has advanced various domains, including robotics, machine learning, and engineering, by providing innovative solutions rooted in the intelligence of nature.

1.1. Problem Statement

Communication among I.D.S. nodes within the complex environment of MANETs must be robustly secure to safeguard against eavesdropping and tampering with intrusion alerts and collaboration messages. This added layer of security presents a multifaceted challenge due to the highly dynamic, decentralized nature of MANETs. Ensuring the confidentiality and integrity of communication while minimizing overhead is paramount for effective intrusion detection. The problem encompasses developing secure mechanisms, encryption protocols, and authentication techniques to protect the sensitive data exchanged among I.D.S. nodes without introducing excessive latency or resource overhead. Addressing this challenge is essential to the overall security and reliability of intrusion detection in MANETs, as the secure exchange of vital intrusion-related data is foundational to network defense and threat response.

1.2. Motivation

The motivation for addressing secure communication in I.D.S. for MANETs stems from the fundamental need for confidentiality, integrity, and trust in these dynamic and decentralized networks. MANETs are deployed in sensitive contexts like military operations, disaster response, and healthcare, where intrusion alerts and collaboration among I.D.S. nodes are pivotal for threat response and network security. Ensuring the confidentiality of these messages is essential to prevent adversaries from gaining insights into network defense strategies or manipulating alerts for malicious purposes. The risk of eavesdropping, data tampering, and message interception is significant. Therefore, developing secure communication mechanisms, encryption protocols, and authentication techniques is crucial to protect the exchange of vital intrusion-related data while minimizing network overhead and strengthening the trustworthiness of intrusion detection in MANETs.

1.3. Research Objective

The core research objective of this study, "Efficacious Krill Herd Optimized Random Forests," is to develop and evaluate an Intrusion Detection System (I.D.S.) within the dynamic context of Mobile Ad Hoc Networks (MANETs) using the "Efficacious Krill Herd Optimized Random Forests" classification algorithm. This research is primarily driven by the critical need to establish a highly accurate and robust intrusion detection solution adept at addressing the dynamic and decentralized nature of MANETs. The goal is to craft an I.D.S. that precisely identifies security threats, optimizing the network's security posture. The anticipated outcome is an innovative intrusion detection system capable of effectively distinguishing between authentic security threats and benign network activities, ultimately reinforcing security in mission-critical contexts such as military operations, disaster response, and healthcare deployments.

2. Literature Review

"PSO - IDS" [10] introduces a feature selection approach that harnesses multi-objective Particle Swarm Optimization (PSO) to enhance I.D.S. The innovative method optimizes the selection of relevant features, effectively reducing the dimensionality of the dataset while retaining critical information. Doing so enhances the performance of I.D.S. by improving accuracy, reducing computational overhead, and bolstering network security in resource-constrained IoT environments. "Federated Learning for Heterogeneous Network IDS" [11] presents a novel I.D.S. approach that adapts to the challenges of heterogeneous networks. It employs a stacked-unsupervised federated learning strategy, allowing various networks with differing characteristics to improve I.D.S. capabilities collaboratively. This federated learning approach promotes model generalization, thus enhancing the detection of network intrusions across diverse network topologies, thereby strengthening security in complex and varied network environments.

"Framework for Cyber Attack Detection" [12] introduces a comprehensive framework to facilitate cyber-attack detection by efficiently classifying I.D.S. datasets. The framework streamlines the data processing and classification tasks crucial for I.D.S. Providing a structured and systematic approach aids in developing accurate and efficient I.D.S., ensuring that a wide range of cyber threats are effectively identified and mitigated. "Anomaly-Based IoT IDS with CNN and MOECSA" [13] leverages a Convolutional Neural Network (CNN) and a multi-objective enhanced Capuchin Search Algorithm (MOECSA) to improve threat detection. The CNN is adept at identifying complex patterns within IoT data, while the MOECSA optimizes the I.D.S. process. This combination enhances network security by efficiently identifying anomalies and adapting to the dynamic nature of IoT networks.

“Hybrid IDS for Wireless IoT Networks” [14] proposes a hybrid I.D.S. system designed specifically for wireless IoT networks. This system combines deep learning algorithms with I.D.S. to enhance security. Its essential contribution lies in the improved detection of security threats in IoT environments, as deep learning techniques are applied to analyze data patterns and detect intrusions. This hybrid approach enhances network security by capitalizing on the strengths of both I.D.S. and deep learning. “GAN-Based Synthetic Attack Data Generation” [15] introduces a model that employs Generative Adversarial Networks (GANs) to generate synthetic attack data for I.D.S. The critical contribution is creating synthetic data, which accurately simulates various attack scenarios. This synthetic data enriches I.D.S.'s training and validation processes, ensuring that models are well-prepared to identify diverse intrusion types.

“Ensemble Learning for IDS” [16] specializes in creating influential ensembles of deep neural networks for I.D.S., primarily aiming to enhance the precision and consistency of I.D.S. This research significantly bolsters network security measures' precision and overall efficacy by constructing a collaborative ensemble of models working to identify security threats. “AutoML-Based Ensemble for Network IDS” [17] presents an optimized ensemble prediction model using AutoML and a soft voting classifier for network I.D.S. By automating the model development process and utilizing ensemble classifiers, this system effectively improves the precision and efficacy of I.D.S., offering a higher degree of protection against security threats. “Moth-Flame Optimization for ITS” [18] explores the application of Moth-Flame Optimization in ensemble classification for I.D.S. within Intelligent Transport Systems (ITS) in smart cities. Employing Moth-Flame Optimization in ensemble classifiers significantly enhances the correctness and consistency of I.D.S. in the context of intelligent transport systems, contributing to safer and more secure urban environments.

“Cooperative IDS with Deep Q Network in MEC” [19] introduces the concept of task offloading for a cooperative I.D.S. based on a Deep Q Network in Mobile Edge Computing (MEC). The primary objective is optimizing I.D.S. resource allocation in edge computing environments. By utilizing the Deep Q Network approach, this system enhances the efficiency and accuracy of I.D.S., enabling collaborative intrusion detection while minimizing resource consumption in MEC environments. “Enhanced Empirical-Based IDS” [20] introduces an improved empirical-based component analysis approach for I.D.S. in Wireless Sensor Networks (WSN), with a focus on enhancing the accuracy and robustness of I.D.S. By applying advanced component analysis techniques, this system effectively improves the precision and effectiveness of I.D.S. in identifying and mitigating security threats in WSNs. “Hybrid Data-Driven Model” [21] presents a hybrid data-driven model for I.D.S. in Vehicular Ad-hoc Networks (VANETs), emphasizing the enhancement of security and threat detection capabilities within VANETs. By combining various data-driven techniques and models, this hybrid approach further improves the precision and effectiveness of I.D.S. in the context of VANETs, ensuring safer and more secure vehicular communication.

“LSO-FFNN” [22] combines the principles of swarm intelligence with neural network architecture for optimization tasks. LSO-FFNN leverages the collective behavior of locust swarms to iteratively update neural network parameters, aiming to achieve optimal solutions. Despite its potential, LSO-FFNN may face challenges such as slow convergence, limited exploration, and sensitivity to parameter settings. Further research is needed to explore strategies for improving the convergence speed and exploration capabilities of LSO-FFNN, as well as optimizing parameter configurations to enhance its overall effectiveness in various optimization and classification tasks.

“ABC-DA-ANN” [23] integrates two bio-inspired optimization algorithms into an Artificial Neural Network (ANN) framework. This hybrid approach harnesses the exploration and exploitation capabilities of both A.B.C. and DA algorithms to optimize the parameters of the neural network, enhancing its learning and generalization abilities. While ABC-DA-ANN shows promise in improving optimization and classification tasks, challenges such as parameter tuning complexities and potential algorithmic conflicts may arise. Further research is needed to explore strategies for effectively balancing the optimization processes of A.B.C. and DA within the ANN framework, optimizing parameter configurations, and enhancing the overall performance of ABC-DA-ANN across diverse applications.

2.1. Summary

The collection of studies presented encompasses a wide range of advancements in Intrusion Detection Systems (I.D.S.), each addressing specific challenges and contributing to improving network security. These studies introduce innovative approaches and techniques tailored to boost I.D.S.'s accuracy, robustness, and efficacy across various domains. From exploring machine learning algorithms for anomaly-based detection in IoT environments to simplifying the deployment of I.D.S. on cloud platforms, each study represents a significant step forward in developing and implementing effective security solutions. The advancements in feature selection, misbehavior classification, and attack detection further strengthen the overall security posture of networked systems. Despite the diverse approaches and methodologies, common goals of scalability, real-time threat detection, and adaptability underscore the importance of ongoing research in intrusion detection.

2.2. Research Gap

A notable research gap identified in the literature is the limited exploration of integrating and optimizing Intrusion Detection Systems (I.D.S.) for emerging technologies and network architectures. While existing studies have made significant strides in improving the accuracy and efficiency of I.D.S., there remains a lack of comprehensive frameworks tailored to novel environments such as edge computing, 5G networks, and Internet of Things (IoT) ecosystems. These environments present unique challenges, including resource constraints, dynamic network topologies, and heterogeneous device capabilities, necessitating specialized I.D.S. solutions. Hence, the need for research is expected to design I.D.S. frameworks suited to these emerging technologies, which will be specially designed to overcome issues such as scalability, real-time threat detection, and adaptability to dynamic conditions in the network. Integrating advanced features such as deep learning and reinforcement learning in I.D.S. frameworks tailored explicitly to the environments cited above can significantly promote their resilience to evolving cyber threats.

