IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Adaptive Intrusion Detection For IOT Networks Using Bio-Inspired Optimization To Mitigate Ddos Attacks

[1]Tamilarasan G, [2]Shri Vishva P, [3]Dr.P.Senthil Pandian, [4]Dr.J.Hemalatha, [5]Mr.C.PiravinKumar

[1,2]UG Student, Department of Computer Science and Engineering, AAA College of Engg & Tech, Amathur, Sivakasi

[3]Associate Professor, Department of Computer Science and Engineering, AAA College of Engg & Tech Amathur, Sivakasi.

[4]Professor & Head, Department of Computer Science and Engineering, AAA College of Engg & Tech Amathur, Sivakasi.

[5]Assistant Professor, Department of Computer Science and Engineering, AAA College of Engg & Tech Amathur, Sivakasi.

Abstract: The rapid expansion of the Internet of Things (IoT) has enabled transformative applications across various sectors such as healthcare, smart cities, and industrial automation. However, this surge in connectivity has simultaneously exposed IoT networks to heightened cybersecurity threats, particularly Distributed Denial of Service (DDoS) attacks. Due to limited processing and security capabilities, IoT devices are easily compromised, making traditional Intrusion Detection Systems (IDS) inadequate in such environments. This study introduces an adaptive and lightweight IDS framework that utilizes a hybrid ensemble of bio-inspired optimization techniques—Spotted Hyena Optimizer (SHO), Parrot Optimizer (PO), and Grey Wolf Optimizer (GWO)—for feature selection. By embedding a self-attention mechanism, the system dynamically identifies key features that enhance detection accuracy while minimizing computational costs. The selected features are used to train machine learning classifiers including Decision Tree, SVM, Random Forest, XGBoost, and shallow Neural Networks. Evaluated on the UNSW-NB15 dataset, the proposed model demonstrates high performance with reduced false positives and latency, offering a scalable and real-time solution for DDoS mitigation in IoT ecosystems.

Keywords: Internet of Things (IoT), Distributed Denial of Service (DDoS), Intrusion Detection System (IDS), Bio-Inspired Optimization, Feature Selection, Machine Learning.

1.Introduction:

The Internet of Things (IoT) has revolutionized the way physical systems interface with the digital realm by enabling seamless communication among interconnected devices. Applications in areas such as patient monitoring, smart infrastructure, and automated manufacturing have demonstrated the potential of IoT to enhance operational efficiency and decision-making. However, the increasing reliance on IoT also introduces new cybersecurity vulnerabilities. Devices with constrained computational power and minimal built-in security are particularly vulnerable to Distributed Denial of Service (DDoS) attacks, which exploit these limitations to disrupt network functionality.

Conventional DDoS prevention mechanisms often struggle to perform effectively in IoT environments due to the heterogeneous and resource-constrained nature of the devices involved. Moreover, static detection models fail to adapt to dynamic traffic patterns and emerging threats. These limitations call for more intelligent, adaptive, and lightweight solutions that can operate efficiently within the unique constraints of IoT systems. One promising direction is the integration of bio-inspired optimization techniques, which mimic natural behaviors to solve complex computational problems, particularly in high-dimensional feature spaces encountered in network security.

Traditional DDoS mitigation techniques are often ineffective in IoT environments due to device heterogeneity, constrained resources, and highly dynamic conditions. Therefore, there is an urgent need for lightweight and adaptive security solutions tailored specifically for IoT systems. One promising direction involves the use of bio-inspired optimization algorithms—computational techniques that emulate natural processes and behaviors observed in biological systems. Algorithms such as Genetic Algorithms (GA), Particle Swarm Optimization (PSO), and Ant Colony Optimization (ACO) have already demonstrated effectiveness in tasks like anomaly detection, traffic classification, and resource allocation within cybersecurity contexts.

A critical component in enhancing the accuracy and efficiency of Intrusion Detection Systems (IDS) is **feature selection**, which involves identifying the most relevant attributes from high-dimensional datasets while discarding redundant or irrelevant ones. Efficient feature selection can significantly reduce computational complexity, enhance detection performance, and enable real-Stime decision-making—an essential requirement for IoT-based security systems. While existing methods like filter-based, wrapper-based, and embedded approaches offer various benefits, they often fall short in addressing the adaptive nature of modern DDoS threats, especially in resource-constrained IoT environments.

To address these challenges, this research proposes a novel, **bio-inspired ensemble feature selection approach** that integrates three powerful metaheuristic algorithms: **Spotted Hyena Optimizer (SHO), Parrot Optimizer (PO), and Grey Wolf Optimizer (GWO)**. These algorithms are inspired by the social behaviors and hunting strategies of animal species and are used here to optimize the selection of features dynamically across diverse datasets. Furthermore, the proposed approach leverages a **self-attention mechanism** to weigh feature importance adaptively, making the system more responsive to varying attack patterns.

The primary contributions of this paper are:

I. The development of a dynamic and adaptive feature selection framework using a hybrid ensemble of bio-inspired optimizers tailored for detecting DDoS attacks in IoT environments.

II. Implementation and evaluation of a lightweight IDS architecture based on the selected features, demonstrating robust detection capabilities across different datasets.

