Journal of Theoretical and Applied Information Technology
15th October 2025. Vol.103. No.19

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

ADAPTIVE CARIBOU DEFENSE PROTOCOL (ACDP): A BIO-INSPIRED INTRUSION DETECTION FRAMEWORK FOR SECURING IOMT NETWORKS

BASIL BABY K[1] , Dr. A. NITHYA RANI[2]
[1]PhD Research Scholar, CMS College of Science and Commerce, Chinnavedampatti, Coimbatore, India
[2]Associate Professor, CMS College of Science and Commerce Chinnavedampatti, Coimbatore, India
E-mail: [1]mail4basilbaby@gmail.com, [2]nithyarani.a@gmail.com

ABSTRACT

Network security has become a critical concern due to the increasing complexity of cyber threats targeting interconnected systems. The "Internet of Medical Things" (IoMT) has transformed healthcare by enabling real-time monitoring, remote diagnostics, and automated medical interventions. Integrating IoMT devices into healthcare infrastructures exposes networks to security vulnerabilities, requiring robust intrusion detection mechanisms. "Host-based intrusion Detection Systems" (HIDS) provide a localized security approach, monitoring system logs, processes, and behaviors to detect unauthorized activities. Traditional detection techniques often struggle with evolving threats and resource limitations in IoMT environments. Bio-inspired optimization techniques offer adaptive security enhancements, refining detection mechanisms while minimizing computational overhead. The Adaptive Caribou Defense Protocol (ACDP) leverages nature-inspired intelligence to optimize intrusion detection, ensuring enhanced security resilience. By integrating bio-inspired approaches with HIDS, intrusion detection frameworks can achieve improved adaptability, real-time threat identification, and efficient security enforcement across IoMT networks, mitigating emerging cyber risks effectively.

Keywords: *Host Intrusion Detection Systems - Internet of Medical Things - Intrusion Detection – Cybersecurity in Healthcare - Caribou Optimization*

1. INTRODUCTION

"Wireless Sensor Networks" (WSNs) have revolutionized data collection, transmission, and processing in various sectors, particularly in healthcare. These networks consist of interconnected sensor nodes that monitor physiological parameters, environmental conditions, and medical processes [1]. WSNs are crucial in remote patient monitoring, emergency response systems, and hospital automation. The ability of these networks to collect real-time medical data has enhanced healthcare efficiency, reduced manual intervention, and improved the quality of patient care [2]. Due to their open communication channels and dependency on wireless connectivity, WSNs remain vulnerable to cyber threats, including data breaches, unauthorized access, and network intrusions. Securing these networks requires advanced security mechanisms that detect, prevent, and respond to cyber threats in real time [3].

The "Internet of Medical Things" (IoMT) extends the principles of WSNs into a more advanced interconnected ecosystem, integrating medical devices, wearable sensors, and cloud-based healthcare platforms [4]. IoMT enables seamless communication between smart medical devices, electronic health records (EHRs), and remote healthcare providers, ensuring real-time diagnostics, patient monitoring, and medical intervention. The growing dependence on IoMT has introduced new challenges concerning data security, privacy, and network integrity [5]. IoMT devices, often operating on resource-constrained platforms, face significant risks such as malware infections, data tampering, and denial-of-service (DoS) attacks. Since these medical devices interact with sensitive patient data, the consequences of security breaches can be severe, leading to compromised patient safety, unauthorized alterations in medical prescriptions, and disruptions in critical healthcare services [6].

"Intrusion Detection Systems" (IDS) is a critical security mechanism in IoMT, ensuring early detection of malicious activities, unauthorized network access, and suspicious behavioral patterns [7]. Traditional cybersecurity measures such as

encryption and authentication provide a foundational level of security, but they remain insufficient against advanced cyber threats that continuously evolve. Intrusion detection in IoMT involves monitoring real-time network traffic, analyzing system logs, and identifying anomalies that indicate potential security breaches. IoMT environments require specialized intrusion detection techniques that consider the constraints of medical devices, including limited processing power, low energy consumption, and stringent real-time requirements [8]. By incorporating IDS, healthcare infrastructures can safeguard medical data, ensure device integrity, and maintain seamless communication across IoMT networks [8].

Among the various intrusion detection techniques, "Host-Based Intrusion Detection Systems" (HIDS) are crucial in securing IoMT networks. Unlike "Network-Based Intrusion Detection Systems" (NIDS), which focus on monitoring traffic across an entire network, HIDS operates at the device level, detecting anomalies, unauthorized access, and system irregularities directly within IoMT endpoints [9]. HIDS analyzes system logs, file integrity, and process activities, identifying threats such as malware infections, privilege escalation, and unauthorized modifications in medical applications. Since IoMT devices function in diverse environments ranging from hospital infrastructure to wearable healthcare monitoring systems, HIDS provides a localized security approach that strengthens individual device security and prevents network-wide disruptions [10].

HIDS performs real-time monitoring of system activities, detecting deviations from predefined security policies. By examining system logs and behavior patterns, HIDS identifies unauthorized changes in file structures, configuration settings, and software execution flows. IoMT devices rely on stable and predictable operational behaviors; thus, any anomaly detected by HIDS is an early warning for potential security threats [11]. HIDS offers forensic capabilities, allowing security administrators to analyze logs and determine the origin of an attack, thereby facilitating rapid threat mitigation and future prevention strategies [12]. HIDS in IoMT also enhances security resilience by providing behavioral-based detection mechanisms. Unlike signature-based detection, which relies on predefined attack patterns, behavioral-based detection in HIDS examines real-time deviations from normal device operations. IoMT devices frequently communicate with cloud storage, remote healthcare providers, and centralized hospital management systems, making them susceptible to novel cyber threats [13]. HIDS continuously adapts to evolving threat landscapes, identifying previously unknown attack patterns that traditional security methods might overlook. This adaptive security mechanism ensures that IoMT networks remain protected against zero-day vulnerabilities and advanced persistent threats (APTs) [13].

Integrating HIDS with IoMT frameworks requires efficient optimization techniques to address resource constraints associated with medical devices. Since IoMT devices often operate with limited processing power and battery life, HIDS implementations must be lightweight, ensuring minimal computational overhead. Advanced optimization algorithms enhance HIDS performance, allowing real-time threat detection without compromising device efficiency [14]. By prioritizing essential security checks and minimizing redundant processes, optimized HIDS solutions ensure effective intrusion detection while maintaining uninterrupted medical functionalities [15]. HIDS also plays a vital role in compliance and regulatory adherence within IoMT security frameworks. Healthcare infrastructures must comply with stringent data protection regulations such as the "Health Insurance Portability and Accountability Act" (HIPAA) and the "General Data Protection Regulation" (GDPR). These regulations mandate the secure handling of medical data, ensuring confidentiality, integrity, and availability [16]. HIDS solutions help enforce regulatory compliance by detecting unauthorized data access, ensuring file integrity, and monitoring system logs for security violations. By maintaining comprehensive audit trails and security logs, HIDS enables healthcare organizations to meet regulatory requirements while enhancing overall security resilience.

In IoMT environments, HIDS is a proactive defense mechanism, mitigating security risks before they escalate into major threats. By integrating HIDS with machine learning algorithms, security frameworks can predict potential intrusions based on historical data patterns, improving threat response mechanisms [17]. Predictive security models combined with HIDS enhance real-time intrusion detection capabilities, ensuring robust protection for medical devices, patient records, and healthcare communication networks. The evolving landscape of IoMT security demands a multi-layered security approach, where HIDS functions as a fundamental component in protecting individual

devices from cyber threats [18]. Continuous advancements in HIDS technology contribute to intrusion detection mechanisms' resilience, adaptability, and reliability, ensuring secure medical environments and patient safety across interconnected healthcare infrastructures [19].

Bio-inspired optimization draws inspiration from natural processes to develop efficient problem-solving techniques in various computational domains. Evolutionary behaviors, swarm intelligence, and ecological adaptations provide the foundation for optimization algorithms that enhance decision-making, pattern recognition, and resource allocation [20], [21]. Nature-inspired models, such as genetic algorithms, ant colony optimization, and particle swarm optimization, mimic biological mechanisms to solve complex challenges efficiently. These methods enable adaptive learning, self-organization, and dynamic problem-solving, making them suitable for network security, robotics, and machine-learning applications [22]. In intrusion detection, bio-inspired optimization improves classification accuracy, reduces false positives, and enhances detection efficiency. Optimization techniques refine security models by emulating strategies from wildlife, such as caribou migration and foraging behavior, enabling real-time adaptation to evolving cyber threats. Integrating bio-inspired approaches with HIDS strengthens network resilience, ensuring a proactive defense mechanism for safeguarding critical infrastructures, including IoMT environments [23].

1.1. Challenges

Intrusion detection in the IoMT faces multiple challenges due to the complexity of interconnected medical devices. Resource constraints in IoMT devices limit the computational capabilities required for efficient HIDS. Real-time monitoring demands impose high processing loads, affecting device performance and energy efficiency. The evolving nature of cyber threats introduces sophisticated attack patterns that bypass traditional detection mechanisms. Ensuring seamless integration of HIDS with IoMT frameworks requires optimization techniques to minimize latency and false positives. Compliance with regulatory frameworks such as HIPAA and GDPR complicates security implementation. Secure data transmission and encrypted storage remain critical concerns, particularly in remote patient monitoring systems and cloud-based healthcare platforms.