3. Proficient Red Deer Optimization-Based Relevance Vector Machine (PRDO-RVM)

Proficient Red Deer Optimization-based Relevance Vector Machine (PRDO-RVM) presents a novel fusion of two powerful optimization techniques, promising advancements in machine learning. By integrating the Red Deer Optimization (RDO) algorithm with the Relevance Vector Machine (RVM), PRDO-RVM aims to enhance classification accuracy and model efficiency across various domains. RDO's inspiration from the herd behavior of red deer facilitates robust convergence towards optimal solutions, while RVM's sparse representation ensures computational efficiency and generalization capabilities. This synergistic approach offers a potent framework for addressing complex classification tasks, demonstrating potential in various applications such as pattern recognition, bioinformatics, and financial forecasting.

3.1. Initialization

Initialization is important in the PRDO-RVM algorithm since it initiates the optimization process by setting the initial point. This section discusses the importance of initialization in PRDO-RVM and the involved essential considerations. The first step in the PRDO-RVM is initializing the population of red deer individuals. Let the population size be N and the dimensionality of the problem space D. Each red deer individual is defined as a solution vector X_i of dimension D, where $i = 1,2, \dots, N$. The population initialization process can be mathematically expressed as Equation (1).

$$X_i^{(0)} \in [lb, ub]^D, \;\; i = 1,2, \dots, N \quad (1)$$

Where $X_i^{(0)}$ represents the initial solution vector for the i^{th}red deer individual, and $[lb, ub]$ denotes the lower and higher bounds of the exploration space.

In addition to initializing the population, PRDO-RVM requires the initialization of hyperparameters for the RVM model. These hyperparameters include the kernel function parameters and the regularization parameter. Let $K(\cdot,\cdot)$ denote the chosen kernel function, and α represent the regularization parameter. The initialization of these hyperparameters can be described as Equation (2).

$$K_{init} = K\left(X_i^{(\theta)}, X_j^{(\theta)}\right), \;\; \alpha_{init} = \frac{1}{D}, \;\; i, j = 1,2, \dots., N \quad (2)$$

Where K_{init} denotes the initial kernel matrix based on the initialized solutions and α_{init} represents the initial value of the regularization parameter. Once the population and hyperparameters are initialized, the next step is to evaluate the fitness of each red deer individual. In the context of PRDO-RVM, fitness evaluation involves training the RVM model with the initialized hyperparameters and measuring its performance on a validation dataset. Let y represent the target vector, X denote the feature matrix, and w represent the weight vector of the RVM model. The fitness evaluation process can be formulated as Equation (3).

$$minimize \;\; J(w) = \frac{1}{2}\|y - Xw\|^2 + \frac{\lambda}{2} w^T K_{init} w \quad (3)$$

Where λ represents the regularization parameter, K_{init} is the initial kernel matrix and $||\cdot||$ denotes the Euclidean norm. Solving this optimization problem yields the weight vector w for each red deer individual, which is used to evaluate its fitness. After fitness evaluation, the best red deer individual (leader) is selected based on its fitness value. Let $f(X_i)$ denote the fitness value of the i^{th} red deer individual. The leader selection process can be mathematically expressed as Equation (4).

$$Leader = arg\min_i f(X_i) \quad (4)$$

Where $Leader$ represents the selected leader based on the minimum fitness value among all red deer individuals. In PRDO-RVM, it is essential to handle boundary constraints to ensure that the initialized solutions remain within the feasible region of the search space. This is particularly important to prevent solutions from wandering into infeasible regions, which could degrade the optimization process. Let lb_d and ub_d denote the lower and upper bounds for the d^{th}dimension of the search space, respectively. The boundary-handling process can be formulated as Equation (5).

$$X_{i,d}^{(0)} = \begin{cases} lb_d & if \; x_{i,d}^{(0)} < lb_d \\ ub_d & if \; x_{i,d}^{(0)} < lb_d \\ x_{i,d}^{(0)} & otherwise \end{cases} \quad (5)$$

Where $x_{i,d}^{(0)}$ represents the d^{th}component of the i^{th}red deer individual's solution vector. This ensures that each initialized solution remains within the specified bounds of the search space.

Algorithm 1. Initialization
Input: • Population size N • The dimensionality of the problem space D • Lower bound lb and upper bound ub of the search space **Output:** • The initial population of red deer individuals $X^{(0)} = \left[X_1^{(0)}, X_2^{(0)}, \dots \dots, X_N^{(0)}\right]$ • Initial kernel matrix K_{init} • Initial regularization parameter α_{init} **Procedure:** 1. Initialize the population: • For $i = 1$ to N: • Generate a random solution vector $X_i^{(0)}$of dimension D within the bounds $[lb, ub]^D$. $X_i^{(0)} \in [lb, ub]^D, \quad i = 1,2, \dots, N$ • Initialize the hyperparameters: • Set the initial regularization parameter K_{init} based on the initialized solutions: $K_{init} = K\left(X_i^{(0)}, X_j^{(0)}\right), \quad i, j = 1,2, \dots \dots, N$ 2. Return $X^{(0)}, K_{init}$, and α_{init} as the output of the initialization process.

3.2. Objective Function Evaluation

In PRDO-RVM, objective function evaluation is a critical step that determines the capability of each red deer. The objective function in PRDO-RVM aims to minimize the error between the predicted outputs of the RVM and the actual target values while penalizing complexity through regularization. Let y represent the target vector, X denote the feature matrix, w represent the weight vector of the RVM model, and λ denote the regularization parameter. The objective function can be mathematically formulated as Equation (6).

$$J(w) = \frac{1}{2}\|y - Xw\|^2 + \frac{\lambda}{2} w^T K w \quad (6)$$

Where $\|\cdot\|$ denotes the Euclidean norm, and K is the kernel matrix. It consists of two terms: the data fidelity term, measuring the difference between predicted and actual outputs, and a regularization term, which penalizes model complexity. Objective function evaluation involves training the RVM model with the initialized hyperparameters and measuring its performance on a validation dataset. The training aims to find the optimal weight vector w that minimizes the objective function $J(w)$. This can be achieved through techniques such as gradient descent or Bayesian inference. Once the RVM model is trained, the fitness of each red deer individual is calculated based on its ability to minimize the objective function. Let $f(X_i)$ denote the fitness value of the i^{th}red deer individual. This fitness calculation can be expressed as Equation (7).

$$f(X_i) = J(w_i) \quad (7)$$

Where w_i represents the weight vector obtained by training the RVM model with the solution vector $X_i$corresponding to the i^{th} red deer individual. After fitness evaluation, the best red deer individual (leader) is selected based on its fitness value. The leader selection process aims to identify the solution vector that yields the minimum objective function value. Mathematically, the leader selection can be formulated as Equation (8).

$$Leader = \arg\min_i f(X_i) \quad (8)$$

Where $Leader$ represents the selected leader based on the minimum fitness value among all red deer individuals. In PRDO-RVM, this evaluation of the objective function is indispensable for directing the optimization procedure towards solutions that minimize prediction error and model complexity. Through training the RVM model with initialized hyperparameters and its evaluation in the validation dataset, PRDO-RVM ensures that each red deer contributes effectively to the overall optimization process.

Algorithm 2. Objective function evaluation
Input: • Target vector y • Feature matrix X • Kernel matrix K • Regularization parameter λ • Weight vector $w_i$for each red deer individual X_i **Output:** • Fitness value $f(X_i)$ for each red deer individual X_i **Procedure:** **1. Training the RVM Model:** • Train the RVM model using the provided feature matrix X and target vector y, with the initialized hyperparameters (kernel matrix K and regularization parameter λ). • Obtain the weight vector w_i for each red deer individual X_i by solving the RVM optimization problem. **2. Fitness Calculation:** • For each red deer, an individual X_i calculate the fitness value $f(X_i)$ based on the trained RVM model. • Use the obtained weight vector $w_i$to compute the objective function value $J(w_i)$ for each individual. **3. Leader Selection:**

- Select the leader red deer individual based on the minimum objective function value among all red deer individuals.
- Identify the red deer individual X_{Leader} corresponding to the selected leader.

The fitness values obtained through objective function evaluation drive the leader's selection and influence the optimization algorithm's subsequent steps, ultimately discovering high-quality solutions for the regression or classification task.

3.3. Leader Selection

In PRDO-RVM, leader selection is pivotal in determining the guiding individual for the optimization process. It is imperative to underscore the significance of fitness evaluation. Each red deer individual's fitness is calculated based on its ability to minimize the objective function, as determined through the training of the RVM model. Mathematically, the fitness value $f(X_i)$ of the i^{th} red deer individual X_i can be mathematically represented as Equation (9).

$$f(X_i) = J(w_i) \quad (9)$$

Where $J(w_i)$ represents the objective function value attained by the RVM model trained with the solution vector X_i. The leader selection process aims to identify the red deer individual with the minimum fitness value, signifying its superior performance in minimizing the objective function. Mathematically, the leader X_{Leader} is selected as Equation (10).

$$X_{Leader} = \arg\min_i f(X_i) \quad (10)$$

Where X_{Leader} denotes the selected leader, and $argmin$ signifies the argument that minimizes the function. Once the leader X_{Leader} is determined and serves as the guiding individual for the subsequent optimization iterations. The leader influences the movement of other red deer individuals in the population, directing the search toward promising regions of the solution space. The leader's solution vector may undergo refinement or adaptation based on the optimization process's progress.

The leader's pivotal role in PRDO-RVM cannot be overstated. It serves as a reference point for other individuals and embodies the collective intelligence of the population, encapsulating the most promising solution discovered thus far.

The dynamic leader adjusts according to the changing optimisation landscape and guides the population toward fitness regions, thus inducing efficient exploration and exploitation of the solution space.