2.Literature Review:

[1] Pooja Kumari, Ankit Kumar Jain, "A comprehensive study of DDoS attacks over IoT network and their countermeasures," *Computers and Security*, vol. 127, p. 103096, 2023.

In this paper, Kumari and Jain provide a comprehensive overview of DDoS attacks in IoT networks, analyzing the various methods used to detect and mitigate such attacks. They delve into the vulnerabilities of IoT systems and propose countermeasures to prevent large-scale DDoS attacks that can incapacitate IoT services. The study highlights different attack vectors and the mechanisms through which attackers exploit IoT vulnerabilities. Furthermore, the paper discusses the effectiveness of traditional DDoS defense mechanisms and the need for innovative solutions that integrate IoT-specific countermeasures**.**

[2] Bindu Bala, Sunny Behal, "AI techniques for IoT-based DDoS attack detection: Taxonomies, comprehensive review and research challenges," *Computer Science Review*, vol. 52, p. 100631, 2024.

This paper by Bala and Behal presents a comprehensive review of AI-based techniques for detecting DDoS attacks in IoT environments. The authors classify various AI methodologies, including machine learning, deep learning, and hybrid models, and evaluate their effectiveness in countering DDoS attacks in IoT networks. The paper provides an extensive taxonomy of AI approaches, offering insight into the challenges that come with detecting unknown attack types. It also discusses future research directions, emphasizing the need for adaptive, real-time detection systems in the face of evolving IoT threats.

[3] Saurav Kumar, Ajit Kumar Keshri, "An effective DDoS attack mitigation strategy for IoT using an optimization-based adaptive security model," *Knowledge-Based Systems*, vol. 299, p. 112052, 2024.

Kumar and Keshri propose an optimization-based adaptive security model designed specifically for mitigating DDoS attacks in IoT environments. The model leverages optimization techniques such as genetic algorithms and swarm intelligence to dynamically adjust security measures based on network traffic patterns and attack characteristics. The adaptive nature of the model ensures that the security mechanisms can evolve as the nature of DDoS attacks changes, making the system resilient against a wide variety of attack scenarios.

[4] Monika Roopak, Gui Yun Tian, Jonathon Chambers, "An Intrusion Detection System Against DDoS Attacks in IoT Networks," ***IEEE Xplore*****, 2020.**
Roopak, Tian, and Chambers introduce an Intrusion Detection System (IDS) specifically designed to detect DDoS attacks in IoT networks. The IDS uses a combination of statistical analysis and machine learning techniques to identify traffic anomalies that are indicative of DDoS attacks. The paper highlights the importance of real-time monitoring in IoT networks to detect and mitigate attacks before they can cause significant damage. The authors demonstrate the efficacy of their IDS through simulations and propose further enhancements to improve detection accuracy in large-scale IoT networks.

[5] Vanlalruata Hnamte, Ashfaq Ahmad Najar, Hong Nhung-Nguyen, Jamal Hussain, Manohar Naik Sugali, "DDoS attack detection and mitigation using deep neural network in SDN environment," ***Computers and Security*****, vol. 138, p. 103661, 2024.**
This paper focuses on the use of deep neural networks (DNN) for detecting and mitigating DDoS attacks in a Software-Defined Networking (SDN) environment. Hnamte et al. propose a DNN-based approach that learns to distinguish between legitimate and malicious traffic by analyzing network flow data. The paper presents a detailed architecture for integrating the DNN model into SDN controllers, enabling real-time detection and mitigation of DDoS attacks. The proposed solution is scalable, efficient, and effective at handling high-volume attack traffic typical of DDoS incidents.

[6] Kavitha D, Ramalakshmi R, "Machine learning-based DDoS attack detection and mitigation in SDNs for IoT environments," ***Journal of the Franklin Institute*****, vol. 361, p. 107197, 2024.**
Kavitha and Ramalakshmi's work explores the use of machine learning algorithms for detecting and mitigating DDoS attacks in SDN-enabled IoT networks. They analyze the challenges posed by large-scale IoT deployments, where traditional DDoS mitigation techniques often fall short. The authors propose a hybrid model that combines supervised and unsupervised learning algorithms, such as Support Vector Machines (SVM) and clustering, to enhance the accuracy of attack detection. Their model significantly reduces false positives and improves the efficiency of mitigation strategies.

[7] Usman Haruna Garba, Adel N. Toosi, Muhammad Fermi Pasha, Suleman Khan, "SDN-based detection and mitigation of DDoS attacks on smart homes," ***Computer Communications*****, vol. 221, pp. 29–41, 2024.**
This study investigates the use of SDN-based approaches for DDoS attack detection and mitigation in smart home environments. Garba et al. propose a framework that utilizes SDN controllers to detect abnormal traffic patterns that may indicate a DDoS attack. The framework employs machine learning techniques to classify traffic as either benign or malicious, with automated mitigation actions taken in response to detected threats. The authors also consider the integration of this framework with existing smart home IoT systems, ensuring scalability and ease of deployment.

[8] Gaurav Dhiman, Vijay Kumar, "Spotted hyena optimizer: A novel bio-inspired based metaheuristic technique for engineering applications," ***Advances in Engineering Software*****, vol. 114, pp. 48–70, 2017.**
Dhiman and Kumar introduce the Spotted Hyena Optimizer, a novel bio-inspired metaheuristic technique designed for solving complex engineering problems, including DDoS attack mitigation. The algorithm mimics the hunting strategy of spotted hyenas and is applied to optimize feature selection and parameter tuning in machine learning models for DDoS detection. The paper presents the optimizer's advantages over traditional methods, including its ability to find optimal solutions in complex, high-dimensional spaces such as those encountered in IoT networks.