1.2 Motivation and Objective

The increasing adoption of the IoMT has introduced significant security concerns, necessitating robust intrusion detection mechanisms. Medical devices, electronic health records, and remote monitoring systems require protection from cyber threats that could compromise patient safety and data integrity. HIDS provides a localized security approach, ensuring device-level monitoring and anomaly detection. The primary motivation is to enhance real-time threat identification while minimizing computational overhead in resource-constrained IoMT environments. The objective is to develop an optimized HIDS framework that ensures accurate intrusion detection, reduces false positives, and aligns with regulatory compliance. Strengthening security resilience in IoMT networks ensures uninterrupted healthcare services while mitigating unauthorized access and data breach risks.

1.3 Research Gap

Existing security frameworks in the IoMT lack efficient intrusion detection mechanisms tailored for resource-constrained medical devices. Traditional "Network-Based Intrusion Detection Systems" (NIDS) focus on network traffic but fail to address security threats at the device level. HIDS offers localized monitoring, yet current implementations struggle with high computational overhead, leading to inefficiencies in real-time threat detection. Adaptive optimization techniques for HIDS in IoMT remain underexplored, limiting the ability to mitigate emerging cyber threats effectively. Existing models also exhibit high false-positive rates, reducing reliability in intrusion detection. Addressing these gaps requires a lightweight, adaptive HIDS framework that enhances security while ensuring minimal impact on device performance and real-time healthcare operations.

2. LITERATURE REVIEW

"DSRNN-ISCOA" [24] integrated a dynamically stabilized recurrent neural network (DSRNN) with an intensified sand cat swarm optimization (ISCOA) technique for securing wireless sensor networks (WSNs). An adaptive multi-scale differential filter preprocessed data by removing redundancies, while the Wolf-Bird Optimization Algorithm selected relevant features. DSRNN classified network traffic, detecting black holes, grey holes, flooding, and TDMA attacks. ISCOA optimized DSRNN's weight parameters, enhancing accuracy by adapting to attack patterns.

"Boost-WSN-IDS" [25] introduced a LEACH-based dataset simulating DoS attacks in wireless sensor networks, including wormhole, black hole, grey hole, flooding, and TDMA-based attacks. Boosting-based models such as LightGBM, XGBoost, and Bagging were used for intrusion detection. Feature selection minimized computational complexity while maintaining high accuracy. LightGBM, with its leaf-wise growth strategy, excelled in memory efficiency and speed, making it ideal for resource-limited WSNs.

"E2E-CNN1D" [26] introduced a lightweight 1D convolutional neural network for detecting advanced cyber threats in industrial IoT networks. Using the Edge-IIoTset dataset with 14 attack categories, the model employed end-to-end learning, eliminating extensive feature engineering. A preprocessing step normalized input data for consistency, while CNN1D extracted hierarchical features to capture local and temporal attack patterns. K-fold cross-validation improved generalization and reduced overfitting. "UAV-IDS Datasets" [27] analyzed datasets for intrusion detection in UAV communication networks, categorizing them based on intra-UAV and inter-UAV security challenges. Key factors like attack types, data distribution, and network protocols were evaluated to assess suitability for machine learning-based IDS. A novel taxonomy highlighted gaps in existing datasets and the need for broader attack scenarios. Recommendations were provided for dataset selection based on UAV network configurations and threat models. "MAFA-LSTM" [28] combined a memetic self-adaptive firefly algorithm (MAFA) with LSTM for intrusion detection in IoT networks. A perturbation operator in MAFA prevented local optima, ensuring optimal LSTM hyperparameters. After noise removal and feature normalization, MAFA selected the most relevant security parameters, which LSTM then analyzed for temporal attack patterns. This hybrid approach improved precision, recall, and accuracy, surpassing traditional deep learning methods.

"SA-PVAE-GAN" [29] introduced a security framework for wireless sensor networks (WSN) by integrating self-attention, provisional variational auto-encoders (PVAE), and GANs. A preprocessing module extracted key network features, while PVAE encoded data into a latent space, learning normal and attack traffic distributions. Self-attention improved feature learning by capturing long-range dependencies. A GAN-based approach generated synthetic attack samples to enhance training, with a discriminator refining intrusion detection. "DRL-IDS Guide" [30] explored deep reinforcement learning (DRL) for intrusion detection in IoT networks, analyzing architectures, training strategies, and real-world applications. A design framework focused on reward functions, action spaces, and state representations while addressing exploration-exploitation trade-offs and computational constraints. Key challenges such as training instability, high-dimensional action spaces, and adversarial attacks were identified. The study emphasized federated learning, transfer learning, and self-adaptive DRL for improving detection.

"ML-DDOS-SDIoT" [31] introduced a machine learning-based security framework to mitigate DDoS attacks in software-defined IoT (SD-IoT) networks. A feature engineering pipeline extracted key traffic patterns, while a multi-stage classifier combining SVM, RF, and DNN ensured hierarchical threat detection. The SDN controller dynamically adjusted flow rules based on real-time IDS feedback, with a feedback loop enabling adaptive retraining. An anomaly detection mechanism flagged threats before full-scale attacks, reducing detection latency and resource usage.

"M-CNN-IDS" [32] introduced an optimized CNN-based intrusion detection system (IDS) for enhanced cybersecurity. An advanced feature extraction layer captured spatial and temporal attack patterns more effectively than standard CNNs. A lightweight architecture enabled deployment in resource-constrained environments while batch normalization and dropout layers reduced overfitting. A hybrid data augmentation technique improved the detection of rare attacks. "E-IDS-WSN" [33] introduced an E-shaped machine learning framework for intrusion detection in WSN. It featured three core components: feature selection, ensemble classification, and adaptive security policies. An evolutionary optimization algorithm selected key security attributes, while an ensemble of classifiers (decision trees, SVMs, and RNNs) improved detection accuracy. The model dynamically adjusted classification thresholds based on network conditions for real-time adaptability. Reinforcement learning-based security policies enabled continuous evolution against emerging threats. "RBM-LSTM-IDS" [34] combined Restricted Boltzmann Machine (RBM) and Long Short-Term Memory (LSTM) networks for detecting routing attacks in IoT networks. RBM extracted high-relevance features from raw traffic, while LSTM analyzed sequential patterns to identify anomalies like blackhole and wormhole attacks. A dynamic thresholding mechanism

adapted to network conditions, minimizing false positives. The model continuously learned from traffic behavior, distinguishing legitimate routing changes from attacks.

"WOGRU-IDS" [35] combined the Whale Optimization Algorithm (WOA) and Gated Recurrent Unit (GRU) networks for intrusion detection in IoT-assisted Wireless Sensor Networks (WSNs). WOA selected optimal features, reducing computational overhead while preserving accuracy. The GRU model analyzed these features, detecting threats like sinkholes, Sybil, and selective forwarding attacks. With adaptive learning, the system updated itself to counter evolving threats. Unlike traditional IDS, it achieved high accuracy with low energy consumption, making it ideal for resource-constrained IoT-WSN environments. "GA-RF-IDS" [36] integrated the Genetic Algorithm (GA) with Random Forest (RF) to enhance intrusion detection in IoT networks. GA optimized RF's decision trees, selecting the most relevant features for classification. The network was divided into subdomains, each managed by a controller node running the optimized RF model. These nodes operated independently or cooperatively, analyzing traffic and detecting threats while balancing precision and recall to minimize false positives. Extensive testing on NSW-NB15 and NSL-KDD datasets showed higher accuracy than traditional RF-based IDS.

"D-NIDS" [37] introduces a domain-invariant network intrusion detection system to improve threat detection across different network environments. The model leverages deep learning techniques to extract invariant features, ensuring consistent performance in varying domains. The system enhances generalization and robustness against cyber threats by addressing distribution shifts in network traffic data. The approach minimizes dependency on specific datasets, making it adaptable to diverse network conditions. Through advanced feature learning and anomaly detection, DI-NIDS effectively identifies malicious activities, providing a scalable and reliable cybersecurity solution for modern network infrastructures.

"D-MAN" [38] presents an effective technique for detecting minority attacks in network intrusion detection systems (NIDS) using deep learning and sampling strategies. The approach addresses data imbalance by employing advanced sampling techniques to enhance minority class detection. A deep learning model is trained on enriched datasets, improving sensitivity to rare cyber threats. By refining feature extraction and classification, the system enhances accuracy in identifying underrepresented attack types. The method ensures robust intrusion detection, reducing false negatives and strengthening cybersecurity defenses. This framework provides a more balanced and efficient solution for detecting minority attacks in evolving network environments.

Bio-inspired optimization in the research outcomes demonstrates how natural foraging strategies can strengthen intrusion detection by enhancing adaptability under dynamic IoMT conditions [39] - [50]. The results validate that the algorithm reduces false alarms and improves detection accuracy through elite selection and adaptive search [51] - [64]. This shows the study's contribution in transferring biologically inspired intelligence into practical security mechanisms for resource-constrained medical devices [65] – [78].

3. ADAPTIVE CARIBOU DEFENSE PROTOCOL (ACDP)

The Adaptive Caribou Defense Protocol (ACDP) begins with the essential step of initializing the caribou herd, which simulates the setup of potential solutions within the HIDS. Each caribou in this initialization phase represents a unique configuration of HIDS parameters designed to establish a diverse starting population of possible solutions. This step lays the groundwork for the optimization process, ensuring that the HIDS can adapt, evolve, and become more effective at detecting intrusions.