In PRDO-RVM, leader selection is intertwined with convergence criteria, which dictate when the optimization process terminates. Convergence is usually reached when the fitness values of consecutive leaders do not improve minimally or when a fixed number of iterations are achieved. The final leader will correspond to the best solution achieved by the algorithm, which gives insight into the optimal configuration of the RVM model for the problem at issue.

Algorithm 3: Leader Selection

Input:
- Fitness values $f(X_i)$ for each red deer individual X_i.

Output:
- Selected leader red deer individual X_{Leader}

Procedure:

1. Leader Selection:
- Identify the red deer individual X_i with the minimum fitness value among all individuals.
- Set the red deer individual with the minimum fitness value as the leader.

3.4. Herding Behavior

In PRDO-RVM, herding behaviour is a crucial aspect inspired by the collective movement tendencies observed in red deer populations. Herding behaviour in PRDO-RVM simulates the collective movement of red deer individuals towards regions with higher fitness values in the solution space. The leader, a red deer individual, plays a central role in guiding the herding behaviour of the population. Other individuals are influenced by the leader's solution vector, adjusting their positions towards regions the leader indicates as promising. Mathematically, the adjustment of individual positions is governed by the leader's solution vector and the diversity within the population.

$$X_i^{(t+1)} = X_i^{(t)} + \Delta X_i^{(t)} \quad (11)$$

Where $X_i^{(t)}$ represents the solution vector of the i^{th}red deer individual at iteration t, and $\Delta X_i^{(t)}$ denotes the adjustment vector influenced by the leader's solution and the diversity within the population. Herding behavior manifests a balance between exploration and exploitation of PRDO-RVM. Individual movement towards promising areas leads to exploring varied regions of the solution space while exploiting those areas with higher fitness to improve the quality of solutions. This leads to a dynamic exploration-exploitation trade-off, enabling more efficient and effective optimization processes.

$$\Delta X_i^{(t)} = \alpha.\text{Explore} + \beta.\text{Exploit} \quad (12)$$

Where α and β represent the exploration and exploitation coefficients, respectively. The adjustment vector $\Delta X_i^{(t)}$is a combination of exploration and exploitation components, guiding the movement of individuals towards promising regions. Herding behavior in PRDO-RVM is adaptive to fluctuations in the environment and landscapes of the problems. Individuals react to changes in movement strategy based on problem characteristics and leader dynamics. This adaptability will ensure the optimization process stays robust and responsive to conditions under varying problem scenarios.

$$Explore = \Delta X_{Leader}^{(t)} - X_i^{(t)} \quad (13)$$

$$Exploit = \frac{1}{N}\sum_{j=1}^{N}\left(X_j^{(t)} - X_i^{(t)}\right) \quad (14)$$

Algorithm 4: Herding Behavior

Input:
- Current positions $X^{(t)}$ of all red deer individuals
- Leader's position $X_{Leader}^{(t)}$
- Population size N
- Exploration coefficient α and exploitation coefficient β

Output:
- Adjustment vectors $\Delta X_i^{(t)}$for all red deer individuals

Procedure:

1. Calculate Exploration Component:
- For each red deer individual $X_i^{(t)}$:
- Compute the exploration component as $Explore = X_{Leader}^{(t)} - X_i^{(t)}$

2. Calculate Exploitation Component:
- For each red deer individual $X_i^{(t)}$:
- Compute the average displacement of all individuals relative to the i^{th}individual as:

$$AvgDisplacement = \frac{1}{N}\sum_{j=1}^{N}\left(X_j^{(t)} - X_i^{(t)}\right)$$

3. Adjustment Vector Calculation:
- For each red deer individual $X_i^{(t)}$:
- Compute the adjustment vector $X_i^{(t)}$ as a combination of exploration and exploitation components:

$$\Delta X_i^{(t)} = \alpha.Explore + \beta.Avgdisplacement.$$

Where $X_{Leader}^{(t)}$ represents the solution vector of the leader at iteration t, N is the population size, and $\sum_{j=1}^{N}\left(X_j^{(t)} - X_i^{(t)}\right)$ computes the average displacement of all individuals relative to the i^{th} individual.

3.5. Dynamic Parameter Adaptation

Dynamic parameter adaptation is a crucial aspect of the PRDO-RVM algorithm, enabling the adjustment of control parameters during optimization. PRDO-RVM incorporates adaptive control parameters that dynamically evolve throughout optimization. These parameters include exploration and exploitation coefficients, regularization parameters, and step sizes. By adapting these parameters based on the optimization progress and problem characteristics, PRDO-RVM ensures efficient exploration and exploitation of the solution space. Dynamic parameter adaptation enables PRDO-RVM to adjust the balance between exploration and exploitation as the optimization process unfolds. Mathematically, the exploration and exploitation coefficients (α and β) are adaptively updated based on the performance feedback obtained during optimization. This adaptive adjustment ensures that the algorithm effectively explores diverse regions while exploiting promising areas for optimization.

$$\alpha^{(t+1)} = f_\alpha\left(\alpha^{(t)}\right) \quad (15)$$

$$\beta^{(t+1)} = f_\beta\left(\beta^{(t)}\right) \quad (16)$$

Where $\alpha^{(t)}$ and $\beta^{(t)}$ represent the exploration and exploitation coefficients at iteration t, respectively, and f_α and f_β denote the adaptive update functions. In addition to exploration and exploitation coefficients, PRDO-RVM dynamically adapts the regularization parameter (λ) during optimization. This is why adapting the regularization parameter ensures the RVM model remains well-regularized throughout the optimization process, controlling the model complexity tradeoff against accuracy. The adaptive update of the regularization parameter can be expressed as Equation (17).

$$\lambda^{(t+1)} = f_\lambda\left(\lambda^{(t)}\right) \quad (17)$$

Where $\lambda^{(t)}$ represents the regularization parameter at iteration t, and f_λ denotes the adaptive update function. Dynamic parameter adaptation within PRDO-RVM also includes the adaptation of step sizes utilized in optimization algorithms like gradient descent and Bayesian inference. Step sizes are used as controls over the updated magnitude of parameters while also influencing the overall convergence behavior of the optimization process.

In that sense, PRDO-RVM improves the convergence speed and stability by adaptively adjusting the step sizes according to the optimization progress. The adaptive update of step sizes can be mathematically formulated as Equation (18).

$$StepSize^{(t+1)} = f_{StepSize}\left(StepSize^{(t)}\right) \quad (18)$$

Where $StepSize^{(t)}$ represents the step size at iteration t, and $f_{StepSize}$ denotes the adaptive update function.

Algorithm 5: Dynamic Parameter Adaptation

Input:
- Current values of control parameters (exploration and exploitation coefficients, regularization parameter, step sizes)
- Optimization progress indicators (e.g., fitness values, convergence criteria)

Output:
- Updated values of control parameters

Procedure:

1. Adaptive Exploration Coefficient (α):

- Determine an adaptive update function f_α based on the optimization progress and problem characteristics.
- Update the exploration coefficient as $\alpha^{(t+1)} = f_\alpha \alpha^{(t)}$.

2. Adaptive Exploitation Coefficient ($\boldsymbol{\beta}$):

- Determine an adaptive update function f_β based on the optimization progress and problem characteristics.
- Update the exploitation coefficient as $\beta^{(t+1)} = f_\beta\left(\beta^{(t)}\right)$.

3. Adaptive Regularization Parameter ($\boldsymbol{\lambda}$):

- Determine an adaptive update function f_λ based on the optimization progress and problem characteristics.
- Update the regularization parameter as $\lambda^{(t+1)} = f_\lambda(\lambda^t)$.

4. Adaptive Update of Step Sizes:

- Determine an adaptive update function $f_{StepSize}$ based on the optimization progress and problem characteristics.
- Update the step sizes used in optimization algorithms (e.g., gradient descent, Bayesian inference) as $StepSize^{(t+1)} = f_{StepSize}\left(StepSize^{(t)}\right)$.

3.6. Collaborative Learning

Collaborative learning in PRDO-RVM refers to the cooperative exchange of information among red deer individuals to enhance the optimization process. Collaborative learning in PRDO-RVM fosters information sharing among red deer individuals, allowing them to benefit from the population's collective knowledge and experiences. This information exchange occurs through communication, imitation, and adaptation, enabling individuals to learn from each other's successes and failures. The collaborative learning process in PRDO-RVM facilitates knowledge transfer between individuals, thereby accelerating the exploration and exploitation of the solution space. Mathematically, knowledge transfer can be represented as the exchange of solution vectors and associated fitness values among red deer individuals. This exchange of information enriches the population's collective understanding of the optimization problem, leading to more informed decision-making.

$$X_i^{(t+1)} = X_i^{(t)} + \Delta X_i^{(t)} \quad (19)$$

$$\Delta X_i^{(t)} = \sum_{j=1}^{N} w_{ij}.\left(X_j^{(t)} - X_i^{(t)}\right) \quad (20)$$

Where $X_i^{(t)}$represents the solution vector of the i^{th}red deer individual at iteration t,$\Delta X_i^{(t)}$ denotes the adjustment vector, N is the population size, $X_j^{(t)}$represents the solution vector of the j^{th}individual, and w_{ij} denotes the collaboration coefficient between individuals i and j. Collaboration coefficients in PRDO-RVM quantify the strength of interaction between red deer individuals during the collaborative learning process. These coefficients determine how individuals influence each other's movements and decisions. Collaboration coefficients can be mathematically defined based on spatial proximity, fitness similarity, and historical performance.

$$w_{ij} = \frac{1}{\sqrt{d_{ij}}} \quad (21)$$

Where in Equation (21) w_{ij}represents the collaboration coefficient between individuals i and j, and d_{ij} denotes the distance or similarity metric between their solution vectors.