[9] Youfa Fu, Dan Liu, Jiadui Chen, Ling He, "Secretary bird optimization algorithm: A new metaheuristic for solving global optimization problems," ***Artificial Intelligence Review*****, vol. 57, pp. 57–123, 2024.**
Fu et al. introduce the Secretary Bird Optimization Algorithm, a new metaheuristic approach that can be applied to global optimization problems, including those in DDoS attack detection and mitigation. The paper demonstrates the algorithm's potential in solving feature selection and model optimization problems for machine learning-based DDoS detection systems. The authors highlight the advantages of using this algorithm in dynamic IoT environments, where network conditions can change rapidly.

[10] Hema Banati, Richa Sharma, Asha Yadav, "Binary Peacock Algorithm: A Novel Metaheuristic Approach for Feature Selection," *SpringerLink*, vol. 41, pp. 216–244, 2024.

In this paper, Banati, Sharma, and Yadav present the Binary Peacock Algorithm, a metaheuristic technique designed for feature selection in machine learning applications. This approach is particularly useful for improving the performance of DDoS detection systems by selecting the most relevant features from large datasets. The paper compares the Binary Peacock Algorithm with other feature selection methods, showing its superior performance in terms of accuracy and computational efficiency in real-time DDoS detection.

[11] Junbo Lian, Guohua Hui, Ling Ma, Ting Zhu, Xincan Wu, Ali Asghar Heidari, Yi Chen, Huiling Chen, "Parrot optimizer: Algorithm and applications to medical problems," *Computers in Biology and Medicine*, vol. 172, p. 108064, 2024.

Lian et al. introduce the Parrot Optimizer, a novel optimization algorithm inspired by the parrot's foraging behavior. The paper demonstrates its application to medical problems, but it also presents the potential uses of the algorithm in DDoS attack mitigation. The authors suggest that the Parrot Optimizer can be applied to IoT networks to optimize the resource allocation for DDoS detection systems, improving their responsiveness to attacks.

[12] Anushiya, V.S. Lavanya, "A new deep-learning with swarm-based feature selection for intelligent intrusion detection for the Internet of things," *Measurement: Sensors*, vol. 26, p. 100700, 2023.

Anushiya and Lavanya propose an innovative deep learning model integrated with swarm-based feature selection for IoT intrusion detection. The model is designed to efficiently detect DDoS attacks in real-time by selecting the most critical features from IoT traffic data. The authors demonstrate the effectiveness of their approach using several IoT network datasets, showing a significant reduction in false positives and enhanced detection accuracy for DDoS attacks.

[13] Fizza Rizvi, Ravi Sharma, Nonita Sharma, Manik Rakhra, Arwa N. Aledaily, Wattana Viriyasitavat, Kusum Yadav, Gaurav Dhiman and Amandeep Kaur, "An evolutionary KNN model for DDoS assault detection using genetic algorithm based optimization," *Springer Nature*, vol. 83, pp. 83005–83028, 2024.

Rizvi et al. introduce an evolutionary K-Nearest Neighbors (KNN) model optimized using a genetic algorithm (GA) for detecting Distributed Denial of Service (DDoS) attacks. The study highlights how combining evolutionary optimization techniques with KNN improves the accuracy of attack detection in dynamic network environments. By optimizing the feature selection process, the proposed model is shown to achieve a higher detection rate and lower false-positive rate when applied to IoT and cloud-based networks.

[14] O. Pandithurai, C. Venkataiah, Shrikant Tiwari and N. Ramanjaneyulu, "DDoS attack prediction using a honey badger optimization algorithm based feature selection and Bi-LSTM in cloud environment," *Expert Systems with Applications*, vol. 241, p. 122544, 2024.

Pandithurai et al. present a novel DDoS attack prediction method that integrates the Honey Badger Optimization (HBO) algorithm with Bi-directional Long Short-Term Memory (Bi-LSTM) networks for feature selection and attack detection in cloud environments. The Honey Badger Optimization algorithm is used to efficiently select the most relevant features from large datasets, while the Bi-LSTM model captures both past and future information in the traffic data. The results demonstrate that the proposed model outperforms traditional methods in predicting DDoS attacks with improved precision and reduced false positives.

[15] Ahmed Ahmim, Faiz Maazouzi, Marwa Ahmim, Sarra Namane, Imed Ben Dhaou, "Distributed Denial of Service Attack Detection for the Internet of Things Using Hybrid Deep Learning Model," *IEEE Xplore*, vol. 11, p. 119862, 2023.

Ahmim et al. propose a hybrid deep learning model for detecting DDoS attacks in the Internet of Things (IoT) environment. The model combines Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) to extract both spatial and temporal features from network traffic data. The hybrid approach significantly improves detection accuracy compared to traditional machine learning models by effectively addressing the challenges posed by the highly dynamic and distributed nature of IoT networks. The study also highlights the model's scalability and robustness in handling large-scale IoT deployments.