3.1 Parameter Definition and Herd Diversity

The first objective in initializing the caribou herd is creating various parameter values for the HIDS. Each caribou, representing a specific configuration, is defined by multiple variables that collectively determine the behavior and sensitivity of the HIDS. Let a vector denote the configuration of each caribou X, as shown in Eq.(1).

$$X = [x_1, x_2, x_3, \dots, x_n] \quad (1)$$

X signifies a complete configuration vector for a single caribou, with each element x_i representing a specific HIDS parameter. These parameters could include thresholds for anomaly detection, logging intervals, and resource allocation levels. The initial population of caribou is selected to ensure diversity across these parameters, promoting a broad exploration of the solution space. Diverse configurations help prevent early convergence to

suboptimal solutions, a common pitfall in optimization processes. The caribou initialization step actively enhances HIDS's adaptive capabilities by populating the herd with configurations spanning a wide range of values.

Resource Allocation in Herd Initialization

Optimizing resource allocation plays a key role in the initial setup of the caribou herd. Each caribou configuration allocates system resources such as CPU and memory to maximize HIDS efficiency without overburdening the host system. The resource allocation for each caribou, denoted as R, can be formulated as Eq.(2).

$$R = \sum_{i=1}^{n} \alpha_i \cdot x_i \quad (2)$$

where α_i indicates the weight of resource impact for each parameter x_iThis equation helps balance the configuration to utilize optimal resources efficiently. Optimized resource allocation ensures that each caribou's configuration remains viable within the HIDS, maximizing performance without causing significant system overhead.

Distance Between Configurations

Caribou within the herd exhibit differences in configuration, creating a measure of "distance" between each pair to maintain diversity. This distance metric, D, evaluates the distinction between the parameter settings of two configurations, labeled X_i and X_j.

$$D(X_i, X_j) = \sqrt{\sum_{k=1}^{n} (x_{i,k} - x_{j,k})^2} \quad (3)$$

In Eq.(3), the distance $D(X_i, X_j)$ Captures the variance across the parameters between two configurations, ensuring various distinct solutions at the onset. Maintaining a diverse distance profile enhances the likelihood of discovering an optimized solution through caribou behavior.

Objective Function for Herd Members

The initialization process requires defining an objective function each caribou will attempt to optimize during the migration and adaptation phases. The objective function $f(X)$ Evaluate the effectiveness of each caribou configuration in detecting potential intrusions, expressed mathematically in Eq.(4).

$$f(X) = \gamma \cdot Accuracy + \delta \cdot Resource\ Efficiency \quad (4)$$

where γ and δ represent the weights for accuracy and resource efficiency. Balancing these elements ensures that each caribou configuration remains aligned with HIDS's goal to detect intrusions while conserving system resources.

Migration Potential

The initialization step assigns each caribou a migration potential M, representing its capacity to move within the parameter space toward better configurations. This migration potential is calculated as Eq.(5).

$$M = \beta \cdot D(X, X_{best}) \quad (5)$$

where β indicates the migration influence factor and X_{best} It is the configuration with the highest initial objective function value. This potential prepares the herd to move toward optimized solutions in subsequent steps.

Intrusion Detection Sensitivity Adjustment

As part of initialization, the sensitivity of intrusion detection in each configuration is set. Sensitivity S determines how aggressively HIDS flags potential threats, calculated for each caribou as shown in Eq.(6).

$$S = \frac{\sum_{i=1}^{n} x_i}{n} \quad (6)$$

This sensitivity score S balances the detection aggressiveness of HIDS, ensuring a range of conservative to aggressive configurations within the herd, which supports comprehensive intrusion detection.

Resource Constraint Verification

Each caribou configuration must adhere to resource constraints to ensure it does not compromise host system stability. A constraint check, $C(X)$, confirms compliance with resource limits, defined by Eq.(7).

$$C(X) = \sum_{i=1}^{n} x_i \leq R_{max} \quad (7)$$

where R_{max} represents the maximum allowable resource usage. Configurations exceeding R_{max} are excluded, maintaining HIDS functionality without excessive resource consumption.

Configuration Adaptability Scoring

Each caribou's adaptability, denoted as A, measures its potential to adjust and respond to changing security conditions. The adaptability score is computed as expressed in Eq.(8).

$$A = \eta \cdot (M + f(X)) \quad (8)$$

Where η indicates the adaptability factor. A high adaptability score signals configurations that can dynamically evolve, guiding the herd toward adaptable, optimized solutions.

3.2 Defining Caribou Movement Patterns

This phase provides a systematic approach to adjust configurations for HIDS optimization. This step aims to simulate the migratory behaviour of caribou to locate high-performing configurations while minimizing resource usage. Implementing movement strategies that mimic caribou allows adaptive parameter space exploration, enabling HIDS to evolve toward configurations with enhanced intrusion detection capabilities. The movement patterns, mathematically represented, dictate each caribou's direction and step size within the parameter landscape, simulating how caribou migrate toward favourable regions for optimized results.

Caribou Migration Model

In defining movement, caribou utilizes a migration model that enables movement toward configurations showing higher objective function values. The migration distance for each caribou M_d can be expressed as Eq.(9).

$$M_d = \lambda \cdot (X_{best} - X_{current}) \quad (9)$$

where λ denotes the migration factor, X_{best} is the position of the configuration with the highest objective function value, and $X_{current}$ represents the current position of the caribou. This equation encourages caribou to move closer to configurations with favourable attributes, simulating the natural tendency to migrate toward optimal conditions.

Position Update Based on Migration Direction

Each caribou's location in the parameter space is updated according to its migration distance to adjust position. The new position X_{new} for each caribou is calculated as Eq.(10).

$$X_{new} = X_{current} + M_d \quad (10)$$

This equation updates the caribou's position by adding the migration distance to the current location. The adjustment brings each caribou closer to optimized configurations, moving it in line with the path set by high-performance points.

Exploration and Diversification of Path

Diversification factors are introduced to increase exploratory movement and prevent the caribou from converging prematurely on local optima. A diversification term D_f for each caribou can be expressed as Eq.(11).

$$D_f = \zeta \cdot |X_{random} - X_{current}| \quad (11)$$

where ζ represents the diversification factor and X_{random} is a randomly selected configuration within the search space. Diversification expands the search space by encouraging movement away from the current location, balancing exploration and exploitation.

Adaptive Step Size Adjustment

Step size determines how rapidly caribou can move toward optimal configurations, and it adapts based on the performance of previous movements. The step size S_s for each caribou is defined as Eq.(12).

$$S_s = \rho \cdot \left(\frac{f(X_{best}) - f(X_{current})}{|X_{best} - X_{current}| + \epsilon} \right) \quad (12)$$

where ρ is a scaling factor, $f(X_{best})$ and $f(X_{current})$ denote the objective function values at the best and current positions, and ϵ is a small constant to avoid division by zero. The step size adjusts adaptively to ensure movement efficiency.

Movement Angle for Optimized Positioning

Directionality in movement is influenced by calculating an angle θ that helps determine the caribou's precise path toward the optimized region. The movement angle θ can be determined as Eq.(13).

$$\theta = tan^{-1}\left(\frac{x_{best} - x_{current}}{y_{best} - y_{current}} \right) \quad (13)$$

where x_{best} and y_{best} are the coordinates of the best configuration in a two-dimensional parameter space, and $x_{current}$ and $y_{current}$ represent the coordinates of the caribou's current position. This angle determines the directional vector along which the caribou moves, refining its path.

Dynamic Adjustment of Migration Influence

The migration influence dynamically adjusts to balance the herd's movement between

exploration and convergence. The migration influence M_i is defined by Eq.(14).

$$M_i = \tau \cdot e^{-k.t} \quad (14)$$

where τ is the initial migration influence, κ is a decay constant, and t represents the number of iterations. This exponential decay function reduces migration influence over time, allowing the caribou to fine-tune positions closer to the optimal point without excessive deviation.

Optimization of Caribou Positions in Movement

In coordination with migration patterns, positioning adjustments ensure that caribou configurations continue optimizing over time. Optimized positioning, P_{opt}, can be modelled as Eq.(15).

$$P_{opt} = X_{new} + S_s \cdot D_f \cdot \cos(\theta) \quad (15)$$

where X_{new} is the updated position, $S_s$represents step size, D_f is the diversification term, and θ is the movement angle. This equation calculates an optimized position, factoring in the caribou's adjusted movement parameters to achieve improved configuration alignment.

3.3 Foraging Behavior Simulation

A dynamic approach to fine-tune HIDS parameters by simulating the foraging behaviour of caribou. This process enables each configuration to explore promising areas of the parameter space that are likely to yield enhanced detection capabilities. Foraging here refers to seeking optimized settings that maximize HIDS's performance in identifying threats. The foraging behaviour maintains the HIDS system's adaptability and effectiveness by continuously adjusting and evaluating configurations.

Fitness Evaluation in the Foraging Process

The foraging behaviour begins by evaluating the fitness of each caribou's configuration, representing the configuration's effectiveness in intrusion detection. The fitness F of each configuration is determined by a specific evaluation function, calculated as represented mathematically in Eq.(16).

$$F = \omega \cdot Detection\ Rate - \psi \cdot Resource\ Usage \quad (16)$$

where ω denotes the weight assigned to the detection rate, and ψ indicates the penalty weight for resource usage. This function prioritizes configurations with high detection accuracy and low resource consumption, guiding caribou to forage in regions that optimize HIDS performance.