The learning process in PRDO-RVM is cooperative and adapts to changes in the optimization landscape and problem characteristics. Every individual learns to dynamically adjust the coefficients of collaboration during optimization progress and environmental conditions. Such adaptation makes the collaborative learning process effective and efficient during optimization.

$$w_{ij}^{(t+1)} = f_w\left(w_{ij}^{(t)}\right) \quad (22)$$

Where $w_{ij}^{(t)}$represents the collaboration coefficient between individuals i and j at iteration f_w denotes the adaptive update function.

Algorithm 6: Collaborative Learning

Input:

- Current positions $X^{(t)}$of all red deer individuals
- Collaboration coefficients $w_{ij}^{(t)}$for all pairs of red deer individuals
- Population size N
- Distance or similarity metric d_{ij}between solution vectors

Output:

- Updated positions $X^{(t+1)}$of all red deer individuals after collaborative learning

Procedure:

1. Collaborative Learning:

- For each red deer individual $X_i^{(t)}$:
- Compute the adjustment vector $\Delta X_i^{(t)}$ based on collaborative information from other individuals.
- Update the position of $X_i^{(t)}$ as $X_i^{(t+1)} = X_i^{(t)} + \Delta X_i^{(t)}$

2. Adjustment Vector Calculation:

- For each red deer individual $X_i^{(t)}$:
- Compute the adjustment vector $\Delta X_i^{(t)}$ by summing the collaborative contributions from other individuals: $\Delta X_i^{(t)} = \sum_{j=1}^{N} w_{ij}^{(t)}.\left(X_j^{(t)} - X_i^{(t)}\right)$.

3. Collaboration Coefficient Update:

- For each pair of red deer, individuals X_i and X_j:

- Update the collaboration coefficient $w_{ij}^{(t+1)}$ based on the optimization progress and environmental conditions.

3.7. Environmental Adaptation

Environmental adaptation in PRDO-RVM refers to the mechanism by which the algorithm dynamically adjusts its behaviour and parameters in response to changes in the optimization landscape. Environmental adaptation in PRDO-RVM begins with sensing and monitoring the optimization environment. Red deer individuals continuously gather information about the optimization landscape, including changes in fitness values, gradients, and convergence patterns. The environmental feedback mechanism in PRDO-RVM enables the algorithm to interpret and respond to the sensory input obtained from the optimization environment. This feedback mechanism involves analyzing the collected information and identifying patterns or trends that may influence the optimization process. The environmental feedback mechanism can be mathematically represented as Equation (23).

$$Feedback = f_{Env}(SensoryInput) \quad (23)$$

Where $Feedback$ denotes the interpreted feedback from the environment, $SensoryInput$ represents the sensory input obtained by red deer individuals, and f_{Env} denotes the environmental feedback function. Environmental adaptation in the PRDO-RVM relies on control parameters and strategy adjustments inferred from the interpreted ecological feedback. This type of adaptive adjustment guarantees that the algorithm is sensitive to changes within the optimisation landscape and progresses with optimal solutions in view. Mathematically, adaptive parameter adjustment can be expressed as Equation (24).

$$Parameter^{(t+1)} = f_{Adapt}\left(Parameter^{(t)}, Feedback\right) \quad (24)$$

Where $Parameter^{(t)}$ represents the control parameter at iteration t, $Parameter^{(t+1)}$ denotes the updated control parameter, $Feedback$ signifies the interpreted environmental feedback, and f_{Adapt} denotes the adaptive adjustment function. Environmental adaptation in PRDO-RVM improves the robustness and flexibility of the algorithm in addressing various optimization scenarios and problem characteristics. PRDO-RVM continuously monitors the optimization environment and changes its behavior appropriately, making it sensitive to the shift in landscape, the level of noise, and the problem complexity, thereby ensuring that it consistently performs well in different problem domains.

$$Parameter^{(t+1)} = Parameter^{(t)} + \Delta Parameter^{(t)}$$

$$\Delta Parameter^{(t)} = \alpha . Feedback \quad (25)$$

Where $Parameter^{(t)}$ represents the control parameter at iteration t, $\Delta Paraneter^{(t)}$ denotes the adjustment to the control parameter, α represents the adaptation rate, and $Feedback$ signifies the interpreted environmental feedback.

Algorithm 7: Environmental Adaptation

Input:
- Sensory input obtained from the optimization environment
- Current values of control parameters
- Environmental feedback function f_{Env}
- Adaptive adjustment function f_{Adapt}
- Adaptation rate α

Output:
- Updated values of control parameters after environmental adaptation

Procedure:

1. Sensing the Environment:
- Red deer individuals continuously gather sensory input from the optimization environment, including changes in fitness values, gradients, and convergence patterns.

2. Environmental Feedback Interpretation:
- Interpret the sensory input obtained from the environment using the environmental feedback function $f_{Env.}$
- Obtain the interpreted environmental feedback, denoted as Feedback.

3. Adaptive Parameter Adjustment:
- For each control parameter $Parameter^{(t)}$at iteration t:
- Update the control parameter adaptively based on the interpreted environmental feedback using the adaptive adjustment function f_{Adapt}:

$$Parameter^{(t+1)} + f_{Adapt}\left(Parameter^{(t)}, Feedback\right)$$

3.8. Enhanced Exploration

PRDO-RVM strategic enhanced exploration would provide the best opportunity to improve on the algorithm's search performance relating to the diversity of regions of solution space. Exploration is an important element of any optimization algorithm, like PRDO-RVM, because it allows for generating new and potentially optimal solutions. A good exploration strategy should ensure a broad search among solution candidates, which avoids early convergence to a suboptimal solution and increases the algorithm's robustness. Mathematically, the exploration-exploitation trade-off can be represented as Equation (26).

$$Adjustment = \alpha . Explore + \beta . Exploit \quad (26)$$

Where α and β denote the exploration and exploitation coefficients, respectively, and Explore and Exploit represent the exploration and exploitation components. Enhanced exploration strategies incorporate adaptive mechanisms to dynamically adjust the exploration behaviour based on the optimization progress and problem characteristics. These adaptive strategies enable the algorithm to focus exploration efforts on regions with high uncertainty or low fitness values, thereby improving the efficiency of the exploration process.

$$Explore = X_{Leader} - X_i \quad (27)$$

$$Exploit = \frac{1}{N}\sum_{j=1}^{N}(X_j - X_i) \quad (28)$$

Where X_{Leader} represents the leader's position, X_i represents the current position of the individual, N denotes the population size, and the summation computes the average displacement of all individuals relative to the i^{th}individual. Enhanced exploration strategies in PRDO-RVM incorporate diversity-promoting mechanisms to encourage the exploration of diverse regions of the solution space. These mechanisms may include introducing randomness in the search process, promoting genetic diversity within the population, or employing multi-start approaches to initiate multiple search trajectories from different starting points.

$$\Delta X_i = \sum_{j=1}^{N} w_{ij} \,.\, (X_j - X_i) \quad (29)$$

$$w_{ij} = \frac{1}{\sqrt{d_{ij}}} \quad (30)$$

Where ΔX_i represents the adjustment vector for the i^{th}individual, w_{ij}denotes the collaboration coefficient between individuals i and j, and d_{ij} represents the distance or similarity metric between their solution vectors.

Algorithm 8: Enhanced Exploration

Input:
- Current positions of red deer individuals $X^{(t)}$
- Fitness values corresponding to each position
- Local search radius r

Output:
- Refined positions of red deer individuals after local search

Procedure:

1. Initialize Refined Positions:
- Initialize an empty list to store the refined positions of red deer individuals.

2. Local Search for Each Individual:
- For each red deer $X_i^{(t)}$:
- Perform a local search within a radius r around the current position $X_i^{(t)}$.
- Evaluate the fitness of each candidate solution within the local search radius.

3. Update Refined Positions:
- Select the candidate solution with the highest fitness within the local search radius for each red deer individual.
- Update the refined position of the red deer individual with the selected candidate solution.

3.9. Adaptive Memory Mechanism

The Adaptive Memory Mechanism in PRDO-RVM is a vital component that enables the algorithm to retain and utilize valuable information obtained during optimization. The Adaptive Memory Mechanism in PRDO-RVM focuses on retaining valuable information gathered throughout the optimization process. This information may include promising solution candidates, historical performance data, and past optimization trajectories. Retaining such information allows the algorithm to leverage past experiences to guide future exploration and exploitation efforts effectively. In PRDO-RVM, the Adaptive Memory Mechanism integrates retained information from various sources using weighted aggregation techniques. The weighted information integration can be mathematically represented as Equation (31).

$$Memory^{(t+1)} = \sum_{i=1}^{N} w_i \,.\, Information_i^{(t)} \quad (31)$$

Where $Memory^{(t+1)}$ denotes the updated adaptive memory at iteration $t+1$, $Information_i^{(t)}$ represents the information retained from source i at iteration t, and w_i denotes the weight assigned to each source of information. The Adaptive Memory Mechanism dynamically adjusts the weights assigned to different sources of information based on their relevance and usefulness in the optimization process. This adaptive weight adjustment ensures that more weight is transferred to reliable and informative sources while less weight is allocated to less relevant or outdated information. The adaptive weight adjustment can be mathematically expressed as Equation (32).

$$w_i^{(t+1)} = f_{Weight}\left(Information_i^{(t)}, Feedback\right) \quad (32)$$

Where $w_i^{(t+1)}$ represents the updated weight assigned to source i at iteration $t+1$, $Feedback$ denotes the environmental feedback obtained from the optimization landscape, and f_{Weight}denotes the adaptive weight adjustment function. The Adaptive Memory Mechanism employs a memory-updating strategy to manage the retention and utilization of information over time.

