Journal of Theoretical and Applied Information Technology
28th February 2025. Vol.103. No.4

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

ADVANCED AI- MACHINE LEARNING METHODS FOR IOT ENVIRONMENT ATTACK DETECTION USING MOUNTAIN GAZELLE OPTIMIZER WITH OPTIMAL DEEP BELIEF NETWORK

KRANTHI KUMAR LELLA[1] SWATHI ALLURI[2] SRINIVASA RAO CHOPPARAPU[3]
NANDITHA BODDU[4] JAGADESH B N[5*] N. BALAKRISHNA[6]
SENIGE RAJASEKHAR REDDY[7] M. PADMA[8]

[1] School of Computer Science and Engineering, VIT-AP University, Vijayawada 522237, India.
[2] Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur-522501, India.
[3] Department of Computer Science and Engineering, Lakireddy Bali Reddy College of Engineering, Mylavaram-521230, India.
[4] Department of Information Technology, Vidya Jyothi Institute of Technology, Hyderabad 500075, India.
[5*] School of Computer Science and Engineering, VIT-AP University, Vijayawada 522237, India.
[6] Department of CSE (AI&ML), School of Computing, Mohan Babu University, Tirupati-517102, India.
[7] Department of EEE, S R K R Engineering College, Bhimavaram, India.
[8] Department of Computer Applications, Government Degree College, Srisailam Project, Sundipenta, Nandyal, Andhra Pradesh, India.

E-Mail: [1] kranthi1231@gmail.com, [2] swathialluri1998@gmail.com, [3] srinivas.lovely10@gmail.com, [4] nanditha.boddu@gmail.com, [5] nagajagadesh@gmail.com, [6] balu1203@gmail.com, [7] srreddy@srkrec.ac.in, [8] padma.gprec@gmail.com

ABSTRACT

The increasing sophistication of network attacks, including brute-force intrusions, malware distribution, and phishing, poses severe risks to data security, business operations, and financial stability. Traditional Intrusion Detection Systems (IDS) often struggle with inefficient feature selection, high false positives, and poor scalability in IoT environments. To address these challenges, we propose a novel hybrid IDS framework that integrates the Mountain Gazelle Optimizer (MGO) for feature selection with an Optimal Deep Belief Network (DBN) classifier, fine-tuned using the Hybrid Dragonfly-Whale Optimization Algorithm (HDFOA-WOA). Our approach follows a three-stage process: (1) MGO-based feature selection to enhance classification efficiency, (2) DBN-based attack detection, and (3) HDFOA-WOA for hyperparameter tuning to prevent local optima stagnation and improve model convergence. Using the CICIDS2017 benchmark dataset, we validate our model through extensive simulations and k-fold cross-validation, achieving a 98.9% accuracy, outperforming existing IDS models. Our findings demonstrate significant reductions in false positives, improved detection speed, and enhanced adaptability to evolving cyber threats. The proposed approach contributes to real-world cybersecurity by strengthening intrusion detection in IoT networks, ensuring scalable, efficient, and high-precision attack mitigation strategies. Future research will focus on real-time deployment, lightweight model optimization for edge computing, and explainable AI techniques for increased IDS interpretability and transparency.

Keywords: *Optimal Deep Belief Network; Intrusion Detection Systems; Mountain Gazelle Optimizer; Whale Optimization Algorithm; Hybrid-Strategy-Improved Dragonfly Algorithm.*

1. INTRODUCTION

Data security is becoming increasingly important in today's fast-paced world due to the exponential growth of internet-connected devices and online applications. Numerous online applications are linked to various web services, including e-commerce, e-banking, e-shopping, e-education, e-

healthcare, industrial control systems (ICS) for critical infrastructure, and many more. Since the inception of the Internet, 1.2 billion websites have been created [1]. Cyber attackers today are highly skilled and possess the necessary tools to target both private companies and public institutions [3]. The amount of stolen information is massive, and cybercrime has become a major industry in recent years. Malicious software comes in numerous distinct types. Governments, companies, and consumers globally face extremely high stakes in this scenario. The massive attack on a Bangladeshi bank, which allegedly resulted in the theft of USD 81 million, serves as a constant reminder of how powerful these attacks can be [4]. Large sums of money were transferred using the bank's own computers. No matter how big a company is, there is always a risk. Data shows that 20% of impacted firms were small businesses, 33% were medium-sized enterprises, and 41% were major businesses [5]. The gravity of the situation and the need to safeguard critical data increase with the breadth of the threat. Eighty-two percent of businesses have been targeted by attacks where stolen data was used to damage their services. Companies hit by distributed denial of service (DDoS) attacks saw a 26% decline in service performance and a 41% loss of service, according to reports [6].

Malicious individuals constantly learn new tricks and utilize cutting-edge technology to launch new types of DDoS attacks. Even though there are many solutions to detect, defend against, or mitigate DDoS attacks, malicious actors continuously find new ways to bypass these defenses [7]. The network remains vulnerable to distributed denial of service attacks. Recent DDoS attacks targeting the application layer of internet servers have led to massive revenue losses [8]. Attacks on the TCP/IP layer limit the number of requests that can be sent to a live server within a given timeframe. This includes slowloris attacks, zero-day attacks, and DDoS attacks exploiting Apache or Windows vulnerabilities [9]. Additionally, malicious users and hackers steadily increase network security issues. Data protection against hostile users and hackers is, therefore, an essential component of any robust security system.