Search Radius Determination

The foraging behaviour includes setting a search radius R_s around each caribou's position, limiting the area where the configuration will be explored. The search radius can be defined as Eq.(17).

$$R_s = \phi \cdot \sqrt{|X_{best} - X_{current}|} \quad (17)$$

where ϕ represents the scaling factor of the search radius, and X_{best} and $X_{current}$ indicate the positions of the best-performing configuration and the caribou's current position, respectively. This controlled radius allows focused exploration around promising configurations.

Directional Adjustment Toward Optimal Solutions

As part of the foraging process, each caribou adjusts its direction based on the position of nearby high-fitness configurations. The directional vector D_v is calculated as Eq.(18).

$$D_v = \gamma \cdot \frac{X_{optimal} - X_{current}}{|X_{optimal} - X_{current}|} \quad (18)$$

where γ is a directional influence factor, $X_{optimal}$ denotes the position of the nearest high-fitness configuration, and $X_{current}$ represents the current configuration. This vector directs the caribou to align its position towards configurations demonstrating optimized performance.

Probabilistic Selection of Neighboring Points

The foraging behaviour simulates the tendency to explore neighbouring points probabilistically, allowing for both intensification and diversification of the search. A probability function P_s defines the likelihood of selecting neighbouring points as Eq.(19).

$$P_s = \frac{F(X_{neighbor})}{\sum_j F(X_j)} \quad (19)$$

where $F(X_{neighbor})$ is the fitness of a neighbouring configuration, and $\sum_j F(X_j)$ represents the total fitness of all considered neighbours. This probability encourages selection based on fitness, guiding exploration to high-performance areas.

Energy-Based Foraging Adjustment

To simulate resource constraints in foraging, an energy factor E_f limits the number of iterations a configuration can spend exploring a particular region. The energy factor is calculated as Eq.(20).

$$E_f = E_{initial} - \delta \cdot T \quad (20)$$

where $E_{initial}$ is the initial energy assigned to the caribou, δ is the decay rate, and T represents the time or number of iterations in the foraging process. As energy depletes, the caribou shifts focus, preventing the over-exploration of a single area.

Adaptive Step Size During Foraging

To ensure efficient movement, each caribou's step size adapts during foraging based on the fitness of nearby configurations. The adaptive step size S_f for foraging is expressed as Eq.(21).

$$S_f = \theta \cdot \left(\frac{F(X_{neighbor}) - F(X_{current})}{|X_{neighbor} - X_{current}| + \epsilon} \right) \quad (21)$$

where θ is the step-size scaling factor, $F(X_{neighbor})$ and $F(X_{current})$ denote fitness values of the neighbour and current configurations, and ϵ prevents division by zero. This step size adjustment ensures that configurations adapt according to the quality of surrounding points.

3.4 Adaptation to Environmental Changes

This step simulates the caribou's real-world adaptability to shifting conditions, enhancing the system's resilience by dynamically adjusting HIDS configurations. Through mathematical adjustments, caribou adapt their parameters, optimizing HIDS performance across different environmental contexts, such as variations in network traffic or user behaviour. The adaptability mechanism incorporates factors including sensitivity, responsiveness, and adjustment thresholds, ensuring that each caribou configuration remains effective despite environmental shifts.

Environmental Change Detection

The first task in environmental adaptation involves detecting changes in the HIDS environment. Environmental change E_c a detection function monitors variations in key indicators such as traffic volume or anomaly patterns, expressed mathematically as Eq.(22).

$$E_c = \xi \cdot |A_t - A_{t-1}| \quad (22)$$

where ξ is a sensitivity factor, A_t represents the anomaly rate at the current time, and A_{t-1} indicates the anomaly rate from the previous time step. This difference signals an environmental change requiring parameter adjustment.

Configuration Sensitivity Adjustment

Upon detecting environmental change, each caribou configuration adjusts its sensitivity to align with the new conditions. The sensitivity S_a for each configuration is recalculated as shown in Eq.(23).

$$S_a = S_{base} + \alpha \cdot E_c \quad (23)$$

where S_{base} is the baseline sensitivity, and α denotes the adjustment factor in response to detected changes. This equation amplifies the configuration's responsiveness to emerging threats while maintaining a controlled sensitivity level.

Threshold Modulation for Resource Efficiency

Adaptation to environmental changes involves recalibrating the thresholds that govern system resource usage. The threshold T_r is modified in response to shifts in anomaly rates, calculated as shown in Eq.(24).

$$T_r = T_{initial} - \beta \cdot E_c \quad (24)$$

where $T_{initial}$ represents the initial resource threshold, and β is the decay rate based on the extent of environmental change. Lowering T_r during high anomaly periods prioritizes resources for detection, maximizing HIDS's operational efficiency.

Adaptive Reweighting of Objective Function

The objective function balances detection accuracy and resource usage and adapts by reweighting components according to environmental shifts. The adapted objective function f_a is expressed as shown in Eq.(25).

$$f_a = (\gamma + \delta \cdot E_c) \cdot Detection\ Rate - (\psi - \eta \cdot E_c) \cdot Resource\ Usage \quad (25)$$

where δ and η are reweighting factors for detection rate and resource efficiency, respectively. This adjustment aligns the objective function to prioritize detection accuracy during periods of increased anomaly activity.

Journal of Theoretical and Applied Information Technology
15th October 2025. Vol.103. No.19

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

Dynamic Migration Influence Adjustment

To balance adaptation and stability, the migration influence M_a adapts according to the rate of environmental change. This dynamic migration influence is calculated as expressed in Eq.(26).

$$M_a = M_{initial} + \theta \cdot 1n(1 + E_c) \quad (26)$$

where $M_{initial}$ is the baseline migration influence, and θ controls the intensity of adaptation. This equation increases the caribou's movement adaptability during periods of heightened activity, allowing configurations to converge toward new optimized solutions.

Adaptive Feedback Loop for Continuous Adjustment

The adaptation process includes a continuous feedback loop F_b, recalibrating configurations to align with ongoing environmental conditions.

$$F_b = \sigma \cdot \left(\frac{f_{previous} - f_{current}}{f_{previous} + \epsilon} \right) \quad (27)$$

In Eq.(27), where σ is the feedback sensitivity factor, $f_{previous}$ and $f_{current}$ represent the objective function values from previous and current iterations, and ϵ ensures numerical stability. This loop enables caribou configurations to adapt responsively, aligning with optimized HIDS goals.

3.5 Selection of Elite Caribou (Best Configurations)

This step involves identifying and prioritizing configurations demonstrating optimal intrusion detection capabilities in the HIDS. By selecting elite caribou, the protocol ensures that configurations with superior performance influence the overall herd, guiding future adjustments toward optimized solutions. This process mirrors the natural selection observed in caribou herds, where only the strongest or best-suited individuals lead the way. The choice of elite caribou is determined by assessing configurations based on specific criteria such as detection accuracy, resource efficiency, and adaptability.

Elite Fitness Evaluation

The selection of elite caribou begins with evaluating the fitness of each configuration to determine those with the highest performance levels. The fitness F_e for each caribou configuration is defined by an equation balancing detection accuracy and resource usage as defined in Eq.(28).

$$F_e - \delta \cdot Detection\ Rate - k.Resource\ Usage \quad (28)$$

where δ denotes the weight given to detection accuracy, and κ is the penalty factor for resource consumption. This function ranks configurations by favoring those with high detection rates and low resource consumption, setting a benchmark for elite selection.

Threshold for Elite Selection

An elite selection threshold T_e determines which configurations are considered the best, setting a minimum fitness level that must be met. The threshold is calculated as a function of the herd's average fitness, as shown in Eq.(29).

$$T_e = \lambda \cdot \frac{\sum_{i=1}^{N} F_{e,i}}{N} \quad (29)$$

Where λ represents a threshold factor, N is the total number of caribou, and $F_{e,i}$ is the fitness of the i-th caribou. This equation selects configurations with fitness scores above the threshold, identifying them as elite.

Ranking of Elite Configurations

Once the threshold is set, elite caribou are ranked based on their fitness scores. The rank R_c of each elite configuration is determined by Eq.(30).

$$R_c = Rank(F_e, descending) \quad (30)$$

Where configurations are ordered in descending order of fitness values F_e, with the highest fitness ranked first. This ranking system establishes a hierarchy among elite caribou, where top-ranked configurations exert greater influence in the optimization process.

Influence Factor for Elite Configurations

Each elite caribou contributes an influence factor I_f, which impacts the migration patterns of other caribou in the herd. The influence factor is calculated based on each elite configuration's rank and fitness, expressed in Eq.(31).

$$I_f = \mu \cdot \frac{1}{R_c} \quad (31)$$

where μ is a scaling constant. The inverse relationship between R_c and I_f ensures that higher-ranked (more fit) configurations influence the herd's movement toward optimized solutions.

Weighted Mean Position of Elite Configurations

The weighted mean position X_w elite configurations are calculated to centralize the

herd's movement around high-performing regions. This mean position is derived from Eq.(32).

$$X_w = \frac{\sum_{j=1}^{E} I_{f,j} \cdot X_j}{\sum_{i=1}^{E} I_{f,j}} \quad (32)$$

where E denotes the number of elite configurations, $I_{f,j}$ represents the influence factor of each elite caribou, and X_j is the position of the j-th elite configuration. This weighted mean centralizes movement around the most optimized regions, anchoring future adjustments.

Elite Configuration Adjustment Threshold

An adjustment threshold A_t ensures that elite configurations maintain their status by adapting to ongoing environmental conditions. The threshold is calculated as shown in Eq.(33).

$$A_t = \theta \cdot |X_{new} - X_w| \quad (33)$$

where θ is a scaling factor, X_{new} is the adjusted position, and X_w is the weighted mean position. Elite caribou exceeding this threshold undergo additional adjustments, maintaining optimized alignment.