[16] Xuan-Ha Nguyen, Kim-Hung Le, "Robust detection of unknown DoS/DDoS attacks in IoT networks using a hybrid learning model," *Internet of Things*, vol. 23, p. 100851, 2023.
Nguyen and Le present a hybrid learning model for detecting both known and unknown Denial of Service (DoS) and Distributed Denial of Service (DDoS) attacks in IoT networks. The proposed model combines the strengths of deep learning and ensemble learning methods to improve the detection capabilities of traditional systems. The model demonstrates its effectiveness in identifying previously unseen attack patterns by learning from a large and diverse set of network traffic features. The study shows that hybrid learning methods can significantly enhance the resilience of IoT networks against evolving cyber threats**.**

[17] Mohamed Aly Bouke, Azizol Abdullah, Sameer Hamoud ALshatebi, Mohd Taufik Abdullah, Hayate El Atigh, "An intelligent DDoS attack detection tree-based model using Gini index feature selection method," *Microprocessors and Microsystems*, vol. 98, p. 104823, 2023.
Bouke et al. propose an intelligent decision tree-based model for DDoS attack detection, utilizing the Gini index for feature selection. The Gini index is applied to select the most significant features from network traffic data, which are then fed into a decision tree classifier for attack detection. The authors argue that this approach provides a balance between high detection accuracy and computational efficiency, making it suitable for real-time monitoring of IoT networks. The study confirms that the proposed method is effective in reducing false positives and enhancing the detection of DDoS attacks in diverse network environments.

[18] Md. Alamgir Hossain, Md. Saiful Islam, "Enhancing DDoS attack detection with hybrid feature selection and ensemble-based classifier: A promising solution for robust cybersecurity," *Measurement: Sensors*, vol. 32, p. 101037, 2024.
Hossain and Islam focus on enhancing DDoS attack detection by integrating hybrid feature selection with ensemble-based classifiers. Their method applies a combination of filter and wrapper feature selection techniques to reduce the dimensionality of network traffic data, followed by an ensemble classifier that combines multiple learning algorithms. The results show that this hybrid approach significantly improves detection performance, providing a promising solution for enhancing cybersecurity in IoT and cloud-based environments. The authors also emphasize the scalability and adaptability of their approach in handling large-scale, dynamic IoT systems.

[19] Anupama Mishra, Neena Gupta, Brij B. Gupta, "Defensive mechanism against DDoS attack based on feature selection and multi-classifier algorithms," *Springer Nature*, vol. 82, pp. 229–244, 2023.
Mishra, Gupta, and Gupta propose a defensive mechanism against DDoS attacks that integrates feature selection and multi-classifier algorithms. The mechanism first uses feature selection techniques to reduce irrelevant and redundant data from network traffic. Then, multiple classifiers, including Support Vector Machine (SVM) and K-Nearest Neighbors (KNN), are applied to classify traffic and identify potential attacks. The multi-classifier approach is designed to improve detection accuracy by leveraging the strengths of various machine learning models. The paper demonstrates that this defensive mechanism is highly effective in mitigating DDoS attacks, with the ability to adapt to new and evolving attack strategies.

3.Problem Statement and Research Motivation:

The rapid adoption of Internet of Things (IoT) technologies has significantly expanded the attack surface in modern cyber-physical systems. Due to their constrained computational resources, heterogeneous architectures, and lack of unified security standards, IoT devices are particularly susceptible to Distributed Denial of Service (DDoS) attacks. These attacks leverage large numbers of compromised devices to flood network infrastructure, leading to service disruption, financial damage, and potential threats to human safety in critical applications.

Traditional Intrusion Detection Systems (IDS) designed for conventional networks often fail to perform effectively in IoT environments. This ineffectiveness stems from several key challenges:

1. **Resource Constraints**: IoT devices typically operate with limited memory, CPU power, and energy supply, making it infeasible to deploy heavy or complex IDS models directly on edge nodes.
2. **High-Dimensional Feature Spaces**: Network traffic data in IoT systems is often high-dimensional and noisy. Processing such data without effective feature selection results in high computational cost, poor detection accuracy, and model overfitting.
3. **Static Feature Selection Approaches**: Most existing IDS frameworks rely on static feature selection techniques, which lack adaptability to evolving network behaviors and zero-day attack patterns, □ reducing their robustness in dynamic environments.

4. **Scalability and Generalization**: Solutions developed for specific datasets or platforms often fail to generalize across diverse IoT deployments due to protocol heterogeneity and varying traffic distributions.
5. **Latency in Decision Making**: Centralized detection frameworks that rely on offloading data to the cloud introduce latency, which is unsuitable for mission-critical IoT applications requiring real-time threat response.

To address these limitations, this research is motivated by the need to develop a **lightweight, adaptive, and accurate IDS framework** tailored for IoT ecosystems. Specifically, we propose an intelligent feature selection mechanism based on **bio-inspired optimization algorithms**, which emulate natural behaviors to explore and exploit complex search spaces efficiently.

Unlike traditional optimization techniques, bio-inspired algorithms such as **Spotted Hyena Optimizer (SHO)**, **Parrot Optimizer (PO)**, and **Grey Wolf Optimizer (GWO)** dynamically adapt to changing input patterns and provide robust global search capabilities. When used in combination, these algorithms can effectively identify the most relevant features from large datasets while minimizing computational overhead.Moreover, the integration of a **self-attention mechanism** within the feature selection pipeline further enhances the model's ability to learn contextual relationships between features, enabling dynamic relevance weighting and improved detection accuracy.