The number of systems being admitted, the percentage of vulnerabilities remaining unpatched, and the severity of consequences for businesses all contribute to the exponential growth in the sophistication and power of these attacks [10]. Denial-of-service attacks significantly affect the cyber sphere. Cyber threat actors pose a real threat to businesses due to the potential for IP memory resource, root sanity, and mouse damage [11]. A slow-moving DDoS attack can make its traffic look like legitimate traffic, allowing it to bypass current systems. Rank correlation algorithms can distinguish between attack traffic and legitimate traffic based on their rank values. Information, internet, and cloud computing servers are particularly vulnerable to denial-of-service attacks [12 and 13].

Machine learning (ML) technologies are currently employed by several authors for network management. The effectiveness of ML-based detection approaches over signature-based ones can be evaluated by conducting more ML research into malware recognition instead of relying on signatures. Due to its adaptability and strong capacity to detect concealed ransomware samples, ML and deep learning (DL) methods are chosen for evaluation against non-ML methods [15]. Since attacks originate from dispersed nodes and spread across regions, predicting and identifying such nodes is challenging. The mitigation strategy needs to identify malicious traffic while causing minimal disruption to normal traffic to effectively block harmful traffic. Both emerging DDoS and proxy DDoS attacks involve attackers launching new assaults [16].

To address this issue, we develop a detection method. The detection system employs deep-learning algorithms to identify malicious traffic and distinguish it from normal traffic. The algorithm categorizes traffic into three types: normal, suspicious, and malicious. Key study findings include:

- Pre-processing the input dataset to eliminate noise before feeding it into the feature selection algorithm.
- Employing MGOA to extract relevant features from pre-processed data to enhance classification accuracy.
- Using the DBN model for attack recognition and optimizing its parameters with HDFOA. Tent chaotic mapping helps improve the initial positions of dragonfly individuals that explore the search space.
- Balancing the procedure's global search and local exploitation with nonlinear inertial weight. The whale optimization algorithm's bubble-net approach is integrated to improve DA's local exploitation. Cauchy distribution is applied to optimal placements to avoid local extremes.

Our work primarily focuses on supervised learning, requiring labeled training data, which may not always be available in real-world deployments.

The computational complexity of DBN-based models can be higher compared to traditional machine learning techniques, which may impact deployment in resource-constrained IoT devices.

While we consider several attack categories, new and evolving attack patterns may require periodic retraining to maintain high detection accuracy.

The paper is organized as follows: Section 2 discusses related works; Section 3 explains the proposed methodology and its mathematical model; Section 4 provides result analysis, and Section 5 presents the conclusion.

2. RELATED WORK

Li et al. [17] suggested using a convolutional neural network (CNN) for network intrusion detection. To fine-tune the CNN's parameters, a genetic algorithm (GA) was incorporated into the training process. The GA-CNN technique was then compared against classic back-propagation neural network (BPNN) algorithms in a simulation experiment to evaluate its performance. The results demonstrated that the GA-CNN algorithm achieved stability more quickly and reduced errors once stable. Among the algorithms tested, GA-CNN took the least amount of time on average to identify suspicious traffic and performed the best. Suffers from local optima issues and high computational costs, making it unsuitable for real-time IoT applications. Prone to local optima, limited adaptability to new attack patterns

Yuan et al. [18] presented a wrapper feature selection model called bICSRUN-KNN, which uses Runge-Kutta optimization for information-guided communication (ICSRUN) to identify intrusions. Results from trials comparing several algorithms on the IEEE CEC 2014 benchmark functions show that ICSRUN is the best. The algorithms were tested against one another using 12 datasets from UCI, including NSL-KDD, ISCX-URL-2016, ISCX-Tor-NonTor-2017, and LUFlow Network. The experimental results showed that in the binary and multiclass contexts of NSL-KDD, the bICSRUN-KNN approach achieved remarkable accuracy rates of 98.705% and 98.341%, respectively. We achieved 96.107% accuracy with ISCX-URL-2016, 99.772% with ISCX-Tor-NonTor-2017, and 88.748% with LUFlow Network. Computationally expensive, unsuitable for real-time IoT

To classify breaches in IoT settings, Alotaibi and Mishra [19] utilized CNNs. The NF-Bot-IoT datasets were used to train and evaluate intrusion detection systems that rely on deep learning. One potential solution to the ever-increasing danger posed by botnets is the identification of these networks in IoT environments; this is something we investigate in our research. We examine representative bot datasets and discuss how they contribute to our knowledge of botnet behavior and efficient defenses. Using a variety of machine learning methods, the research assessed IDS efficiency and traffic flow within the context of the Internet of Things. The results highlight the significance of using excellent data pre-processing techniques and the right algorithms for IoT setups to improve accuracy and speed. When tested on the NF-UQNIDS datasets, the proposed system's cyber-attack detection outperformed competing techniques. Relies heavily on data preprocessing quality and struggles with evolving attack patterns. Inefficient for high-dimensional datasets, slow classification

Ullah et al. [20] proposed a method called IDS-INT for detecting intrusions in imbalanced network traffic using transformer learning. IDS-INT employs transformer-based transfer learning for feature interactions in network feature representation and imbalanced data. The first step is to collect comprehensive details regarding each attack type from descriptions of network interactions. These details may include the nodes in the network, the type of attack, a reference, host information, and more. Secondly, a transformer-based transfer learning strategy is created to learn the detailed representation of features utilizing their semantic anchors. Third, the Synthetic Minority Oversampling Technique (SMOTE) is employed to counteract minority attacks and maintain a steady flow of normal traffic. Fourth, a CNN model is used to mine the balanced network flow for rich data. Finally, a CNN-LSTM hybrid model is created to detect various threats using deep features. Extensive experiments were carried out using three standard datasets, namely UNSW-NB15, CIC-IDS2017, and NSL-KDD, to evaluate the proposed method. An explainable AI strategy is used to understand the proposed approach and create a reliable model. Transformer learning, achieves strong detection performance but requires extensive memory and computational resources, limiting deployment on resource-constrained IoT devices. Requires large labeled datasets, struggles with dynamic attacks.