3.6 Migration and Path Diversification

Migration involves calculated movements based on elite configurations, while path diversification introduces variability, allowing each caribou to explore beyond familiar areas. This step strengthens adaptability, ensuring that the HIDS remains robust and responsive to evolving security threats and changes in the computational environment.

Migration Step Calculation

The migration of each caribou is directed by the previously selected elite configurations, promoting movement towards optimized solutions. The migration step M_s is calculated as Eq.(34).

$$M_s = \zeta \cdot (X_{elite} - X_{current}) \quad (34)$$

where ζ is the migration scaling factor, X_{elite} represents the position of the nearest elite configuration, and $X_{current}$ is the caribou's current position. This migration step pulls each caribou toward high-performing settings identified by elite configurations, guiding the herd collectively toward optimal parameter regions.

Path Diversification Factor

Path diversification introduces exploratory variability to prevent premature convergence on a single solution. The path diversification factor D_v is defined as Eq.(35).

$$D_v = \alpha \cdot |X_{random} - X_{current}| \quad (35)$$

where α is the diversification scaling factor, X_{random} is a randomly chosen position in the search space, and $X_{current}$ represents the caribou's current position. This factor enables the caribou to shift away from familiar paths, periodically increasing exploration.

Combined Migration and Diversification Movement

By integrating migration and diversification, each caribou's next position X_{new} is calculated through as specified in Eq.(36).

$$X_{new} = X_{current} + M_s + D_v \quad (36)$$

where M_s represents the directed migration step and D_v adds exploratory movement. This combined approach enables caribou to progress toward optimized configurations while retaining the flexibility to explore alternative paths.

Migration Decay Function

To adjust migration intensity over time, a migration decay function M_d modulates movement strength across iterations.

$$M_d = \beta \cdot e^{-\lambda \cdot t} \quad (37)$$

In Eq.(37), where β represents the initial migration intensity, λ is the decay rate, and t is the iteration number. This decay function gradually reduces migration influence, encouraging finer adjustments as caribou converge toward optimal settings.

Adaptive Diversification Radius

To dynamically adjust path diversity, an adaptive diversification radius R_d is calculated based on the performance of neighbouring configurations which is represented mathematically in Eq.(38).

$$R_d = \theta \cdot \sqrt{\frac{\sum_{k=1}^{K} (F_k - F_{mean})^2}{K}} \quad (38)$$

where θ scales the radius, F_k denotes the fitness of each neighbouring configuration, F_{mean} is the mean fitness, and K is the number of neighbours. A larger

R_d signals variability, enabling greater diversification in cases of low neighbouring fitness.

3.7 Survival of the Fittest Configuration

This step emulates natural selection, where only configurations with superior performance survive, ensuring the Host-Based Intrusion Detection System (HIDS) consistently progresses toward optimal efficiency. The survival mechanism filters out suboptimal configurations, consolidating the system's resources and focusing computational efforts on configurations that maximize intrusion detection accuracy while conserving resources.

Fitness Threshold for Surviva

The survival process begins by defining a fitness threshold F_{thresh}, which filters configurations based on their effectiveness. This threshold ensures that only configurations meeting a minimum standard are retained. The threshold is calculated as expressed mathematically in Eq.(39).

$$F_{thresh} = \chi \cdot \frac{\sum_{i=1}^{N} F_i}{N} \quad (39)$$

where χ is a scaling factor, N represents the total number of caribou configurations, and F_i is the fitness score of each configuration. By setting this benchmark, the configurations below F_{thresh} are deemed unfit for retention.

Probability of Configuration Survival

Configurations meeting the fitness threshold undergo a probabilistic assessment for survival, allowing a degree of variability in selection. The survival probability P_s for each configuration is given by Eq.(40).

$$P_s = \frac{F_i}{F_{max}} \quad (40)$$

where F_i represents the fitness of the configuration, and F_{max} is the highest fitness score among all configurations. This probability favors higher fitness values, yet allows diversity by not strictly eliminating lower-fitness configurations.

Fitness-Based Resource Allocation

Resources are allocated for configurations surviving the selection process based on their relative fitness. The resource allocation R_a for each configuration is calculated as shown in Eq.(41).

$$R_a = \frac{F_i}{\sum_{j=1}^{S} F_j} \quad (41)$$

where σ is the total available resource budget, F_i represents the fitness of the selected configuration, and S is the total number of surviving configurations. This allocation distributes resources in proportion to fitness, prioritizing highly optimized configurations.

Elimination of Weak Configurations

Configurations with fitness scores below the defined threshold are removed from the population. The elimination indicator E_i is expressed as Eq.(42).

$$E_i = \begin{cases} 1 & if\ F_i < F_{thresh} \\ 0 & otherwise \end{cases} \quad (42)$$

where F_i denotes the configuration's fitness score. An indicator value of 1 signifies elimination, effectively filtering weaker configurations from further optimization cycles.

Iterative Fitness Reinforcement

Surviving configurations undergo an iterative reinforcement to promote continued optimization in successive steps. The reinforcement factor R_f is defined as expressed in Eq.(43).

$$R_f = \rho \cdot F_i \quad (43)$$

where ρ is the reinforcement scaling factor and F_i denotes the fitness of the configuration. Reinforcing fitness enhances each configuration's potential for adaptation, ensuring persistence toward optimal HIDS performance.

3.8 Herd Communication and Information Sharing

` Through effective communication, caribou configurations share valuable information about intrusion detection and parameter tuning. This mechanism draws inspiration from herd dynamics, where individuals communicate beneficial information to adapt better to environmental changes. In HIDS optimization, such shared knowledge strengthens the configurations, improving detection capabilities and aligning the system with real-time demands.

Exchange of Fitness Information

The exchange of fitness information is crucial, allowing each caribou to understand the performance of neighbouring configurations. The shared fitness value F_s for a configuration i the average fitness of nearby configurations is calculated mathematically in Eq.(44).

$$F_s - \frac{\sum_{j \in N(i)} F_j}{|N(i)|} \quad (44)$$

where $N(i)$ represents the set of neighbouring configurations, F_j is the fitness of each neighbouring configuration j, and $|N(i)|$ denotes the total number of neighbours. This averaging process provides a reference fitness level for configurations, guiding their adjustments.

Influence of Shared Knowledge on Position Update

Each configuration adjusts its position in the parameter space using the shared fitness information. The position update $X_{updated}$ the configuration's current and average shared fitness are influenced by the configuration, calculated as shown in Eq.(45).

$$X_{updated} = X_{current} + \eta \cdot (F_s - F_i) \cdot (X_{neighbor} - X_{current}) \quad (45)$$

where η is the learning rate, F_s is the shared fitness, $F_i$represents the current configuration's fitness, and $X_{neighbor}$ denotes the position of a neighboringconfiguration. This equation directs each configuration toward positions aligned with better fitness scores, optimizing detection capabilities.

Shared Parameter Optimization

Caribou configurations benefit from shared parameter optimization, where key parameters for intrusion detection are harmonized across the herd. The optimized parameter P_o each caribou is calculated as the weighted average of parameters from neighbouring configurations.

$$P_o = \frac{\sum_{j \in N(i)} W_j \cdot P_j}{\sum_{j \in N(i)} W_j} \quad (46)$$

In Eq.(46), where $W_j$represents the weight based on each neighbour's fitness F_j, and P_j is the parameter of interest for configuration j. This weighted averaging ensures that parameters align with high-performing configurations, promoting consistency across the herd.

Adaptation of Communication Range

Herd communication includes an adaptive communication range R_c, allowing configurations to adjust their neighbourhood size based on performance. The communication range is expressed as Eq.(47).

$$R_c = \delta \cdot \left(1 - \frac{F_i}{F_{max}}\right) \quad (47)$$

where δ scales the communication distance, F_i is the fitness of the configuration, and F_{max} is the maximum fitness among configurations. This range adapts to encourage interaction with more configurations when fitness is lower, enhancing learning opportunities.

Reinforcement of High-Performance Configurations

To reinforce optimized behaviour, high-performance configurations transmit their settings more frequently. The transmission rate T_r for a configuration is defined as Eq.(48).

$$T_r = \gamma \cdot \frac{F_i}{F_{max}} \quad (48)$$

where γ is a base rate constant, F_i represents the configuration's fitness, and F_{max} is the highest fitness within the herd. This rate increases with fitness, promoting wider dissemination of effective settings across the configurations.

3.9 Continuous Monitoring and Feedback Loop

This step enables real-time responsiveness to environmental changes, ensuring the HIDS remains optimized for accurate intrusion detection and resource efficiency. Through continuous monitoring, the protocol collects performance data, while the feedback loop applies these insights to adjust configurations, promoting sustained optimization across the system.

Performance Monitoring Function

The monitoring process begins with a performance function P_m that quantifies each configuration's effectiveness. This function evaluates detection accuracy and resource usage, providing a basis for adjustments. The performance function is given as shown in Eq.(49).

$$P_m = \alpha \cdot Detection\ Accuracy - \beta \cdot Resource\ Usage \quad (49)$$

where α and β are weighting factors for accuracy and resource consumption, respectively. This calculation highlights configurations that maximize detection efficiency without excessive resource demands, establishing a performance benchmark for the feedback loop.