This research aims to build a hybrid, ensemble-based IDS that:

- Selects optimal features adaptively using nature-inspired intelligence,
- Minimizes resource usage for deployment in edge environments,
- Maintains high detection performance across diverse IoT scenarios.

4.RESEARCH METHODOLOGY

Algorithm : 1- Ensemble Feature Selection

```
Input: Processed dataset X
Output: Final selected_feature _set
Step 1: Apply Self-Attention Mechanism
Step 2: Apply Feature Selection Algorithms
Spotted Hyena Optimizer (SHO)
Parrot Optimization (PO)
Grey Wolf Optimizer (GWO)
Step 3: Combine Feature Sets
Compute the union of selected features: F_final = F_SHO ∪ F_PO ∪ F_GWO
Return F_final
```

Table:1-Ensemble Feature Selection

This section presents the proposed methodology based on an ensemble feature selection approach for improving the accuracy of DDoS attack detection in IoT networks. As illustrated in Fig. 1, the methodology comprises a sequence of steps: **data preprocessing, self-attention mechanism, ensemble feature selection using bio-inspired optimization algorithms, model training, and performance evaluation**. Initially, data preprocessing techniques are applied to handle missing values, normalize the data, and balance class distributions to prepare high-quality input for the next stages. A **self-attention mechanism** is then employed to capture important dependencies and interactions between features, which enhances the feature representation. In the feature selection phase, multiple **bio-inspired optimization algorithms**—such as Particle Swarm Optimization (PSO) and Genetic Algorithms (GA)—are collaboratively used to select an optimal subset of features, leveraging the strengths of each algorithm. The selected features are then fed into machine learning models for training, with an emphasis on minimizing overfitting and improving generalization. Finally, **performance evaluation** is conducted using various metrics such as accuracy, precision, recall, F1-score, and AUC-ROC to comprehensively assess the model's effectiveness. This multi-stage process aims to identify the most relevant features while reducing redundancy and noise, thus significantly enhancing both classification performance and computational efficiency. Furthermore, the ensemble approach ensures robustness and stability in feature selection, contributing to a more reliable DDoS detection framework in dynamic IoT environments.

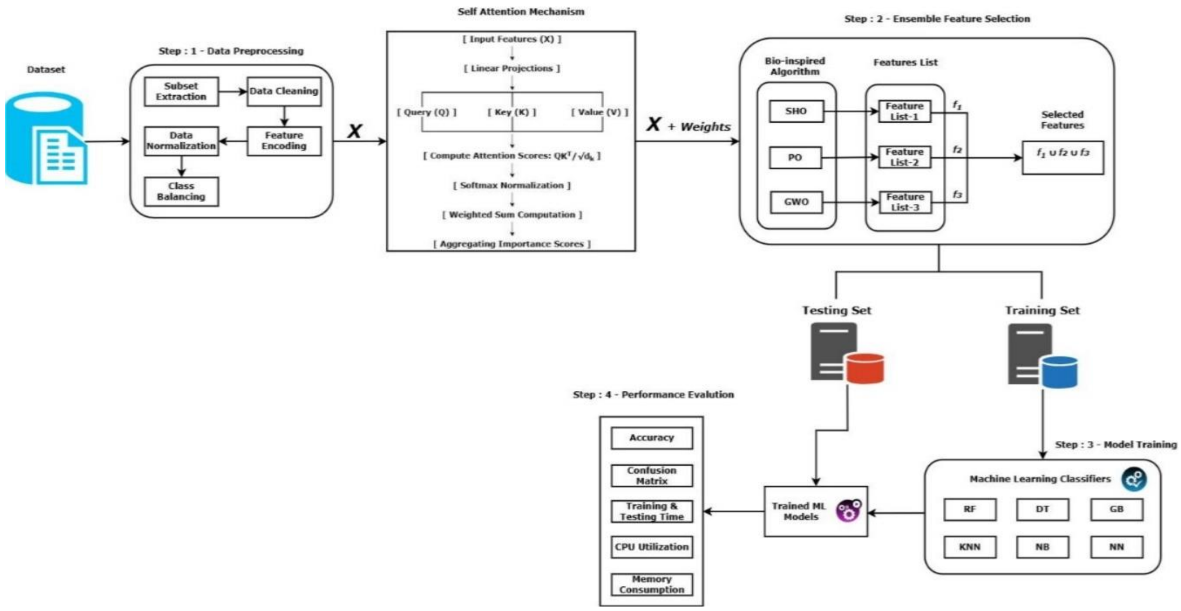


Fig 1: System Architecture

4.1 Preprocessing of Input Features:

The initial phase of the proposed methodology involves multiple preprocessing steps to prepare the dataset for model training. These steps include subset extraction, data cleaning, feature encoding, normalization, and class balancing. Given the focus on DDoS detection, the subset extraction process selects samples corresponding to DoS, DDoS, and normal traffic.

Data cleaning addresses missing values, outdated entries, and outliers to improve data quality and ensure consistency. Categorical features are converted into numerical form using label encoding, enabling compatibility with machine learning algorithms. To bring all feature values within a uniform range, min–max normalization is applied, scaling numerical data to the range [0.0, 1.0].