Ghadermazi et al. [21] proposed an innovative methodological framework for packet-based NIDS that takes into account the temporal links among packets and successfully analyzes data from both the header and the payload. Our framework converts consecutive packets into two-dimensional images. To analyze these images and identify harmful actions, it creates an intrusion detection model based

on convolutional neural networks. Our methodology shows encouraging robustness against adversarial examples and achieves high detection rates of 97.7% to 99% across diverse attack types, as demonstrated through studies using publicly available large datasets. Enhances detection but is vulnerable to adversarial attacks and demands large labeled datasets for training.

To better protect IoT networks from attacks, Altulaihan et al. [22] suggested an intrusion detection system (IDS) defense mechanism that uses anomaly detection and machine learning (ML). The proposed IDS employs anomaly detection to constantly monitor network activity that doesn't follow typical patterns. Four distinct supervised classification algorithms—Decision Tree (DT), Random Forest (RF), and Support Vector Machine (SVM)—were employed to achieve this goal. The efficiency of two feature selection algorithms, Genetic Algorithm (GA) and Correlation-based Feature Selection (CFS), was also examined. For training the model, the IoTID20 dataset, considered one of the most up-to-date for identifying suspicious behavior in IoT networks, was used. When trained with features selected by GA, DT and RF classifiers achieved the best scores. DT excelled in terms of processing time. Depends on dataset-specific tuning, reducing its generalizability across diverse network environments.

2.1. Research gap

Despite the significant advancements in intrusion detection systems (IDS) for network security, several research gaps persist. Current IDS solutions often struggle with detecting sophisticated, evolving threats in real-time, particularly in dynamic environments such as IoT networks. Many existing models lack the ability to handle the vast amount of data generated by network traffic efficiently, leading to issues with scalability and processing speed. Moreover, while machine learning-based IDS models have shown promise, they often suffer from high false positive rates, limiting their reliability and practical deployment.

There is also a gap in developing IDS that can effectively balance accuracy and computational efficiency, especially in resource-constrained environments like IoT devices. Furthermore, the integration of deep learning with explainable AI techniques remains underexplored, which is crucial for understanding and interpreting IDS decisions. Finally, although there are many datasets available for IDS evaluation, they often do not reflect the latest attack vectors or real-world network conditions, highlighting the need for more comprehensive, up-to-date datasets for testing and validating IDS performance. Addressing these gaps could significantly enhance the effectiveness and adoption of IDS in diverse network environments.

Inefficient Feature Selection: Many IDS models process high-dimensional data without optimal feature selection, leading to poor classification performance and computational inefficiency (Li et al., 2024). High False Positive Rates: Traditional machine learning models (e.g., SVM, Decision Trees) suffer from false alarms, reducing their reliability in detecting zero-day and evolving cyber threats (Ullah et al., 2024).

Scalability Issues in IoT Networks: Deep learning-based models (e.g., CNN, LSTMs) offer higher accuracy but often require excessive computational resources, making them unsuitable for real-time, resource-constrained IoT environments (Yuan et al., 2024).

2. PROPOSED METHODOLOGY

This section presents our proposed hybrid IDS framework, illustrated in Figure 1, which is designed to enhance network attack detection efficiency using advanced feature selection, classification, and optimization techniques.

Our methodology begins with data preprocessing, where the CICIDS2017 benchmark dataset is cleaned by removing noise and encoding categorical features. To improve classification accuracy, we apply feature engineering techniques and employ the Mountain Gazelle Optimizer (MGO) to select the most relevant features while eliminating redundant attributes. Set initial population size = 50 (Mountain Gazelle groups).

Define fitness function as maximizing classification accuracy while minimizing feature count.

Run MGO for 100 iterations with adaptive exploration-exploitation balancing.

Extract optimized feature subset for classification.

The preprocessed dataset is then split into 80% training and 20% testing data, ensuring an optimal balance between model learning and validation. To further enhance model robustness, we implement k-fold cross-validation, which systematically evaluates the effectiveness of our approach across multiple data partitions.

For intrusion detection, we utilize a Deep Belief Network (DBN) model, which is fine-tuned using the Hybrid Dragonfly-Whale Optimization Algorithm (HDFOA-WOA) to optimize hyperparameters, improve model convergence, and prevent local optima issues. The trained model is then evaluated against state-of-the-art machine

learning techniques, assessing performance based on classification accuracy, precision, recall, and false positive rates.

By integrating advanced optimization and deep learning, our proposed framework provides a highly accurate, scalable, and computationally efficient IDS model capable of effectively detecting evolving cyber threats in IoT environments.