Error Calculation for Feedback Loop

To facilitate refinement, the feedback loop calculates an error E_f based on the deviation between desired and actual performance metrics. The error function is defined as Eq.(50).

$$E_f = P_{desired} - P_m \quad (50)$$

where $P_{desired}$ represents the target performance, and P_m is the monitored performance. This error quantifies the extent of adjustment needed to align each configuration with optimized performance standards.

Adjustment Rate Based on Error Feedback

The feedback loop applies an adjustment rate A_r proportional to the error, modifying configuration parameters accordingly. The adjustment rate is expressed as Eq.(51).

$$A_r = \gamma \cdot E_f \quad (51)$$

where γ is a scaling factor that controls the magnitude of changes based on E_f. A larger error prompts greater adjustments, enabling rapid adaptation to performance gaps.

Updated Configuration Application

Once the adjustment rate is determined, the configuration update $X_{updated}$ is applied to modify parameter values within HIDS.

$$X_{updated} = X_{current} + A_r \quad (52)$$

In Eq.(52), where $X_{current}$ is the existing configuration, and A_r represents the adjustment rate derived from the feedback loop. This update ensures that each configuration continuously aligns with performance goals, fostering ongoing optimization.

3.10 Fine-tuning through Local Adjustments

Local adjustments refine parameter values, ensuring that high-performing configurations are further optimized. This step employs incremental updates, allowing configurations to achieve optimal detection accuracy and resource utilization based on specific conditions within the HIDS environment.

Calculation of Adjustment Gradient

The fine-tuning process begins with calculating an adjustment gradient. G_a for each configuration to assess the direction and magnitude of improvement needed. This gradient is computed as depicted in Eq.(53).

$$G_a = \frac{\partial P_m}{\partial X} \quad (53)$$

where P_m is the performance metric function, and X represents the configuration parameters. The gradient identifies how changes in each parameter impact performance, guiding local adjustments to improve HIDS effectiveness.

Local Step Size for Parameter Adjustment

Using the adjustment gradient, a local step size S_l is calculated to control the extent of parameter modification, defined as Eq.(54).

$$S_l = \eta \cdot G_a \quad (54)$$

where η is a learning rate that moderates the adjustment intensity. This step size ensures that incremental parameter changes promote stability and prevent overshooting, which can lead to suboptimal performance.

Updated Configuration with Fine-Tuning

After calculating the step size, each parameter within the configuration is fine-tuned, resulting in the updated configuration $X_{fine-tuned}$ which is represented mathematically in Eq.(55).

$$X_{fine-tuned} = X_{current} + S_l \quad (55)$$

where $X_{current}$ represents the existing parameter values, and S_l adds the calculated step for fine-tuning. This update aligns each configuration with incremental enhancements, refining settings that contribute positively to intrusion detection performance.

Convergence Check for Local Adjustments

To ensure stability, a convergence check C_v assesses whether further fine-tuning is necessary. This check is calculated by evaluating the difference between consecutive updates as expressed in Eq.(56).

$$C_v = |X_{fine-tuned} - X_{current}| \quad (56)$$

If C_v falls below a specified tolerance level, no additional adjustments are made. This check ensures that configurations remain focused on precision without redundant modifications.

3.11 Convergence to Optimal Herd Configuration

This phase ensures that all configurations reach a state of optimized performance, balancing detection accuracy, resource allocation, and stability across the herd. Convergence involves identifying configurations that meet the optimized

criteria, and finalizing the most effective herd configuration for sustained HIDS efficiency.

Mean Fitness for Herd Convergence

The convergence process begins by calculating the mean fitness F_{mean} of the herd, which serves as a benchmark for final alignment. This average fitness is computed as expressed in Eq.(57).

$$F_{mean} = \frac{\sum_{i=1}^{N} F_i}{N} \quad (57)$$

where N represents the total number of configurations, and F_i is the fitness of each configuration. The mean fitness provides a standard that reflects the collective performance level across configurations.

Fitness Deviation for Stability Check

To assess convergence, a fitness deviation D_f is calculated to measure the variability among configurations. This deviation is given by Eq.(58).

$$D_f = \sqrt{\frac{\sum_{i=1}^{N}(F_i - F_{mean})^2}{N}} \quad (58)$$

where F_i denotes the fitness of each configuration, and F_{mean} is the mean fitness. A low D_f indicates minimal variation, suggesting that configurations are reaching consistent levels of performance.

Position Adjustment for Final Convergence

Each configuration's position X_{final} is adjusted to align with the mean position of high-fitness configurations, calculated as Eq.(59).

$$X_{final} = X_{mean} + \alpha \cdot (F_{mean} - F_i) \quad (59)$$

where X_{mean} is the mean position of high-fitness configurations, F_{mean} is the mean fitness, and α represents the adjustment scaling factor. This step refines each configuration's position to enhance overall convergence.

Convergence Indicator for Optimal State

A convergence indicator C_{opt} assesses whether the herd has achieved the optimal state. This indicator is defined as Eq.(60).

$$C_{opt} = |F_{mean} - F_{target}| \quad (60)$$

where F_{target} represents the predefined optimal fitness target. A minimal C_{opt} confirms that the herd has reached the desired performance level, signaling final convergence.

4. DATASET DESCRIPTION

The "Network Intrusion Detection" dataset, available on Kaggle, comprises 22,544 instances and 41 features, serving as a benchmark for evaluating intrusion detection systems. This dataset includes diverse simulated intrusions within a military network environment, providing a comprehensive foundation for analyzing various attack types. The features encompass various network traffic attributes, such as protocol type, service, and flag, along with continuous features like duration and byte counts. The dataset is instrumental in training and testing machine learning models to identify malicious activities within network traffic. Researchers and practitioners utilize this dataset to develop and benchmark algorithms for detecting anomalies and enhancing cybersecurity measures. Its structured format and detailed feature set facilitate the application of various analytical techniques, contributing to advancements in network security. The comprehensive dataset makes it a valuable resource for improving intrusion detection methodologies.

5. RESULTS AND DISCUSSIONS

Results and discussion analyze the performance of classification models by interpreting evaluation metrics. The results validate the effectiveness of each model, while the discussion highlights strengths, limitations, and comparative insights. The evaluation provides a clear understanding of detection accuracy, precision, recall, and overall classification efficiency, ensuring an informed assessment of security models in network intrusion detection. Classification algorithms for network intrusion detection are assessed using CL-AC (Classification Accuracy) and F-MSR (F-Measure) metrics. Classification Accuracy (CL-AC) measures the proportion of correctly classified instances, providing an overall performance assessment. F-Measure (F-MSR) represents the harmonic mean of precision and recall, ensuring a balanced evaluation of classification effectiveness, particularly in imbalanced datasets.

Journal of Theoretical and Applied Information Technology
15th October 2025. Vol.103. No.19

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

Fig 1. illustrates the CL-AC and F-MSR; the DI-NIDS model achieves 63.090% CL-AC and 63.071% F-MSR, indicating moderate performance but limited effectiveness in detecting sophisticated intrusions. DMAN improves detection rates, reaching 66.397% CL-AC and 67.327% F-MSR, demonstrating better classification capabilities. The Adaptive Caribou Defense Protocol (ACDP) significantly outperforms both models, achieving 80.232% CL-AC and 80.196% F-MSR, indicating a substantial improvement in detection accuracy and precision. The results highlight ACDP's ability to enhance intrusion detection efficiency, minimize false positives, and strengthen network security. The optimized approach ensures better adaptability in identifying evolving cyber threats.

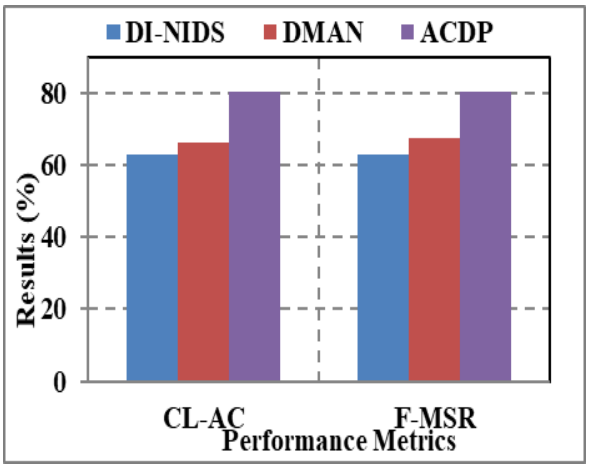
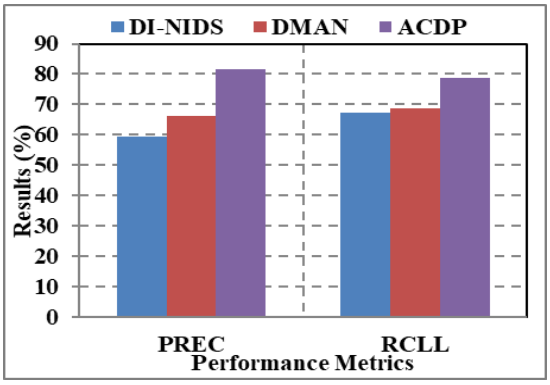


Fig 1. CL-AC and F-MSR

The performance evaluation of classification algorithms in network intrusion detection is measured using Precision (PREC) and Recall (RCLL) metrics. Precision (PREC) quantifies the proportion of correctly identified intrusions among all instances classified as intrusions, indicating the algorithm's ability to reduce false positives. Recall (RCLL) represents the proportion of correctly detected intrusions out of all actual intrusions, assessing the model's effectiveness in minimizing false negatives. Fig 2. Depicts the RCLL and PREC comparison outcomes of the three protocols.