This strategy may involve forgetting outdated or irrelevant information, updating memory based on recent performance feedback, and prioritizing retaining high-quality solution candidates. By continuously updating the adaptive memory, PRDO-RVM ensures that the retained information remains relevant and effective in optimizing the process.

$$Memory^{(t+1)} = Memory^{(t)} + \Delta Memory^{(t)} \quad (33)$$

$$\Delta Memory^{(t)} = \alpha . Feedback \quad (34)$$

Where $Memory^{(t)}$ represents the adaptive memory at iteration t, $\Delta Memory^{(t)}$ denotes the adjustment to the memory, α represents the adaptation rate, and $Feedback$ signifies the interpreted environmental feedback.

Algorithm 9: Adaptive Memory Mechanism
Input: • Current adaptive memory $Memory^{(t)}$ • Information retained from various sources $Information_i^{(t)}$ for $i = 1,2, \dots, N$ • Environmental feedback $Feedback$ **Output:** • Updated adaptive memory $Memory^{(t+1)}$ **Procedure:** **1. Weighted Information Integration:** • To store the weighted information, initialize the empty list. • For each source i: • Compute the weighted information as $w_i . Information_i^{(t)}$. • Store the weighted information in the list. **2. Adaptive Weight Adjustment:** • For each source i: • Update the weight $w_i^{(t+1)}$ based on the relevance and usefulness of the information and the environmental feedback $Feedback$. **3. Memory Updating:** • Combine the weighted information from all sources using the updated weights to obtain the updated adaptive memory. • Update the adaptive memory as $Memory^{(t+1)} = \sum_{i=1}^{N} w_i^{(t+1)} . Information_i^{(t)}$.

3.10. Dynamic Population Management

Dynamic Population Management in PRDO-RVM refers to the adaptive management of the population size and composition throughout the optimization process. Dynamic Population Management involves adapting the population size based on the optimization progress and problem characteristics. This adaptation ensures that the algorithm allocates resources efficiently, avoiding unnecessary computational overhead while maintaining exploration and exploitation capabilities. Mathematically, population size adaptation can be represented as Equation (35).

$$N^{(t+1)} = f_{Adapt}\left(N^{(t)}, Feedback\right) \quad (35)$$

Where $N^{(t)}$ represents the population size at iteration t, $N^{(t+1)}$ denotes the updated population size at iteration $t+1$, Feedbackack signifies the environmental feedback obtained from the optimization landscape, and f_{Adapt} denotes the adaptation function.

Dynamic population management relates to this area, as diversity in the population should be maintained to avoid premature convergence and encourage the search for different parts of the solution space. Ensuring this diversity means introducing new individuals, removing redundant or low-quality individuals, and encouraging genetic diversity through crossover and mutation operations.

$$Diversity = \frac{1}{N}\sum_{i=1}^{N} Dist(X_i, X_{centroid}) \quad (36)$$

$$Dist(X_i, X_{centroid}) = \sqrt{\sum_{j=1}^{d}\left(X_{ij} - X_{centroid_j}\right)^2} \quad (37)$$

Where $Diversity$ represents the diversity measure within the population, N denotes the population size, X_i represents the position of the i^{th} individual, $X_{centroid}$ denotes the centroid of the population, d represents the dimensionality of the solution space, and X_{ij} represents the j^{th} component of the position vector X_i. Dynamic Population Management incorporates adaptive selection mechanisms to determine which individuals are retained, removed, or added to the population at each iteration. These selection mechanisms consider individual fitness, diversity, and contribution to the population's overall performance. By adapting the selection criteria based on the optimization progress, PRDO-RVM ensures that the population evolves effectively towards optimal solutions.

$$Selection\_Probability_i = \frac{f_i}{\sum_{j=1}^{N} f_j} \quad (38)$$

$$Survival_{Probability_i} = \frac{1}{1+e^{-\alpha . f_i}} \quad (39)$$

Where $Selection\_Probability_i$ represents the probability of selecting individual i, f_i denotes the fitness value of individual i, α represents the adaptation rate, and $Survival\_Probability_i$ denotes the probability of individual i surviving the selection process. Dynamic Population Management also includes adaptive evolution strategies to guide the evolution of the population toward regions of the solution space that exhibit promising characteristics. These strategies may involve elitism, where the best-performing individuals are retained in the population, and adaptive mutation rates, which are adjusted based on the optimization progress and problem complexity.

$$Mutation\_Rate = \frac{1}{1+\sum e^{-\alpha . Feedback}} \quad (40)$$

$$Crossover\_Probability = \frac{1}{1+\sum e^{\alpha . Feedback}} \quad (41)$$

Where $Mutation\_Rate$ represents the probability of mutation, $Crossover\_Probability$ denotes the probability of crossover, and $Feedback$ signifies the environmental feedback obtained from the optimization landscape. Dynamic Population Management in PRDO-RVM ensures efficient allocation of resources, promotes exploration and exploitation, and maintains diversity within the population, ultimately leading to the discovery of high-quality solutions for regression or classification tasks. By incorporating adaptive population size adaptation, diversity maintenance strategies, adaptive selection mechanisms, and adaptive evolution strategies, PRDO-RVM enhances its ability to navigate the optimization landscape and achieve superior performance effectively.

Algorithm 10: Dynamic Population Management

Input:

- Current population $P^{(t)}$of red deer individuals
- Fitness values corresponding to each individual
- Environmental feedback $Feedback$
- Control parameters for adaptive selection and evolution strategies

Output:

- Updated population $P^{(t+1)}$ after dynamic population management

Procedure:

1. Population Size Adaptation:

- Adjust the population size based on the optimization progress and environmental feedback:
$$N^{(t+1)} = f_{Adapt}\left(N^{(t)}, Feedback\right).$$

2. Diversity Maintenance:

- Evaluate the diversity within the population using a diversity measure such as the average distance to the centroid.
- Ensure that the diversity measure meets a predefined threshold by adding or removing individuals as necessary to maintain diversity.

3. Adaptive Selection Mechanisms:

- Calculate the selection probability for each individual based on its fitness value: $Selection\_Probability_i = \frac{f_i}{\sum_{j=1}^{N} f_j}$
- Determine the survival probability for each individual based on its fitness value and control parameters: $Survival\_Probability_i = \frac{1}{1+e^{-\alpha.f_i.}}$

4. Selection and Reproduction:

- Select individuals based on their selection probabilities and survival probabilities.
- Apply reproduction operators such as crossover and mutation to generate offspring from selected individuals.

5. Adaptive Evolution Strategies:

- Adjust mutation rates and crossover probabilities based on the optimization progress and environmental feedback: $Mutation_{Rate} = \frac{1}{1+\sum e^{-\alpha.Feedback}}$ and $Crossover\_Probability = \frac{1}{1+\sum e^{\alpha.Feedback}}$.

6. Population Update:

- Combine the offspring generated from reproduction with the current population.
- Select the best-performing individuals to retain in the population using elitism or other selection criteria.
- Update the population with the selected individuals to obtain $P^{(t+1)}$.

3.11. Multi-Objective Optimization Support

Multi-Objective Optimization Support PRDO-RVM: Extend the algorithm to problems with multiple conflicting objectives. In multi-objective optimization, one tries to optimize several conflicting objectives concurrently to obtain a family of trade-off solutions, known as the Pareto front. This can be represented mathematically as Equation (42).

$$Minimize\ f(X) = \left(f_1(X), f_2(X), \dots, f_m(X)\right) \quad (42)$$

Where $f(X)$ represents the vector of objective functions, $f_i(X)$ represents the $i^{th} the$ objective function and X denotes the decision variable vector. In multi-objective optimization, solutions are compared based on Pareto dominance, where one solution is considered superior to another if it is better in at least one objective and not worse in any other objective. Mathematically, solution X_1 dominates solution X_2 if Equations (43) and (44) are satisfied.

$$\forall i \in \{1,2, \dots\dots, m\}\text{:}\ f_i(X_1) \leq f_i(X_2) \quad (43)$$

$$\exists j \in \{1,2, \dots\dots, m\}\text{:}\ f_j(X_1) \leq f_j(X_2) \quad (44)$$

PRDO-RVM supports multi-objective optimization by integrating various strategies to handle the complexity of multi-objective optimization problems. These strategies include:

- **Pareto Front Exploration:** PRDO-RVM explores the Pareto front by simultaneously optimizing multiple objectives, allowing it to discover diverse trade-off solutions.

- **Pareto Ranking:** PRDO-RVM ranks solutions based on their Pareto dominance relationships, allowing it to identify non-dominated solutions that form the Pareto front.

- **Diversity Preservation:** PRDO-RVM maintains diversity within the population to ensure that a diverse set of Pareto-optimal solutions is explored and retained.

- **Convergence to Pareto Front:** PRDO-RVM aims to converge to the Pareto front by adapting its search behavior and population management strategies to focus on exploring and refining solutions along the Pareto front.

In PRDO-RVM, fitness assignment for multi-objective optimization involves evaluating the quality of solutions based on their position relative to the Pareto front. Non-dominated solutions are assigned a higher fitness value, indicating their superiority in the optimization landscape. Mathematically, fitness assignment can be represented as Equation (45).

$$Fitness(X) = Pareto\_Rank(X) \quad (45)$$

Where $Fitness(X)$ represents the fitness value of solution X, and $Pareto\_Rank(X)$ denotes the Pareto rank of solution X.