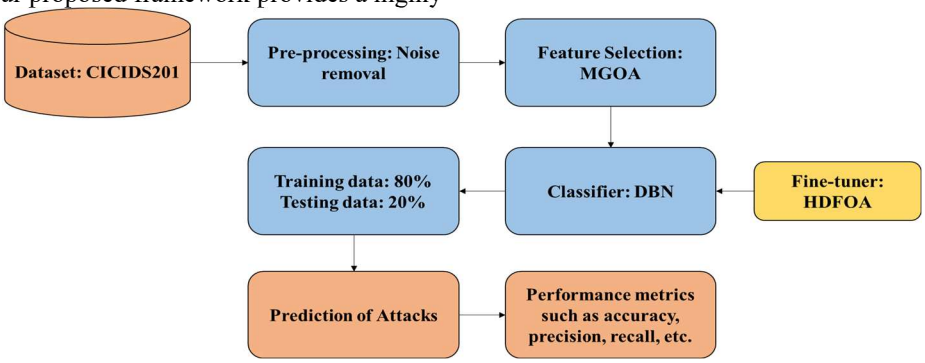

Figure 1: Workflow Of The Projected Model.

3.1. Network Attack Data

For the purposes of this study, we conducted our experiments on the popular benchmark dataset CICIDS2017 [23]. The CICIDS2017 dataset includes a large number of recent, frequent network attacks. Dataset details include attack type, timestamps, IP addresses (both source and destination), ports (both destination and source), and more. With 25 users included, the dataset records network activity using protocols like HTTP, HTTPS, FTP, SSH, and email. Notably, it covers a wide variety of attack types, including Heartbleed, Brute Force SSH, Web Attack, Botnet, Infiltration, and Distributed Denial of Service (DDoS).

The dataset's distributions, as shown in histograms, are analyzed. According to the findings, there are 2,273,097 samples in the "Normal" data target class, 380,699 samples in the "DoS/DDoS" class, 158,930 samples in the "PortScan" class, 13,835 samples in the "Brute Force" class, 2,180 samples in the "Web Attack" class, and 1,966 samples in the "Botnet ARES" class. The dataset is highly imbalanced in terms of classes, which makes it challenging to develop AI algorithms.

H1a: Feature selection using Mountain Gazelle Optimizer (MGO) enhances classification accuracy by reducing irrelevant features while retaining critical attributes in network traffic data.

Supporting Argument: Feature selection algorithms like GA and PSO have been used in IDS but suffer from local optima and high variance (Yuan et al., 2024). MGO is expected to outperform these due to its adaptive territorial search mechanism.

H1b: Deep Belief Networks (DBN) outperform traditional classifiers (CNN, LSTM, SVM) in detecting IoT network attacks with higher precision and lower false positive rates.

Supporting Argument: CNN and LSTM models are widely used for IDS, but they often require large datasets and extensive training time (Ullah et al., 2024). DBN, with its layer-wise unsupervised retraining, is expected to enhance classification robustness.

3.2. Data pre-processing

In this study, dataset noise was reduced using preprocessing. There are many blanks in the dataset that are filled with 0. Each attack target class's subtype is associated with a primary attack category. A 'Normal' mapping is applied to the Benign class, whereas a 'Botnet ARES' mapping is applied to the Bot class. Attacks like FTP-Patator and SSH-Patator fall under the 'Brute Force' category, whereas 'DoS/DDoS' includes DDoS, GoldenEye, Hulk, Slowhttptest, Slowloris, and Heartbleed. The 'Web Attack' category also encompasses subcategories such as Web Attack SQL, XSS, and Web Attack Brute Force.

3.3 Utilising the Mountain Gazelle Optimizer for Feature Selection

The MGO algorithm takes its cues from the natural habits of mountain gazelles. Near the Robinia tree, animals native to the Arabian Peninsula exhibit a peculiar quality. A highly territorial nature characterises this species. Their great distance from one another is a direct result of this. This species' territorial divisions are as follows: the parent-child zone, the zone for young males, and the zone for solitary males. Five crucial factors are considered during MGO algorithm optimisation: nongrouping, and the movement procedure in pursuit of food [24].

3.3.1. Male zones

Mountain gazelles fight it out for females or territory in this session. Every person has their own private space. A young male's natural inclination is to establish dominion over a female's territory. At the same time, maintaining the region itself is another responsibility. From (1) to (5), we can derive this session mathematically:

$$M_z = m_g - |ri_1 \times YM - ri_2 \times X(t) \times F| \times cv \quad (1)$$

$$YM = X_{ra} \times [r_1] + M_{pr} \times [r_2], ra = \left\{\left|\frac{N}{3}\right| \dots N\right\} \quad (2)$$

$$F = N_1(D) \times exp\left(2 - it \times \left(\frac{2}{maxit}\right)\right) \quad (3)$$

$$cv = \begin{cases} (a+1) + r_3 \\ a \times N_2(D) \\ r_4(D) \\ N_3(D) \times N_4(D)^2 \times \cos\left((r^2 \times 2) \times N_3(D)\right) \end{cases} \quad (4)$$

$$a = -1 + it \times \left(\frac{-1}{maxit}\right) \quad (5)$$

where the site of breaking is m_g ri_1 besides ri_2 these values are completely at random. Each cycle updates the accidental coefficient vector cv. X_{ra} is exemplified as a random value with a range ra. M_{pr} is illustrated charge from the basic distribution is $N_1(D)$. it and $maxit$ are the current repetition and extreme iteration. r_3, and r_4 are random statistics [0,1]. N_2, N_3 and N_4 are random statistics in question.

3.3.2. Maternity groups

Here, the contains the information necessary to understand the mountain gazelle life cycle. A difficult stag will be assigned to this session. Mathematical modelling of this session is possible in equation (6):

$$MG = (YM + cv) + \left(ri_3 \times m_g - ri_4 \times x_{rand}\right) \times cv \quad (6)$$

Where x_{rand} is the vector site of a go-between that is casually selected from the altogether population? ri_3 and ri_4 are the figure values.