The DI-NIDS model achieves 59.285% PREC and 67.373% RCLL, reflecting moderate detection capabilities but a relatively higher false-positive rate. DMAN improves precision and recall, reaching 66.091% PREC and 68.610% RCLL, demonstrating better classification reliability. The Adaptive Caribou Defense Protocol (ACDP) significantly enhances intrusion detection, achieving 81.598% PREC and 78.840% RCLL,



showcasing superior precision and recall.

Fig 2. RCLL and PREC

The optimized ACDP model effectively balances false positives and false negatives, ensuring more reliable and adaptive threat detection. The results confirm that ACDP outperforms other models by efficiently identifying intrusions while maintaining high detection accuracy, improving network security resilience against evolving threats.

The evaluation of classification algorithms in network intrusion detection is analyzed using Attack Detection Rate (ADR), Miss Detection Rate (MDR), and Detection Rate (DR). Attack Detection Rate (ADR) quantifies the proportion of actual intrusions correctly identified by the model, ensuring effectiveness in threat recognition. Miss Detection Rate (MDR) measures the proportion of actual intrusions that were not detected, reflecting the model's susceptibility to false negatives. Detection Rate (DR) represents the model's overall accuracy in identifying intrusions across different attack types. Fig 3. Illustrates the ADR, MDR and DR of the three protocols.

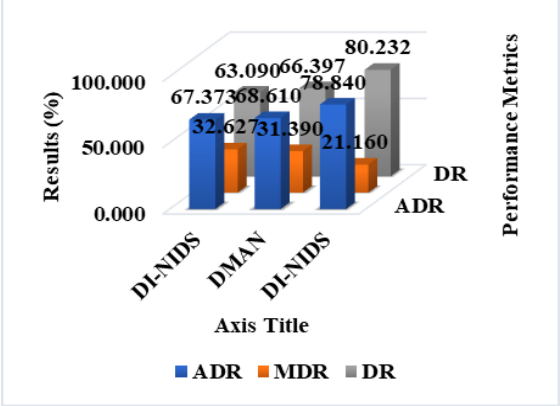


Fig 3. ADR, MDR and DR

The DI-NIDS model achieves 67.373% ADR, 32.627% MDR, and 63.090% DR, indicating moderate detection efficiency but a high miss detection rate, affecting its reliability in identifying complex threats. The DMAN model demonstrates improved detection capabilities, with 68.610% ADR, 31.390% MDR, and 66.397% DR, reflecting better intrusion recognition but presenting challenges in reducing false negatives.

The ACDP significantly enhances intrusion detection performance, achieving 78.840% ADR, 21.160% MDR, and 80.232% DR. The lower miss detection rate highlights ACDP's ability to reduce false negatives while maintaining high detection accuracy. The superior performance of ACDP confirms its effectiveness in identifying security threats with improved adaptability, reducing misclassification rates, and strengthening overall network protection. The results validate that ACDP optimizes intrusion detection by balancing high detection accuracy with minimized false negatives, ensuring a robust security framework for evolving cyber threats.

6. CONCLUSION

Intrusion detection in the IoMT requires a robust security framework to safeguard medical devices and sensitive data from cyber threats. The integration of HIDS enhances security by providing real-time monitoring, anomaly detection, and localized threat prevention. The comparative analysis of classification models, including DI-NIDS, DMAN, and ACDP, demonstrates detection performance, precision, and recall variations. While traditional deep-learning-based models such as DI-NIDS and DMAN improve detection capabilities, limitations in false-positive reduction and adaptive learning impact their effectiveness. The ACDP exhibits superior classification accuracy, achieving high detection rates while maintaining lower false negatives. Evaluating performance metrics such as accuracy, precision, recall, attack detection rate, and detection rate highlights the necessity of optimizing intrusion detection for IoMT applications. ACDP's bio-inspired optimization approach strengthens network security by ensuring an adaptive and resource-efficient detection mechanism. Reducing false positives and improving classification reliability contribute to the resilience of medical infrastructures, preventing unauthorized access and data breaches. Integrating optimized HIDS in IoMT frameworks enhances cybersecurity, ensuring real-time defense mechanisms against evolving cyber threats. Strengthening detection methodologies through advanced optimization techniques ensures continued improvements in securing interconnected medical environments.

REFERENCES:

[1]. Joshi, D. P. Kanungo, and R. K. Panigrahi, "WSN-Based Smart Landslide Monitoring Device," IEEE Trans Instrum Meas, vol. 72, pp. 1–12, 2023, doi: 10.1109/TIM.2023.3269746.

[2]. S. Rani, A. Kataria, S. Kumar, and P. Tiwari, "Federated learning for secure IoMT-applications in smart healthcare systems: A comprehensive review," Knowl Based Syst, p. 110658, 2023, doi: https://doi.org/10.1016/j.knosys.2023.110658

[3]. A. Nazari, M. Kordabadi, R. Mohammadi, and C. Lal, "EQRSRL: an energy-aware and QoS-based routing schema using reinforcement learning in IoMT," Wireless Networks, 2023, doi: 10.1007/s11276-023-03367-9.

[4]. S. Rani, A. Kataria, S. Kumar, and P. Tiwari, "Federated learning for secure IoMT-applications in smart healthcare systems: A comprehensive review," Knowl Based Syst, p. 110658, 2023, doi: https://doi.org/10.1016/j.knosys.2023.110658.

[5]. S. Singh, A. S. Nandan, G. Sikka, A. Malik, and A. Vidyarthi, "A secure energy-efficient routing protocol for disease data transmission using IoMT," Computers and Electrical Engineering, vol. 101, p. 108113, 2022, doi: https://doi.org/10.1016/j.compeleceng.2022.108113

[6]. N. Singh and A. K. Das, "Energy-efficient fuzzy data offloading for IoMT," Computer Networks, vol. 213, p. 109127, 2022, doi: https://doi.org/10.1016/j.comnet.2022.109127

[7]. J. Liang, M. Ma, and X. Tan, "GaDQN-IDS: A Novel Self-Adaptive IDS for VANETs Based on Bayesian Game Theory and Deep Reinforcement Learning," IEEE Transactions on Intelligent Transportation Systems, vol. 23, no. 8, pp. 12724–12737, 2022, doi: 10.1109/TITS.2021.3117028.

[8]. X. Liu et al., "IDSSI: Image Deturbulence with Semantic and Spatial–Temporal Information," Pattern Recognit, vol. 156, p. 110813, 2024, doi: https://doi.org/10.1016/j.patcog.2024.110813

[9]. S. Tabbassum and R. K. Pathak, "Effective data transmission through energy-efficient clustering and Fuzzy-Based IDS routing approach in WSNs," Virtual Reality &

Intelligent Hardware, vol. 6, no. 1, pp. 1–16, 2024, doi: https://doi.org/10.1016/j.vrih.2022.10.002

[10]. X. Gao, Q. Wu, J. Cai, and Q. Li, "A Fusional Intrusion Detection Method Based on the Hierarchical Filtering and Progressive Detection Model," IEEE Access, vol. 11, pp. 131409–131417, 2023, doi: 10.1109/ACCESS.2023.3335669.

[11]. A. Singh, J. Nagar, J. Amutha, and S. Sharma, "P2CA-GAM-ID: Coupling of probabilistic principal components analysis with generalised additive model to predict the k−barriers for intrusion detection," Eng Appl Artif Intell, vol. 126, p. 107137, 2023, doi: https://doi.org/10.1016/j.engappai.2023.107137

[12]. O. Friha, M. A. Ferrag, M. Benbouzid, T. Berghout, B. Kantarci, and K.-K. R. Choo, "2DF-IDS: Decentralized and differentially private federated learning-based intrusion detection system for industrial IoT," Comput Secur, vol. 127, p. 103097, 2023, doi: https://doi.org/10.1016/j.cose.2023.103097

[13]. Y. Xiang, D. Li, X. Meng, C. Dong, and G. Qin, "ResNeSt-biGRU: An Intrusion Detection Model Based on Internet of Things," Computers, Materials and Continua, vol. 79, no. 1, pp. 1005–1023, 2024, doi: https://doi.org/10.32604/cmc.2024.047143

[14]. Martins, J. S. Resende, P. R. Sousa, S. Silva, L. Antunes, and J. Gama, "Host-based IDS: A review and open issues of an anomaly detection system in IoT," Future Generation Computer Systems, vol. 133, pp. 95–113, 2022, doi: https://doi.org/10.1016/j.future.2022.03.001

[15]. F. Tlili, S. Ayed, and L. Chaari Fourati, "Exhaustive distributed intrusion detection system for UAVs attacks detection and security enforcement (E-DIDS)," Comput Secur, vol. 142, p. 103878, 2024, doi: https://doi.org/10.1016/j.cose.2024.103878

[16]. A. Paya, S. Arroni, V. García-Díaz, and A. Gómez, "Apollon: A robust defense system against Adversarial Machine Learning attacks in Intrusion Detection Systems," Comput Secur, vol. 136, p. 103546, 2024, doi: https://doi.org/10.1016/j.cose.2023.1035

[17]. A. V Turukmane and R. Devendiran, "M-MultiSVM: An efficient feature selection assisted network intrusion detection system using machine learning," Comput Secur, vol. 137, p. 103587, 2024, doi: https://doi.org/10.1016/j.cose.2023.103587

[18]. T.-P. Nguyen, J. Cho, and D. Kim, "Semi-supervised intrusion detection system for in-vehicle networks based on variational autoencoder and adversarial reinforcement learning," Knowl Based Syst, vol. 304, p. 112563, 2024, doi: https://doi.org/10.1016/j.knosys.2024.112563

[19]. M. Shoab and L. Alsbatin, "GRU Enabled Intrusion Detection System for IoT Environment with Swarm Optimization and Gaussian Random Forest Classification," Computers, Materials and Continua, vol. 81, no. 1, pp. 625–642, 2024, doi: https://doi.org/10.32604/cmc.2024.053721

[20]. J. Ramkumar, R. Vadivel, 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.