PRDO-RVM employs evolutionary operators such as selection, crossover, and mutation to guide the search process toward Pareto-optimal solutions. These operators are adapted to handle multiple objectives by considering Pareto dominance relationships and maintaining diversity within the population.

$$Updated\_Mutation\_Rate = \sum \frac{1}{1+\sum e^{-\alpha.Feedback}} \quad (46)$$

$$Updated\_Crossover\_Probability = \sum \frac{1}{1+\sum e^{\alpha.Feedback}} \quad (47)$$

Where $Mutation\_Rate$ represents the probability of mutation, $Crossover\_Probability$ denotes the probability of crossover, and $Feedback$ signifies the environmental feedback obtained from the optimization landscape.

Algorithm 11: Multi-Objective Optimization Support

Input:
- Current population $P^{(t)}$ of red deer individuals
- Fitness values corresponding to each individual for multiple objectives
- Environmental feedback $Feedback$
- Control parameters for adaptive evolution strategies

Output:
- Updated population $P^{(t+1)}$after multi-objective optimization support

Procedure:

1. Pareto Front Exploration:
- Explore the Pareto front by optimizing multiple objectives using the red deer optimization algorithm.
- Generate a diverse set of solutions that represent different trade-offs between the objectives.

2. Pareto Ranking:
- Rank solutions in the current population based on their Pareto dominance relationships.
- Identify non-dominated solutions that form the Pareto front.

3. Diversity Preservation:
- Maintain diversity within the population by applying selection, crossover, and mutation operators that promote genetic diversity.
- Ensure that a diverse set of Pareto-optimal solutions is explored and retained.

4. Convergence to Pareto Front:
- Adapt the search behaviour and population management strategies to focus on exploring and refining solutions along the Pareto front.
- Guide the algorithm towards convergence to the Pareto front by adjusting evolutionary operators and selection mechanisms.

5. Fitness Assignment:
- Evaluate the quality of solutions relative to their position on the Pareto front.
- Assign the fitness values to solutions based on their Pareto rank, where the non-dominated solutions are assigned the higher values.

6. Evolutionary Operators:
- To direct the search procedure toward Pareto-optimal solutions.
- Adapt these operators to handle multiple objectives by considering Pareto dominance relationships and maintaining diversity within the population.

7. Adaptive Evolution Strategies:
- Adjust mutation rates and crossover probabilities based on the optimization progress and environmental feedback:

3.12. Convergence Analysis

Convergence analysis in a PRDO-RVM is essential to evaluate an algorithm's performance and behavior concerning optimization processes, among many others. Convergence criteria specify whether the optimization process has converted to a satisfactory solution. Thus, the criteria for convergence for PRDO-RVM can be stated as a maximum number of iterations achieved, a certain fitness improvement level reached, or a predefined tolerance level regarding solution accuracy. Mathematically, convergence criteria can be expressed as Equation (48).

$$Stop_{Criterion} = \begin{cases} True, \; if \; convergence \; criteria \; met \\ False, \qquad\qquad\qquad\qquad otherwise \end{cases} \quad (48)$$

The convergence rate analysis studies how quickly the algorithm converges to an optimal solution as the number of iterations increases. Convergence rate can give insights into the efficiency of algorithms in finding high-quality solutions. Mathematically, the convergence rate can be quantified using metrics such as the rate of improvement in fitness values over iterations:

$$Convergence\_Rate = \frac{f^{(t)} - f^{(t-1)}}{f^{(t-1)}} \quad (49)$$

Where $f^{(t)}$represents the fitness value at iteration t, and $f^{(t-1)}$ denotes the fitness value at the previous iteration. Convergence analysis techniques used in PRDO-RVM monitor the various convergence indicators, including fitness trends, population diversity, and solution stability, across the optimization process. Convergence analysis techniques allow practitioners to evaluate the convergence behavior and identify potential. Convergence monitoring involves continuously tracking convergence indicators throughout the optimization process and taking corrective actions if convergence issues are detected. This iterative process allows practitioners to ensure that the algorithm progresses toward convergence effectively and efficiently. Mathematically, convergence monitoring can be expressed as Equation (50).

$$Monitor\left(P^{(t)}, f^{(t)}, Other_{Indicators}\right) \quad (50)$$

Where $P^{(t)}$represents the current population, $f^{(t)}$ denotes the fitness values at iteration t, and $Other\_Indicators$ includes additional convergence indicators such as population diversity or solution stability.

Algorithm 12: Convergence Analysis

Input:
- Current population $P^{(t)}$of red deer individuals
- Fitness values corresponding to each individual
- Convergence criteria (e.g., maximum number of iterations, tolerance for fitness improvement)
- Control parameters for convergence analysis

Output:
- Convergence status (True if convergence criteria met, False otherwise)
- Convergence rate
- Convergence analysis metrics (e.g., fitness trends, population diversity)

Procedure:

1. Initialize Convergence Analysis:
- Set initial iteration count $t = 0$.
- Initialize convergence status as False.

2. Convergence Monitoring Loop:
- While convergence criteria not met:
- Increment iteration count t.
- Perform optimization iterations using PRDO-RVM to update the population and fitness values.
- Check convergence criteria to determine if convergence has been achieved:
- If convergence criteria are met, set convergence status as True and break out of the loop.

3. Convergence Rate Calculation:
- Calculate the convergence rate based on the change in fitness values over iterations:

$$Convergence\_Rate = \frac{f^{(t)} - f^{(t-1)}}{f^{(t-1)}}$$

where $f^{(t)}$represents the fitness value at iteration t, and $f^{(t-1)}$denotes the fitness value at the previous iteration.

4. Convergence Analysis Metrics:
- Monitor convergence indicators such as fitness trends, population diversity, and solution stability throughout the optimization process.
- Collect relevant convergence analysis metrics for further analysis and visualization.

3.13. Result Analysis

Result analysis is also an important constituent of the PRDO-RVM that offers insights into how optimization occurs and aids in interpreting algorithm performance. Performance metrics evaluation involves the assessment of the quality of solutions that PRDO-RVM has produced using specific metrics measured according to the task.

They may include measures of predictive accuracy, model complexity, generalization ability, and computational efficiency. Mathematically, performance metrics evaluation can be represented as Equation (51).

$$Metric = f(Predicted, Actual) \quad (51)$$

Where $Metric$ represents the performance metric, $Predicted$ denotes the predicted values generated by the optimized model, and $Actual$ represents the ground truth or actual values. For multi-objective optimization problems, visualizing the Pareto front allows practitioners to understand the trade-offs between conflicting objectives and identify promising solutions along the front. Pareto front visualization techniques include scatter plots, radar charts, and parallel coordinate plots, which display the distribution of solutions in the objective space. Mathematically, the Pareto front can be visualized as Equations (52) and (53).

$$Objective_1 = f_1(X) \quad (52)$$

$$Objective_2 = f_2(X) \quad (53)$$

Where $Objective_1$ and $Objective_2$ represent the two objectives and $f_1(X)$ and $f_2(X)$ denote the corresponding objective function values. Model interpretability analysis involves examining the learned model's coefficients, support vectors, feature importance scores, and decision boundaries to gain insights into the relationships between input variables and output predictions. Techniques such as coefficient plots, feature importance plots, and decision boundary visualization aid in understanding the model's behaviour and identifying influential features. Mathematically, model interpretability analysis can be expressed as Equation (54).

$$Interpretability\_Metric = g\left(Model_{Coefficients}, Suppor_{Vectors}, Feature_{Importance}\right) \quad (54)$$

Where $Interpretability\_Metric$ represents a measure of model interpretability, and g denotes the interpretability analysis function. Visualizing the convergence behaviour of PRDO-RVM allows practitioners to track the progress of the optimization process, identify convergence patterns, and detect convergence issues or inefficiencies. Convergence behaviour visualization techniques include convergence plots, convergence rate plots, and convergence trajectory plots, illustrating the evolution of fitness values over iterations. Mathematically, convergence behaviour visualization can be represented as Equation (55).

$$Fitness^{(t)} = h(t) \quad (55)$$

Where $Fitness^{(t)}$ represents the fitness value at iteration t, and $h(t)$ denotes the convergence behaviour visualization function.

Algorithm 13: Analysing Results

Input:
- Optimized model parameters or solutions obtained by PRDO-RVM
- Ground truth or actual values for evaluation
- Control parameters for visualization

Output:
- Visualization of optimization outcomes
- Performance metrics evaluation results
- Model interpretability analysis insights
- Convergence behavior visualization

Procedure:

1. Performance Metrics Evaluation:
- Calculate relevant performance metrics (e.g., accuracy, precision, recall) by comparing predicted and actual values.
- Store the performance metric values for further analysis and visualization.

2. Multi-Objective Optimization:
- Visualize the Pareto front using scatter plots or radar charts if the optimization task involves multiple objectives.
- Plot solutions distribution in the objective space to identify trade-offs between conflicting objectives.

3. Model Interpretability Analysis:
- Analyze the learned model's coefficients, support vectors, feature importance scores, or decision boundaries.
- Generate plots (e.g., coefficient plots, feature importance plots, decision boundary visualization) to interpret the model's behaviour and identify influential features.

4. Convergence Behavior Visualization:
- Visualize the convergence behaviour of PRDO-RVM by plotting fitness values over iterations.
- Generate convergence plots, convergence rate plots, or convergence trajectory plots to track the optimization progress and identify convergence patterns or issues.

5. Output Visualization:
- Display the visualization of optimization outcomes, including Pareto front visualization (if applicable), model interpretability analysis plots, and convergence behaviour plots.
- Present performance metrics evaluation results and insights gained from model interpretability analysis in a clear and accessible format.