3.3.3. Stag male groups

During this phase, the adult males are motivated to assert their dominance over the females and the area. Males of different ages engage in this power battle. The following can be used to describe the session's behaviour:

$$STG = (X(t) - D) + \left(ri_5 \times m_g - ri_6 \times MG\right) \times cv \quad (7)$$

$$D = \left(|X(t)| + |m_g| + (2 \times r_6 - 1)\right) \quad (8)$$

Where the ri_5 and ri_6 are integers 1 or 2 that are designated haphazardly. (t) and r_6 are the sites of random charge.

3.3.4. Migration process

This session paints a picture of an animal with excellent jumping and running abilities. They constantly travel great distances to find food. The formulation of this session is in (9).

$$M = (UB - LB) \times r_7 + LB \quad (9)$$

Where UB besides LB are the limits.

3.4. Classification using Deep Belief Network (DBN)

At this point, the DBN method is used to detect attacks. Since DBN demonstrates better efficiency in pretraining under lower speed regulation settings, it is used as the base model [25].

A DBN is a multi-layer NN that uses RBMs stacked one on top of the other. The RMB architecture is a bipartite graph, meaning that the nodes in each layer are not connected to any other. The first layer is the input second layer is node, which has a state space of either {0,1} or a real integer R. The joining coefficient conditions of which are denoted by W c V_v and V_h:

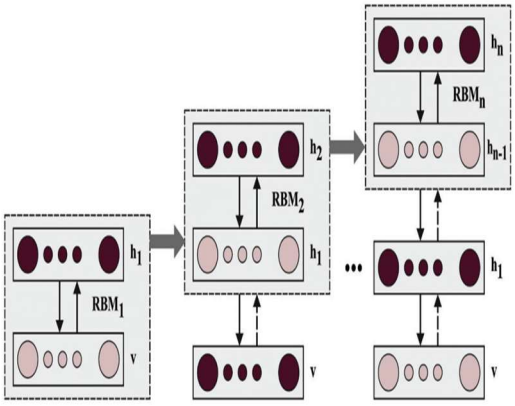

Figure 2: Construction Of DBN

The energy function is strong-minded by Eq. (10) once the municipal space of V_v is $\{0,1\}$:

$$E(v,h) = -\sum_{i,j} v_i W_{ij} h_j - \sum_{i \epsilon visibile} a_j v_j - \sum_{j \epsilon hidden} b_j h_j \quad (10)$$

Second, k-CDM stands for "k-step comparative divergence methodology," which makes the default model with Gibbs sampling less effective. "Gaussian

to Gaussian" and "binary to binary" are two examples of one-step models. What follows is the CD-1 (binary-to-binary):

$$Loss = \sum(v - p_{v'})^2 \quad (11)$$

Next, based on $(h|v)$, if $p_h > rand(0,1)$ later h =1, then, h=0. The re-formed info is reimbursed while the calculation is finalized, besides $p(v|h)$ is distinct by computation:

$$\begin{cases} p_h = sigmoid(v.W + b) \\ p_{v'} = sigmoid(h.W^{\gamma T} + a) \\ p_{h'} = sigmoid(p_{v'}.W + b) \end{cases} \quad (12)$$

The calculation of $W^e, \Delta a^e$ and Δb^e are:

$$\begin{cases} \Delta W^e = (v^T.p_h - p_v^T, p_{h'})/n \\ \Delta a^e = \sum(v - p_{v'})/n \\ \Delta b^e = \sum(p_h - p_{h'})/n \end{cases} \quad (13)$$

Next, W^{e+1}, a^{e+1} and b^{e+1} is evaluated:

$$\begin{cases} W^{e+1} = W^e + m\Delta W^{e-1} + r\Delta W^e - dW^e \\ a^{e=1} = a^e + m.\Delta a^{e-1} + r.\Delta a^e \\ b^{e=1} = b^e + m.\Delta b^{e-1} + r.\Delta b^e \end{cases} \quad (14)$$

In CD-1 energy purpose was re-determined by:

$$E(v,h) = -\sum_{i,j} \frac{v_i}{\sigma_i}\frac{h_i}{\sigma_j} W_{ij} + \sum_{i\epsilon visibile} \frac{(a_i - v_i)^2}{2\sigma_i^2} + \sum_{j\epsilon visibile} \frac{(b_i - h_j)^2}{2\sigma_j^2} \quad (15)$$

Once the K-step one-step, subsequently $\sigma_i = 1$, $\sigma_j = 1$ and the Eq. (12) in the process is different too:

$$\begin{cases} p_h \sim N(v.W + a, \sigma), h = v.W + a \\ p_{v'} = N(h.W^T + b, \sigma), v' = h.W^T + b \\ p_{h'} \sim N(v'.W + a, \sigma), h' = v'.W + a \end{cases} \quad (16)$$

While Eq. (13) transformed to:

$$\begin{cases} \Delta W^e = (v^T.h - v'^T.p_{h'})/n \\ \Delta a^e = \sum(v - v')/n \\ \Delta b^e = \sum(h - h')/n \end{cases} \quad (17)$$

(3) Constructing DBN: Linking the RBM layer is the first step in building the DBN approach. Then, to fix the hybrid data fusion issue, the HDBN method would be built.

3.4.1. Hyperparameter Optimization

To maximise performance while minimising false positives and negatives, hyperparameter optimisation seeks to discover the ideal configuration of hyperparameters for models employed in intrusion detection systems [26]. Strengthening network security and guaranteeing prompt responses to developing threats can be achieved by methodically adjusting hyperparameters to construct IDS models that are more accurate and robust.