As the classification task is binary, class balancing is performed to equalize the number of instances in both classes. This step mitigates the bias toward the majority class and enhances the model's ability to detect minority-class instances accurately.

4.2 Self-Attention-Based Feature Importance Estimation

The self-attention mechanism plays a key role in assessing the relevance of different features within the input data by learning their relationships. Unlike conventional feature selection methods that assign static importance scores, self-attention dynamically adapts based on the data context.

The process begins by transforming input features into three distinct representations—queries, keys, and values—using trainable weight matrices. These representations help the model determine which features should influence others during the learning process. By computing similarity scores through dot-product operations between queries and keys, and normalizing them using the softmax function, the model assigns attention weights to each feature.

These weights indicate the relative importance of one feature in relation to the others, allowing the model to focus on informative attributes while diminishing the influence of redundant or noisy ones. Finally, by aggregating these weights across the dataset, the model derives a global importance score for each feature. High-ranking features are retained, while lower-ranking ones may be discarded. This mechanism enables efficient, automated feature selection in high-dimensional IoT data without manual intervention, improving both performance and interpretability. Unlike traditional feature selection methods (e.g., filter- and wrapper-based approaches), self-attention does not treat input features equally. Instead, it uses dynamic attention scores to identify which features are most significant. The mechanism computes a weighted sum of all input features, giving more importance to the relevant ones while suppressing the irrelevant ones.

Traditional feature selection methods often operate on fixed importance scores, such as correlation coefficients. In contrast, self-attention dynamically learns the importance of features from the input data itself. Features that contribute more to the decision-making process are assigned higher attention weights, whereas irrelevant features are down-weighted or suppressed.

One of the key advantages of the self-attention mechanism is its ability to process high-dimensional datasets efficiently. This efficiency is especially notable when compared to classic wrapper-based techniques, such as Recursive Feature Elimination (RFE), which can result in exponential increases in computational cost. Unlike traditional feature selection methods, which require domain knowledge and manual intervention (e.g., removal of correlated features), self-attention automates this process. By allowing the model to understand the value of each feature without human involvement, self-attention provides a more streamlined and effective approach to feature selection.

The first step is input representation, in which the dataset is shown as a feature matrix X, with each column denoting a feature and each row representing a sample. Next phase is Linear projections where three learnable weight matrices are applied to the input to generate Query (Q), Key (K), and Value (V). Learnable weight matrices are used to convert each feature vector into three representations: Query(Q)- What more features

$$Q = XW_Q, \quad K = XW_K, \quad V = XW_V$$

does this feature seek? Key (K)- The information provided by this feature. The weighted sum will be calculated using the value (V) as the actual feature representation

where WQ, WK, WV are learnable weight matrices. This transformation allows the model to learn different aspects of the features and capture relationships between them. The computation of attention scores comes next where the Dot-product attention is used by the attention mechanism to calculate feature similarity scores. This stage establishes the relative importance of each attribute to the others

where dk is the dimension of key vectors. Next is Softmax Normalization. Using the Softmax function, the

$$\text{Attention\_Score} = \frac{QK^T}{\sqrt{d_k}}$$

attention scores are transformed into probabilities. In order to interpret the attention weights as probabilities, this guarantees that they add up to 1. Attention weights will be higher for features that are more important and aids in figuring out which characteristics influence decisions the most. After this, the weighted sum of values is calculated because the features with higher attention weights contribute more to the final output (i.e) focuses

$$\text{Attention\_Weights} = \text{softmax}(\text{Attention\_Score})$$

$$\text{Output} = \text{Attention\_Weights} \times V$$

on features that provide the most meaningful information while reducing the impact of less relevant ones.

Next, Attention weights are aggregated for feature selection. We calculate the average attention weight for all samples to ascertain the overall significance of each attribute. Each feature is given a feature importance score

$$\text{Feature\_Importance} = \text{mean}(\text{Attention\_Weights}, \text{axis} = 0)$$

as a consequence. High-scoring features are chosen, while low-scoring ones may be eliminated. This eliminates the need for human selection methods and offers a ranking of feature relevance.

Finally, selecting the most important features is done that helps in enhances model efficiency, reduces overfitting, and improves performance. The main advantage of using Self attention is, instead of assessing each feature separately, it captures their dependencies, allocates feature importance dynamically, enabling data-driven decision-making, and does away with the necessity for feature engineering by hand. It works effectively with high-dimensional datasets that makes the model easier to understand using attention scores. Self-attention analyzes the relationships between features to automatically choose the most pertinent ones. An effective feature selection technique for machine learning, cybersecurity, and healthcare applications

4.3 Bio-Inspired Feature Selection for IoT Intrusion Detection

Due to the high dimensionality of network traffic data in IoT systems, selecting the most relevant features is crucial for accurate and efficient intrusion detection. This study adopts an ensemble-based approach that combines three bio-inspired optimization algorithms to perform feature selection: Spotted Hyena Optimizer (SHO), Parrot Optimizer (PO), and Grey Wolf Optimizer (GWO). Each of these algorithms draws inspiration from natural behaviors that help guide the search for optimal solutions.