6. Result Analysis and Interpretation:
- Analyze the visualization outputs and performance metrics evaluation results to interpret the optimization outcomes effectively.
- Identify strengths, weaknesses, and areas for improvement in the optimized solutions or models.
- Communicate the analysis findings and insights to stakeholders concisely and understandably.

By following this algorithm, practitioners can effectively analyze and visualize the results obtained from PRDO-RVM, gain insights into the optimization process and outcomes, and make informed decisions to address the optimization task's objectives.

4. Dataset Information

The NSK-KDD dataset remains the foundation of network security, as it contains an all-inclusive number of 5,209,458 network traffic records. These records embrace a wide spectrum of network behaviors, including benign and malicious activities. However, network traffic data also suffers duplication. This may result in certain inaccuracies reflected during the analysis and model training process. Therefore, the prime focus in the NSK-KDD dataset is centered on the 1,152,281 unique records without redundancy. These unique records are distinct network activity instances that maintain data integrity and allow for building a more reliable I.D.S. The unique records are very important for fine-tuning machine learning models to detect security threats accurately, making intrusion detection and security measures more precise and reliable. This dedicated focus on uniqueness within the NSK-KDD dataset guarantees that network security professionals can construct more resilient and efficient security solutions, thereby fortifying network environments against potential attacks and vulnerabilities.

Table 1. Dataset's feature information

Attribute	*Description*
Duration	*Records connection time for security insights*
Failed Logins	*Counts unsuccessful login attempts*
Flag	*Indicates connection status*
Inbound /Outbound Connections	*Quantifies network flow*
IP Addresses	*Reveals source and destination hosts*
Label (Class)	*Categorizes connections as typical or attack types for precise intrusion detection*
Port Numbers	*Identifies application ports*
Protocol Type	*Categorizes communication protocols*
Service	*Specifies network applications*
Successful Logins	*Tracks legitimate user access*

5. Performance Metrics

Measure the efficiency, effectiveness, and quality of a system, process, or algorithm using quantitative indicators, enabling evaluation, comparison, and optimization across specific objectives and criteria.

5.1. Precision

It calculates the percentage of true positive predictions out of the total positive predictions, helping to understand how well a model identifies positive cases.

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives} \tag{56}$$

5.2. Recall

It computes the number of relevant items retrieved against the total number of applicable items available in a dataset to quantify the degree to which the system captures all relevant content.

$$Recall = \frac{No.of\ Relevant\ items\ retrieved}{Total\ number\ of\ relevant\ items} \tag{57}$$

5.3. Classification Accuracy

CA indicates the number of accurate classifications the model achieves out of the number of classifications attempted.

$$Classification\ Accuracy = \frac{No.of\ accurately\ detected\ intrusions}{Total\ number\ of\ intrusions} \tag{58}$$

5.4. F-Measure

The F-measure is a critical metric for optimizing intrusion detection performance in I.D.S. It helps security professionals fine-tune I.D.S. parameters to meet specific security and operational requirements by finding the optimal compromise between precision and recall.

$$F - Measure = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{59}$$

5.5. Matthews Correlation Coefficient

MCC is a fair measure in I.D.S., considering both true positives and negatives, with false positives and false negatives taken care of. It gives a more accurate assessment of I.D.S.'s ability to classify intrusions and non-intrusions correctly. Hence, this metric is used more than others to assess classification quality.

$$MCC = \frac{TP \times TN}{\sqrt{(TP+FP)(TP+FN))(TN+FP)(TN+FN)}} \times 100 \tag{60}$$

5.6. Fowlkes-Mallows Index

In I.D.S., the FMI can be used to assess the level of agreement or concordance between the clustering results generated by the system and the ground truth of intrusion categories. It helps understand how well the I.D.S. captures the underlying structure in intrusion data.

$$FMI = \frac{TP}{\sqrt{(TP+FP)(FP+FN)}} \times 100 \tag{61}$$

6. Results and Discussion

A section in research analyzes data outcomes, compares them with objectives, and interprets findings to highlight improvements, implications, and potential future enhancements.

6.1. Precision and Recall Analysis

Figure 1 provides a detailed Precision and Recall Analysis, essential for evaluating the efficacy of LSO-FFNN, ABC-DA-ANN, and PRDO-RVM classification algorithms. Precision and recall metrics offer nuanced insights into the algorithms' performance, particularly in scenarios with imbalanced class distributions or varying costs associated with misclassifications. The tabulated precision and recall values in Table 2 elucidate the algorithms' discriminative abilities. LSO-FFNN achieves 52.257% and 49.247%, respectively. Despite its potential, LSO-FFNN exhibits suboptimal performance, attributed to inherent limitations such as constrained exploration, sluggish convergence rates, and susceptibility to parameter sensitivity, which may hinder its precision and recall capabilities. In contrast, ABC-DA-ANN showcases improved precision and recall, recording 60.955% and 60.795%, respectively. This integration capitalizes on the synergies between optimization techniques, facilitating enhanced exploration and exploitation of the solution space, thereby elevating precision and recall metrics. The better performance is reached with PRDO-RVM, a fusion of Proficient Red Deer Optimization and Relevance Vector Machine, boasting precision and recall values of 86.864% and 88.389%, respectively.

PRDO-RVM's superiority stems from its proficient optimization capabilities, robustness to noisy data, adaptability across diverse problem domains, mitigation of overfitting risks, and inherent interpretability, culminating in exceptional precision and recall outcomes.

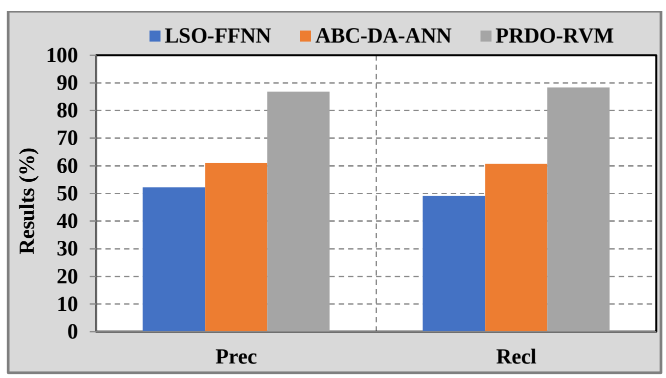

Fig. 1 Precision and Recall Analysis

Table 2. Result - Precision and Recall values

$Classification\ Algorithms$	$Precision$	$Recall$
$LSO - FFNN$	52.257	49.247
$ABC - DA - ANN$	60.955	60.795
$PRDO - RVM$	86.864	88.389

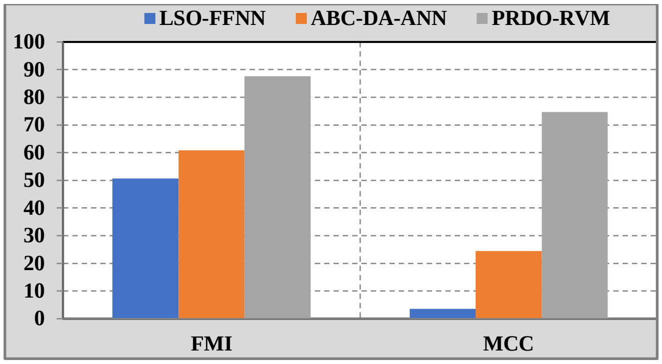

Fig. 2 FMI and MCC analysis

Figure 1 furnishes a comprehensive Precision and Recall Analysis, elucidating the intricate performance differentials among classification algorithms. By navigating the technical nuances of each algorithm's strengths and limitations, stakeholders can make informed decisions regarding algorithm selection tailored to specific application necessities and computational constraints.

6.2. Fowlkes-Mallows Index and Matthews Correlation Coefficient Analysis

Figure 2 comprehensively analyzes the FMI and MCC for LSO-FFNN, ABC-DA-ANN, and PRDO-RVM. These metrics serve as pivotal indicators of clustering quality and overall classification accuracy, essential for evaluating the efficacy of machine learning models across various domains. LSO-FFNN demonstrates moderate performance in terms of FMI. and MCC. While this approach benefits from the inherent parallelism and decentralized decision-making of swarm intelligence, it often struggles with limited exploration and slow convergence. The exploration limitation arises from the algorithm's reliance on local information exchange, potentially leading to premature convergence and suboptimal solutions. Additionally, the convergence rate of LSO-FFNN may be hindered by the complex, high-dimensional search spaces typical in neural network optimization tasks. The sensitivity of LSO-FFNN to parameter settings poses a challenge, requiring careful tuning to achieve optimal performance. ABC-DA-ANN presents improved clustering quality and classification accuracy compared to LSO-FFNN. This approach is not without its drawbacks. One significant challenge is the complexity associated with parameter tuning, as both the optimization algorithms and neural network architecture require careful configuration to achieve optimal performance. Moreover, integrating multiple optimization techniques may introduce algorithmic conflicts, where the exploration and exploitation strategies of A.B.C. and DA may not align seamlessly. This can result in inefficient search behavior and suboptimal convergence rates, particularly in high-dimensional search spaces. The computational costs of running multiple optimization algorithms concurrently can be substantial, potentially limiting the scalability of ABC-DA-ANN to large-scale datasets or resource-constrained environments.