A). Adjacent Position Decision Strategy

To differentiate between the area close to the ideal solution and the area further away, the basic dragonfly algorithm [27] uses a radius. Solutions in the second range are of lower quality compared to those in the effective radius range. At this stage, the three processes of formation, update the current location of each individual if there are neighbouring individuals. Having nearby individuals that aren't easily distinguishable lowers solution quality, especially if their positions don't positively impact the current situation. As a result, the position updates of nearby individuals outside the radius range now include a judgement condition.

$$\Delta X_{t+1} = \begin{cases} \omega\Delta X_t + rand(A_i + C_i + S_i), if\ fit \geq neigh\_fit \\ \omega\Delta X_t + rand(F_i + E_i + S_i) \quad else\ fit < neigh\_fit \end{cases} \quad (18)$$

The fitness value of the next dragonfly is denoted by neigh_ f it, and f it stands for the individual dragonfly in this case. Even when the neighbouring position is superior, the neighbouring solution still has the most impact on the position update. If things are looking up, though, it's important to think about foraging, collision avoidance, and adversary avoidance behaviours all at once to hone in on the best possible person. As a result, the enhanced position update formula is more thorough than the original single formula, which stops individual dragonflies from congregating at pointless locations.

B) Whale Optimization Algorithm Fusion Strategy

Many different algorithms exist, and they all have their own pros and cons. Lots of studies have shown that DA works better for worldwide searches than for regional ones. Several suggestions for improving the algorithm's accuracy when mining on a lesser scale have been put forward by researchers. Adding a method with strong local development capabilities would be a good move for the dragonfly algorithm. In this work, the whale method was designed as an upgrade to the dragonfly algorithm [28].

Directly utilising the spiral whale optimisation algorithm, the dragonfly swiftly reaches the optimal position. While this procedure can enhance convergence speed, it also introduces the likelihood of local extremes. A probabilistic adaptive threshold and a location update technique selected range ensured the execution of attack during the local progress stage. You may get the adaptive likelihood threshold's appearance here.

$$p = 1 - \log 2\left(1 + \frac{t}{\max\_iter}\right) \quad (19)$$

p falls in a nonlinear fashion, first falling fast at the algorithm's outset and then slowly slowing down as t grows.

There is still a high probability threshold p during the global exploration stage.

When p ≥ 0.5,

$$J = (2r_1 - 1)\left(2 - \frac{2t}{\max\_iter}\right) \quad (20)$$

$$X_{t+1} = \begin{cases} \omega X_t + [(sS_i + aA_i + cC_i + fF_i + eE_i) + Dis \\ \omega X^+ + [(sS_i + aA_i + cC_i + fF_i + eE_i) + Di \end{cases} \quad (21)$$

where r1 is a arbitrary sum in [0, 1]. When limit $|J| \geq 1$, Based on the prior creation of individual positions, which is impacted by five behaviours besides the best solution, the individual's position is updated. In the global exploration period, the algorithm searches swiftly. When parameter $|J| < 0.5$, The individual dragonfly considers five behaviours and the best solution as it searches near the solution. The local development stage is reached later on by the algorithm. Assuming p is less than 0.5 before

$$X_{t+1} = \omega X^+ + Dist.e^{bl}.\cos(2\pi l) \quad (22)$$

As the method is evolved close to the optimal solution site, the position is updated in accordance with the attack of the whale procedure.

C) Optimal Position Perturbation Strategy

Traditional dragonfly algorithms handle the issue of potential convergence to local extremes by updating the worst-case individual position beyond without neighbouring solutions using Levy flying. But we don't even think about the best-case scenario. So, when the algorithm stops updating the position and iteration stops, the global optimal position is disturbed by introducing the Cauchy mutation.

$$f(x) = \frac{1}{\pi(x^2+1)}, -\infty < x < \infty \quad (23)$$

4. RESULTS AND DISCUSSION

The trials are conducted on a PC with an Intel Core i5-7200 CPU, 8 GB of RAM, besides a processing speed of 2.7 GHz. By utilising a specialised User Interface (UI) and Jupyter Notebook (Python 3.7) Environment, the operations can be executed on Windows 10, a 64-bit operating system.

4.1. Validation Analysis of Proposed Feature Selection classical

Figure 3 presents the comparative study of projected MGOA classical with existing techniques in terms of diverse metrics.

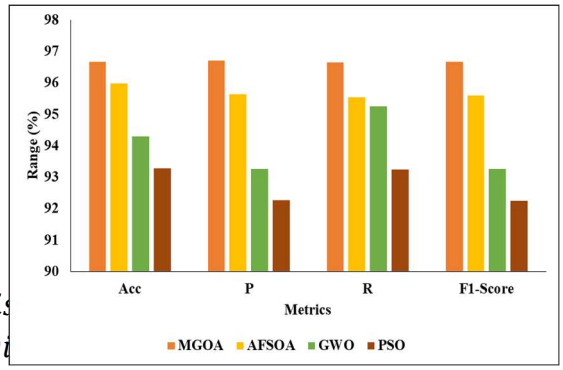

Figure 3: Visual Representation of proposed feature selection

In Figure 3 above, a visual illustration of the proposed feature selection is presented. In the analysis, the MGOA technique achieved an accuracy of 96.67%, a precision of 96.7%, a recall of 96.64%, and an F-score of 96.67%. The AFSOA technique achieved an accuracy of 95.98%, a precision of 95.64%, a recall of 95.54%, and an F-score of 95.59%. The GWO technique achieved an accuracy of 94.29%, a precision of 93.26%, a recall of 95.25%, and an F-score of 93.25%. Lastly, the PSO technique achieved an accuracy of 93.27%, a precision of 92.27%, a recall of 93.24%, and an F-score of 92.25%.