The SHO algorithm simulates the collaborative hunting strategy of hyenas to explore the feature space effectively, identifying potentially valuable subsets. The PO algorithm mimics the learning and communication behavior of parrots, allowing it to refine candidate solutions over time through imitation and feedback. Lastly, the GWO algorithm leverages the social hierarchy and hunting coordination of grey wolves to fine-tune the selection process by converging toward the most promising features.

By integrating these three methods, the ensemble balances exploration and exploitation—ensuring comprehensive coverage of the feature space while avoiding overfitting. This hybrid feature selection strategy significantly reduces the computational burden while maintaining high detection accuracy, making it well-suited for lightweight intrusion detection in real-time IoT environments.

This study incorporates bio-inspired optimization techniques for feature selection. These techniques are modeled on the biological traits and social behavior of animals. Specifically, the social interactions of animals inspire the implementation of three algorithms: the Spotted Hyena Optimizer (SHO), the Parrot Optimizer (PO), and the Grey Wolf Optimizer (GWO). These algorithms are selected based on their unique optimization capabilities, thereby ensuring a balanced trade-off between exploration and exploitation in the feature selection process.

The SHO algorithm mimics the cooperative hunting behavior of spotted hyenas to explore the feature space efficiently. It performs well with large-scale IoT datasets by eliminating redundant features, reducing computational costs, and enhancing classification accuracy. The PO algorithm is inspired by the behavioral patterns of parrots, particularly their ability to learn from their environment and replicate optimal solutions. PO leverages reinforcement-based learning to maintain solution diversity and fine-tune feature subsets, thereby delivering high performance in high-dimensional spaces typical of IoT security datasets. The GWO algorithm, which simulates the leadership hierarchy and hunting strategy of grey wolves, is effective for refining feature selection through exploitation. It identifies the most relevant features for anomaly and intrusion detection, making it suitable for IoT-based security applications.

By integrating SHO, PO, and GWO, this ensemble method enhances feature selection accuracy through a complementary balance of exploration (via SHO and PO) and exploitation (via GWO). Each algorithm contributes its respective strengths, resulting in a robust and generalizable feature selection mechanism. The hybrid approach reduces processing overhead by retaining only the most pertinent features, thereby yielding faster and more resource-efficient models. This ensemble technique significantly enhances cybersecurity frameworks, improves detection accuracy, and optimizes resource utilization for mitigating IoT-based threats such as DDoS attacks.

The Spotted Hyena Optimizer (SHO) contributes strong exploratory capabilities, enabling the discovery of diverse and impactful feature subsets. The Parrot Optimizer (PO) continuously learns and improves feature selection through environmental adaptation. The Grey Wolf Optimizer (GWO), focused on exploitation, fine-tunes the best-selected attributes to ensure optimal classification. The synergistic combination of SHO, PO, and GWO facilitates a comprehensive, accurate, and efficient feature selection process, ideally suited for real-time intrusion detection in IoT ecosystems**.**

4.3.1 Spotted Hyena Optimizer (SHO)

The Spotted Hyena Optimizer (SHO) is a nature-inspired metaheuristic algorithm based on the behavior and social structure of *Crocuta crocuta*, commonly known as the spotted hyena. Spotted hyenas are large, dog-like carnivores known for their unique characteristics such as intelligence, social complexity, and efficient hunting strategies. They live for approximately 10–12 years in the wild and up to 25 years in captivity. The four recognized species of hyenas include the spotted hyena, striped hyena, brown hyena, and aardwolf. Among them, the spotted hyena is the largest and most skillful hunter.Spotted hyenas are often referred to as "laughing hyenas" due to their vocalizations resembling human laughter. Their reddish-brown fur is marked with black spots. These animals are capable of forming large social groups, sometimes exceeding 100 members, based on kinship and mutual association. Their group behaviour includes tracking prey by sight, smell, and sound. These hunting and social strategies are mathematically modelled in the SHO algorithm for optimization purposes.

Fig 2: Attacking prey

4.3.1.1 Encircling Behavior:

The target behaviour or objective is considered as the best solution and the other search agents can update their positions with respect to obtained best solution. Spotted hyenas can know where their prey is and surround them [10]. We consider the current best candidate is the spotted hyena closest to the target or prey because of search space not known a priori. The locations of other search agents are updated after the best search solution is defined.

4.3.1.2. Hunting

The next step of SHO algorithm is the hunting strategy which makes a cluster of optimal solutions against the best search agent and updates the positions of other search agents. In order to mathematically imitate the hunting behaviour of spotted hyena, we suppose that the best search agent is optimum, which is consider as the location of prey, the other search agent towards the best search agent, constantly update their positions until to find the best solutions, then save the best solution [11].

4.3.1.3. Attacking Behaviour

The best solution and updates the positions of other search agents on the basis of the position of the best agent, the spotted hyena attack the prey constantly updates their position [12].

4.3.1.4. Search for Behaviour

The searching mechanism describes the exploration capability of an algorithm. The proposed SHO algorithm ensures thus capability using random values which are greater than or less than 1. The vector is also responsible to show the more randomized behavior of SHO and avoid local optimum [13].