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

[22]. R. Karthikeyan and R. Vadivel, "Boosted Mutated Corona Virus Optimization Routing Protocol (BMCVORP) for Reliable Data Transmission with Efficient Energy Utilization," Wirel Pers Commun, 2024, doi: 10.1007/s11277-024-11155-7.

[23]. 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, Sep. 2024, doi: 10.22247/ijcna/2024/45.

[24]. A. Punitha, P. Ramani, E. P, and S. S, "Dynamically stabilized recurrent neural network optimized with intensified sand cat swarm optimization for intrusion detection in wireless sensor network," Comput Secur, vol. 148, p. 104094, 2025, doi: https://doi.org/10.1016/j.cose.2024.104094

[25]. D. N, J. Katiravan, S. P. D. M, and S. S. V A, "Intrusion Detection in Novel WSN-Leach

Dos Attack Dataset using Machine Learning based Boosting Algorithms," Procedia Comput Sci, vol. 230, pp. 90–99, 2023, doi: https://doi.org/10.1016/j.procs.2023.12.064

[26]. Hassini, S. Khalis, O. Habibi, M. Chemmakha, and M. Lazaar, "An end-to-end learning approach for enhancing intrusion detection in Industrial-Internet of Things," Knowl Based Syst, vol. 294, p. 111785, 2024, doi: https://doi.org/10.1016/j.knosys.2024.111785

[27]. A. B. Mohammed and L. C. Fourati, "Investigation on datasets toward intelligent intrusion detection systems for Intra and inter-UAVs communication systems," Comput Secur, vol. 150, p. 104215, 2025, doi: https://doi.org/10.1016/j.cose.2024.104215

[28]. J. Nayak, P. P. Priyadarshani, and P. B. Dash, "Improved perturbation-based hybrid firefly algorithm and long short-term memory based intelligent security model for IoT network intrusion detection," Computers and Electrical Engineering, vol. 121, p. 109926, 2025, doi: https://doi.org/10.1016/j.compeleceng.2024.109926

[29]. Meenakshi and D. Karunkuzhali, "Enhancing cyber security in WSN using optimized self-attention-based provisional variational auto-encoder generative adversarial network," Comput Stand Interfaces, vol. 88, p. 103802, 2024, doi: https://doi.org/10.1016/j.csi.2023.103802

[30]. J. F. Cevallos M., A. Rizzardi, S. Sicari, and A. Coen Porisini, "Deep Reinforcement Learning for intrusion detection in Internet of Things: Best practices, lessons learnt, and open challenges," Computer Networks, vol. 236, p. 110016, 2023, doi: https://doi.org/10.1016/j.comnet.2023.110016

[31]. J. Bhayo, S. A. Shah, S. Hameed, A. Ahmed, J. Nasir, and D. Draheim, "Towards a machine learning-based framework for DDOS attack detection in software-defined IoT (SD-IoT) networks," Eng Appl Artif Intell, vol. 123, p. 106432, 2023, doi: https://doi.org/10.1016/j.engappai.2023.106432

[32]. R. A. Abed, E. K. Hamza, and A. J. Humaidi, "A modified CNN-IDS model for enhancing the efficacy of intrusion detection system," Measurement: Sensors, vol. 35, p. 101299, 2024, doi: https://doi.org/10.1016/j.measen.2024.101299

[33]. S. Kannadhasan and R. Nagarajan, "Intrusion detection in machine learning based E-shaped structure with algorithms, strategies and applications in wireless sensor networks," Heliyon, vol. 10, no. 9, p. e30675, 2024, doi: https://doi.org/10.1016/j.heliyon.2024.e30675

[34]. R. Sahay, A. Nayyar, R. K. Shrivastava, M. Bilal, S. P. Singh, and S. Pack, "Routing attack induced anomaly detection in IoT network using RBM-LSTM," ICT Express, vol. 10, no. 3, pp. 459–464, 2024, doi: https://doi.org/10.1016/j.icte.2024.04.012

[35]. K. Ramana, A. Revathi, A. Gayathri, R. H. Jhaveri, C. V. L. Narayana, and B. N. Kumar, "WOGRU-IDS — An intelligent intrusion detection system for IoT assisted Wireless Sensor Networks," Comput Commun, vol. 196, pp. 195–206, 2022, doi: 10.1016/j.comcom.2022.10.001.

[36]. S. Z. Majidian, S. TaghipourEivazi, B. Arasteh, and A. Ghaffari, "Optimizing random forests to detect intrusion in the Internet of Things," Computers and Electrical Engineering, vol. 120, p. 109860, 2024, doi: https://doi.org/10.1016/j.compeleceng.2024.109860

[37]. S. Layeghy, M. Baktashmotlagh, and M. Portmann, "DI-NIDS: Domain invariant network intrusion detection system," Knowl Based Syst, vol. 273, p. 110626, 2023, doi: https://doi.org/10.1016/j.knosys.2023.110626

[38]. R. Harini, N. Maheswari, S. Ganapathy, and M. Sivagami, "An effective technique for detecting minority attacks in NIDS using deep learning and sampling approach," Alexandria Engineering Journal, vol. 78, pp. 469–482, 2023.

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

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

[41]. J. Ramkumar, R. Karthikeyan, and K. O. Nitish, "Securing Library Data With Blockchain Advantage," in Enhancing Security and Regulations in Libraries with

Blockchain Technology, 2024, pp. 117–138. doi: 10.4018/979-8-3693-9616-2.ch006.

[42]. P. S. Ponnukumar, N. I. Francis Xavier, and R. Jaganathan, "Stable Plithogenic Cubic Sets," J. Fuzzy Ext. Appl., vol. 6, no. 2, pp. 410–423, 2025, doi: 10.22105/jfea.2025.449408.1422.

[43]. V. Valarmathi and J. Ramkumar, "Modernizing Wildfire Management Through Deep Learning and IoT in Fire Ecology," in Machine Learning and Internet of Things in Fire Ecology, 2024, pp. 203–229. doi: 10.4018/979-8-3693-7565-5.ch0010.

[44]. Suchitra, R. Karthikeyan, J. Ramkumar, and V. Valarmathi, "Enhancing Recurrent Neural Network Performance For Latent Autoimmune Diabetes Detection (LADA) Using Exocoetidae Optimization," J. Theor. Appl. Inf. Technol., vol. 103, no. 5, pp. 1645–1667, 2025, [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-105000948603&partnerID=40&md5=66c8f111b153fed68b3d0ea9c88c411e

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

[46]. S. P. Priyadharshini, F. Nirmala Irudayam, and J. Ramkumar, "An Unique Overture of Plithogenic Cubic Overset, Underset and Offset," in Studies in Fuzziness and Soft Computing, vol. 435, 2025, pp. 139–156. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-105001675443&doi=10.1007%2F978-3-031-78505-4_7&partnerID=40&md5=e9def8c6a233de4fbf8f1549ad72027f

[47]. R. Jaganathan, K. Rajendran, and P. S. Ponnukumar, "Peregrine Falcon Optimization Routing Protocol (PFORP) for Achieving Ultra-Low Latency and Boosted Efficiency in 6G Drone Ad-Hoc Networks (DANET)," Int. J. Comput. Digit. Syst., vol. 17, no. 1, pp. 1–18, 2025, doi: 10.12785/ijcds/1571111848.

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

[49]. 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.

[50]. 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.

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

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

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

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

[55]. 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.

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

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

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

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

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

[61]. 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, Institute of Electrical and Electronics Engineers Inc., 2022. doi: 10.1109/ICACTA54488.2022.9753203.

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

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

[64]. Jayaraj, J. Ramkumar, M. Lingaraj, and B. Sureshkumar, "AFSORP: Adaptive Fish Swarm Optimization-Based Routing Protocol for Mobility Enabled Wireless Sensor Network," Int. J. Comput. Networks Appl., vol. 10, no. 1, pp. 119–129, 2023, doi: 10.22247/ijcna/2023/218516.

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

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

[67]. 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.

[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)," Int. J. Comput. Networks Appl., vol. 11, no. 5, pp. 726–748, 2024, doi: 10.22247/ijcna/2024/45.

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

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

[71]. R. Jaganathan, S. Mehta, and R. Krishan, Bio-Inspired intelligence for smart decision-

making, vol. i. 2024. doi: 10.4018/9798369352762.

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

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

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

[75]. 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," 2022 Int. Conf. Adv. Comput. Technol. Appl. ICACTA 2022, pp. 1–6, Mar. 2022, doi: 10.1109/ICACTA54488.2022.9752899.

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

[77]. L. Mani, S. Arumugam, and R. Jaganathan, "Performance Enhancement of Wireless Sensor Network Using Feisty Particle Swarm Optimization Protocol," ACM Int. Conf. Proceeding Ser., pp. 1–5, Dec. 2022, doi: 10.1145/3590837.3590907.

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