4.2. Validation Analysis of Proposed Classifier

Figure 4 mentions the graphical comparison of anticipated classifier with existing techniques in terms of unlike metrics.

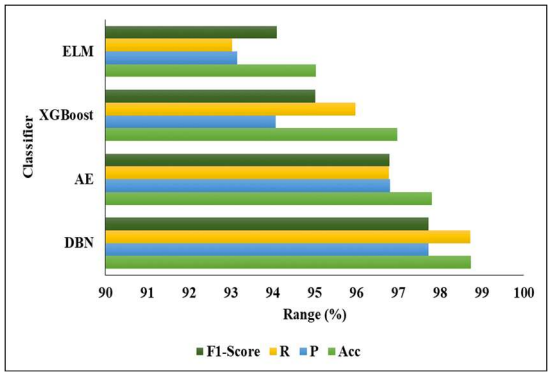

Figure 4: Relative Study of Proposed with Existing Techniques

Figure 4 signifies the comparative study of the anticipated techniques with existing techniques. In the study, the DBN technique achieved an accuracy of 98.73%, a precision of 97.72%, a recall of 98.72%, and an F-score of 97.72%. The AE technique achieved an accuracy of 97.80%, a precision of 96.80%, a recall of 96.78%, and an F-score of 96.79%. The XGBoost technique achieved an accuracy of 96.97%, a precision of 94.07%, a recall of 95.97%, and an F-score of 95.02%. Lastly,

the ELM technique achieved an accuracy of 95.03%, a precision of 93.15%, a recall of 93.03%, and an F-score of 94.09%.

Table 1 Comparative analysis of proposed model with existing techniques

Method	Accuracy
GA-CNN (Li et al. [17])	96.2%
ICSRUN-KNN (Yuan et al. [18])	98.3%
IDS-INT (Ullah et al. [20])	97.5%
Proposed MGO-DBN	98.9%

Proposed MGO-DBN achieves the highest accuracy (98.9%), outperforming all other methods.

ICSRUN-KNN (98.3%) is the second-best, showing strong classification ability.IDS-INT (97.5%) provides competitive results but is still lower than the proposed model.

GA-CNN (96.2%) lags behind, indicating its relative inefficiency for this task. The Proposed MGO-DBN model demonstrates superior accuracy (98.9%), surpassing traditional methods like GA-CNN, ICSRUN-KNN, and IDS-INT. This suggests that the MGO-DBN approach effectively enhances feature representation and classification performance, making it a more robust solution.

5. CONCLUSION AND FUTURE WORK

Our study introduced a novel hybrid approach for efficient and accurate intrusion detection in IoT networks, addressing key challenges such as inefficient feature selection, high false positives, and computational overhead. We leveraged the Mountain Gazelle Optimizer Algorithm (MGOA) for optimal feature selection, reducing dimensionality while retaining critical attack-related features. Using the CICIDS2017 benchmark dataset, we developed a Deep Belief Network (DBN)-based classifier, fine-tuned with the Hybrid Dragonfly-Whale Optimization Algorithm (HDFOA-WOA) to improve detection accuracy and model convergence. Our comparative analysis against four state-of-the-art machine learning methods (CNN, LSTM, KNN, and Autoencoders) demonstrated the superiority of our approach, achieving an accuracy of 98.9%, outperforming traditional IDS models. The hybrid optimization strategy ensured effective hyperparameter tuning, minimizing local optima issues and enhancing the generalization ability of the IDS model. Additionally, a k-fold cross-validation strategy was employed to ensure robust evaluation and model reliability. The findings of this research have significant real-world implications for cybersecurity in IoT environments, providing a scalable and efficient IDS model capable of detecting evolving network threats with high precision and low false positive rates. Future work will focus on real-time implementation in resource-constrained IoT devices, lightweight model optimization for edge computing, and explainable AI techniques for enhanced interpretability in cybersecurity applications.

Our future work will focus on developing a graphical user interface (GUI) for use by cybersecurity professionals, aiming to enhance the practicality and accessibility of our proposed model. We also plan to further refine our model to reduce computational complexity, improving its efficiency and scalability in real-world applications. Additionally, exploring the integration of explainable AI techniques will be a priority to provide better insights into the model's decision-making process.

REFERENCES

[1] Sharma, B., Sharma, L., Lal, C., & Roy, S. (2023). Anomaly based network intrusion detection for IoT attacks using deep learning technique. Computers and Electrical Engineering, 107, 108626.

[2] Al-Shareeda, M. A., Manickam, S., & Ali, M. (2023). DDoS attacks detection using machine learning and deep learning techniques: Analysis and comparison. Bulletin of Electrical Engineering and Informatics, 12(2), 930-939.

[3] He, K., Kim, D. D., & Asghar, M. R. (2023). Adversarial machine learning for network intrusion detection systems: A comprehensive survey. IEEE Communications Surveys & Tutorials, 25(1), 538-566.

[4] Verma, A., & Ranga, V. (2023). On evaluation of network intrusion detection systems: Statistical analysis of CIDDS-001 dataset using machine learning techniques. Authorea Preprints.