Table 3. Result - FMI. and MCC

$Classification\ Algorithms$	FMI	MCC
$LSO - FFNN$	50.730	3.527
$ABC - DA - ANN$	60.875	24.416
$PRDO - RVM$	87.623	74.649

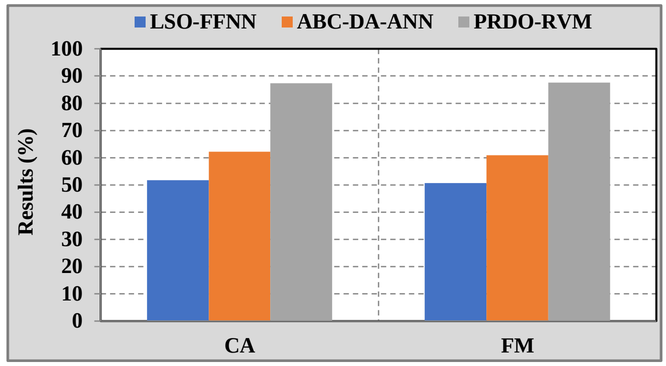

Fig. 3 Classification Accuracy and F-Measure analysis

PRDO-RVM is the top performer, showcasing exceptional clustering quality and classification accuracy. The proficiency of PRDO-RVM stems from its efficient optimization strategies and robust classification capabilities. The exploration vs exploitation trade-off is balanced within the Red Deer Optimization algorithm to explore the solution space thoroughly while efficiently converging to high-quality solutions. The Relevance Vector Machine offers properties of sparsity, which reduce overfitting risk and contribute to better model interpretability. PRDO-RVM's adaptability to diverse problem domains further solidifies its position as a compelling choice for clustering and classification tasks in real-world applications.

6.3. Classification Accuracy and F-Measure Analysis

Figure 3 presents an in-depth exploration of Classification Accuracy (CA) and F-Measure (FM), critical metrics for evaluating the performance of classification algorithms, namely LSO-FFNN, ABC-DA-ANN, and PRDO-RVM. These metrics offer nuanced insights into the algorithms' ability to accurately classify instances and strike a balance between precision and recall, crucial for various real-world applications. The tabulated CA and FM values in Table 4 provide a quantitative assessment of each algorithm's classification performance. LSO-FFNN demonstrates reasonable performance with CA and FM scores of 51.741 and 50.707, respectively. Despite its potential, LSO-FFNN may encounter challenges such as slow convergence and limited exploration stemming from the complex, high-dimensional search space inherent in neural network optimization tasks.

Table 4. Result - CA and FM

$Classification\ Algorithms$	CA	FM
$LSO - FFNN$	51.741	50.707
$ABC - DA - ANN$	62.255	60.875
$PRDO - RVM$	87.325	87.620

ABC-DA-ANN showcases improved performance, yielding CA and FM values of 62.255 and 60.875, respectively. This amalgamation of optimization techniques within a neural network framework enables ABC-DA-ANN to achieve better classification accuracy and a more balanced F-Measure than LSO-FFNN.

The challenges, such as parameter tuning complexities and potential algorithmic conflicts, may affect its robustness and scalability, necessitating careful optimization and fine-tuning. PRDO-RVM emerges as the frontrunner, exhibiting remarkable CA and FM scores of 87.325 and 87.620, respectively. This algorithm's proficiency lies in its efficient optimization strategies and robust classification capabilities, complemented by the sparsity-inducing properties of the Relevance Vector Machine. PRDO-RVM achieves high classification accuracy and a balanced F-Measure, making it a compelling choice for classification tasks across diverse domains.

7. Conclusion

This research addresses the critical need for robust I.D.S. in MANETs to mitigate the diverse security threats in these dynamic and decentralized networks. By introducing the PRDO-RVM, the research proposes an innovative approach to intrusion detection that leverages the sparsity-inducing properties of RVM and the efficient optimization capabilities of Red Deer Optimization. By evaluating PRDO-RVM using the NSK-KDD dataset, the study demonstrates its superior classification accuracy and efficiency compared to existing I.D.S. solutions. This underscores the potential of PRDO-RVM as a reliable and scalable security mechanism for MANETs, offering enhanced protection against malicious activities and ensuring the integrity and availability of network communications.

As MANETs continue to play a vital role in various applications, the findings of this research contribute significantly to advancing the security posture of MANETs and addressing the evolving challenges posed by malicious actors in dynamic network environments.

Future enhancements may include integrating Krill Herd Optimization and RF to improve intrusion detection accuracy in MANETs. This hybrid approach leverages K.H.O.'s optimization capabilities and RF's ensemble learning framework to address the dynamic nature of security threats.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this work.

Funding Statement

All research processes and findings have been conducted independently without financial, personal, or professional influences.

References

[1] Quy Khanh Vu, Nam Vi Hoai, and Linh Dao Manh, "A Survey of State-of-the-Art Energy Efficiency Routing Protocols for MANET," *International Journal of Interactive Mobile Technologies*, vol. 14, no. 9, pp. 215-226, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[2] Raed Alsaqour et al., "Genetic Algorithm Routing Protocol for Mobile Ad Hoc Network," *Computers, Materials & Continua*, vol. 68, no. 1, pp. 941-960, 2021.[CrossRef] [Google Scholar] [Publisher Link]

[3] Lavanya Poluboyina et al., "Multimedia Traffic Transmission Using MAODV and M-MAODV Routing Protocols Over Mobile Ad-hoc Networks," *International Journal of Computer Network and Information Security*, vol. 14, no. 3, pp. 47-62, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[4] Faik Kerem Örs, and Albert Levi, "Data Driven Intrusion Detection for 6LoWPAN Based IoT Systems," *Ad Hoc Networks*, vol. 143, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[5] R. Thanuja, and A. Umamakeswari, "Black Hole Detection Using Evolutionary Algorithm for IDS/IPS in MANETs," *Cluster Computing*, vol. 22, pp. 3131-3143, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[6] S. Sankara Narayanan, K. Chidambarathanu, and L.C. Meena, "PFR Based Technique to Detect Intruder in MANET," *Journal of Advanced Research in Dynamical and Control Systems*, vol. 12, no. 2 pp. 597-601, 2020. [Publisher Link]

[7] N. Rajendran, P.K. Jawahar, and R. Priyadarshini, "Makespan of Routing and Security in Cross Centric Intrusion Detection System (CCIDS) Over Black Hole Attacks and Rushing Attacks in MANET," *International Journal of Intelligent Unmanned Systems*, vol. 7, no. 4, pp. 162-176, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[8] Nisha Soms, and P. Malathi, "Secured and Anonymous Data Transmission in Manet Environment Using Zone-Based Intrusion Detection System," *Concurrency and Computation Practice and Experience*, vol. 31, no. 12, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[9] N. Rajendran, P.K. Jawahar, and R. Priyadarshini, "Cross Centric Intrusion Detection System for Secure Routing Over Black Hole Attacks in MANETs," *Computer Communications*, vol. 148, pp. 129-135, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[10] Shalini Subramani, and M. Selvi, "Multi-Objective PSO Based Feature Selection for Intrusion Detection in IOT Based Wireless Sensor Networks," *Optik*, vol. 273, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[11] Gustavo de Carvalho Bertoli et al., "Generalizing Intrusion Detection for Heterogeneous Networks: A Stacked-Unsupervised Federated Learning Approach," *Computers and Security*, vol. 127, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[12] Durgesh Srivastava et al., “A Framework for Detection of Cyber Attacks by The Classification of Intrusion Detection Datasets,” *Microprocessors and Microsystems*, vol. 105, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[13] Hossein Asgharzadeh et al., “Anomaly-Based Intrusion Detection System in The Internet of Things Using a Convolutional Neural Network and Multi-Objective Enhanced Capuchin Search Algorithm,” *Journal of Parallel and Distributed Computing*, vol. 175, pp. 1-21, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[14] Judy Simon et al., “Hybrid Intrusion Detection System for Wireless IOT Networks Using Deep Learning Algorithm,” *Computers and Electrical Engineering*, vol. 102, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[15] Vikash Kumar, and Ditipriya Sinha, “Synthetic Attack Data Generation Model Applying Generative Adversarial Network for Intrusion Detection,” *Computers and Security*, vol. 125, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[16] F. Folino et al., “On Learning Effective Ensembles of Deep Neural Networks for Intrusion Detection,” *Information Fusion*, vol. 72, pp. 48-69, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[17] Murad Ali Khan et al., “An Optimized Ensemble Prediction Model Using Automl Based on Soft Voting Classifier for Network Intrusion Detection,” *Journal of Network and Computer Applications*, vol. 212, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[18] Thippa Reddy Gadekallu et al., “Moth-Flame Optimization Based Ensemble Classification for Intrusion Detection in Intelligent Transport System for Smart Cities,” *Microprocessors and Microsystems*, vol. 103, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[19] Xu Zhao et al., “Task Offloading of Cooperative Intrusion Detection System Based on Deep Q Network in Mobile Edge Computing,” *Expert Systems with Applications*, vol. 206, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[20] Liu Zhiqiang et al., “Intrusion Detection in Wireless Sensor Network Using Enhanced Empirical Based Component Analysis,” *Future Generation Computer Systems*, vol. 135, pp. 181-193, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[21] Hind Bangui, Mouzhi Ge, and Barbora Buhnova, “A Hybrid Data-Driven Model for Intrusion Detection in VANET,” *Procedia Computer Science*, vol. 184, pp. 516-523, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[22] I. Benmessahel, K. Xie, M. Chellal, and T. Semong, “A New Evolutionary Neural Networks Based on Intrusion Detection Systems Using Locust Swarm Optimization,” *Evolutionary Intelligence*, vol. 12, no. 2, pp. 131–146, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[23] Waheed Ali H. M. Ghanem et al., “An Efficient Intrusion Detection Model Based on Hybridization of Artificial Bee Colony and Dragonfly Algorithms for Training Multilayer Perceptrons,” *IEEE Access*, vol. 8, pp. 130452-130475, 2020. [CrossRef] [Google Scholar] [Publisher Link]