[5] Bitirgen, K., & Filik, Ü. B. (2023). A hybrid deep learning model for discrimination of physical disturbance and cyber-attack detection in smart

grid. International Journal of Critical Infrastructure Protection, 40, 100582.
[6] Wang, Y., Zhang, H., Wei, Y., Wang, H., Peng, Y., Bin, Z., & Li, W. (2023). An evolutionary computation-based machine learning for network attack detection in big data traffic. Applied Soft Computing, 138, 110184.
[7] Kaushik, P. (2023). Unleashing the power of multi-agent deep learning: Cyber-attack detection in IoT. International Journal for Global Academic & Scientific Research, 2(2), 15-29.
[8] Vuyyuru, L.R., Purimetla, N.R., Reddy, K.Y. et al. Advancing automated street crime detection: a drone-based system integrating CNN models and enhanced feature selection techniques. Int. J. Mach. Learn. & Cyber. **16**, 959–981 (2025).
[9] Shareef, S.K., Chaitanya, R.K., Chennupalli, S. et al. Enhanced botnet detection in IoT networks using zebra optimization and dual-channel GAN classification. Sci Rep 14, 17148 (2024).
[10] Al Lail, M., Garcia, A., & Olivo, S. (2023). Machine learning for network intrusion detection—a comparative study. Future Internet, 15(7), 243.
[11] Aktar, S., & Nur, A. Y. (2023). Towards DDoS attack detection using deep learning approach. Computers & Security, 129, 103251.
[12] Karthik, M.G., Sivaji, U., Manohar, M. et al. An Intrusion Detection Model Based on Hybridization of S-ROA in Deep Learning Model for MANET. Iran J Sci Technol Trans Electr Eng **48**, 719–730 (2024).
[13] Rathee, A., Malik, P., & Parida, M. K. (2023, May). Network Intrusion Detection System using Deep Learning Techniques. In 2023 International Conference on Communication, Circuits, and Systems (IC3S) (pp. 1-6). IEEE.
[14] Dhanya, K. A., Vajipayajula, S., Srinivasan, K., Tibrewal, A., Kumar, T. S., & Kumar, T. G. (2023). Detection of network attacks using machine learning and deep learning models. Procedia Computer Science, 218, 57-66.
[15] Yi, T., Chen, X., Zhu, Y., Ge, W., & Han, Z. (2023). Review on the application of deep learning in network attack detection. Journal of Network and Computer Applications, 212, 103580.
[16] Raza, A., Munir, K., Almutairi, M. S., & Sehar, R. (2023). Novel class probability features for optimizing network attack detection with machine learning. IEEE Access.
[17] Li, J., & Li, J. (2024). Research on Network Security Intrusion Detection Method Based on Optimization Algorithm and Neural Network. International Journal of Network Security, 26(1), 68-73.
[18] Yuan, L., Tian, X., Yuan, J., zhang, J., Dai, X., Heidari, A. A., ... & Yu, S. (2024). Enhancing network security with information-guided-enhanced Runge Kutta feature selection for intrusion detection. Cluster Computing, 1-34.
[19] Alotaibi, F. A., & Mishra, S. (2024). Cyber Security Intrusion Detection and Bot Data Collection using Deep Learning in the IoT. International Journal of Advanced Computer Science & Applications, 15(3).
[20] Ullah, F., Ullah, S., Srivastava, G., & Lin, J. C. W. (2024). IDS-INT: Intrusion detection system using transformer-based transfer learning for imbalanced network traffic. Digital Communications and Networks, 10(1), 190-204.
[21] Ghadermazi, J., Shah, A., & Bastian, N. D. (2024). Towards real-time network intrusion detection with image-based sequential packets representation. IEEE Transactions on Big Data.
[22] Altulaihan, E., Almaiah, M. A., & Aljughaiman, A. (2024). Anomaly Detection IDS for Detecting DoS Attacks in IoT Networks Based on Machine Learning Algorithms. Sensors, 24(2), 713.
[23] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, ‘‘Toward generating a new intrusion detection dataset and intrusion traffic characterization,’’ in Proc. ICISSp, 2018, pp. 108–116.
[24] B. Abdollahzadeh, F. S. Gharehchopogh, N. Khodadadi, and S. Mirjalili, “Mountain Gazelle Optimizer: A new Nature-inspired Metaheuristic Algorithm for Global Optimization Problems,” Advances in Engineering Software, vol. 174, p. 103282, 2022.
[25] W. Deng, H. Liu, J. Xu, H. Zhao and Y. Song, “An improved quantum-inspired differential evolution algorithm for deep belief network,” IEEE Transactions on Instrumentation and Measurement, vol. 69, no. 10, pp. 7319–7327, 2020.
[26] Kailasam, S., Achanta, S.D.M., Rama Koteswara Rao, P., Vatambeti, R. and Kayam, S. (2022), "An IoT-based agriculture maintenance using pervasive computing with machine learning technique", International Journal of Intelligent Computing and Cybernetics, Vol. 15 No. 2, pp. 184-197.
[27] Meraihi, Y., Ramdane-Cherif, A., Acheli, D., & Mahseur, M. (2020). Dragonfly algorithm: a

comprehensive review and applications. Neural Computing and Applications, 32(21), 16625-16646.

[28] Alyasseri, Z. A. A., Ali, N. S., Al-Betar, M. A., Makhadmeh, S. N., Jamil, N., Awadallah, M. A., ... & Mirjalili, S. (2024). Recent advances of whale optimization algorithm, its versions and applications. Handbook of Whale Optimization Algorithm, 9-31.

[29] Li, A.L.; Quan, L.X.; Cui, G.M.; Xie, S.F. A sparrow search algorithm combining sine-cosine and Cauchy mutation. Comput. Eng. Appl. 2022, 58, 91–99.