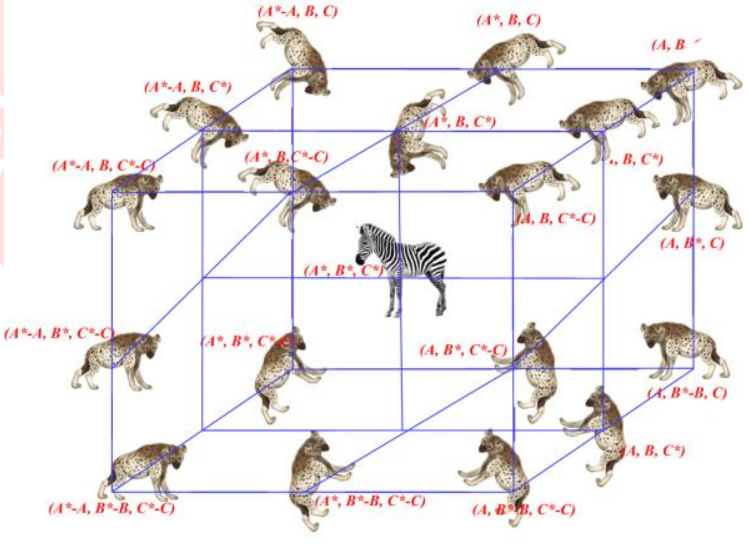
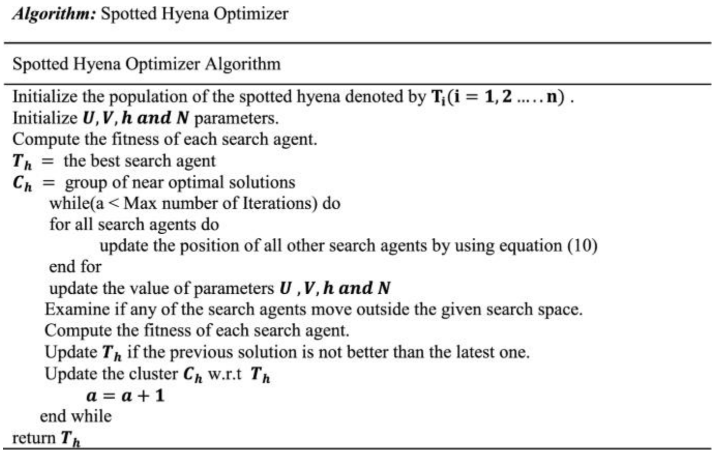


Fig 3: SHO Attacking Prey Mechanism

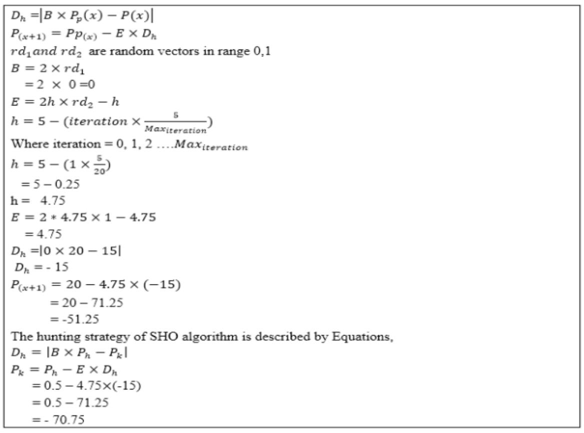
4.3.1.5 Algoritm and Numerical Expression of SHO Algorithm:



```
Algorithm: Spotted Hyena Optimizer

Spotted Hyena Optimizer Algorithm

Initialize the population of the spotted hyena denoted by $T_i (i = 1, 2 ..... n)$ .
Initialize $U, V, h$ and $N$ parameters.
Compute the fitness of each search agent.
$T_h$ = the best search agent
$C_h$ = group of near optimal solutions
    while(a < Max number of Iterations) do
    for all search agents do
        update the position of all other search agents by using equation (10)
    end for
    update the value of parameters $U, V, h$ and $N$
    Examine if any of the search agents move outside the given search space.
    Compute the fitness of each search agent.
    Update $T_h$ if the previous solution is not better than the latest one.
    Update the cluster $C_h$ w.r.t $T_h$
        $a = a + 1$
    end while
return $T_h$
```



$D_h = |B \times P_p(x) - P(x)|$

$P_{(x+1)} = Pp_{(x)} - E \times D_h$

$rd_1 and\ rd_2$ are random vectors in range 0,1

$B = 2 \times rd_1$

$= 2 \times 0 = 0$

$E = 2h \times rd_2 - h$

$h = 5 - (iteration \times \frac{5}{Max_{iteration}})$

Where iteration = 0, 1, 2 ....$Max_{iteration}$

$h = 5 - (1 \times \frac{5}{20})$

$= 5 - 0.25$

h = 4.75

$E = 2 * 4.75 \times 1 - 4.75$

$= 4.75$

$D_h = |0 \times 20 - 15|$

$D_h = - 15$

$P_{(x+1)} = 20 - 4.75 \times (-15)$

$= 20 - 71.25$

$= -51.25$

The hunting strategy of SHO algorithm is described by Equations,

$D_h = |B \times P_h - P_k|$

$P_k = P_h - E \times D_h$

$= 0.5 - 4.75 \times (-15)$

$= 0.5 - 71.25$

$= - 70.75$

Fig 4: Algorithm

Fig 5: Numerical Expression

4.3.2 Parrot Optimizer (PO)

Pyrrhura Molinae, a well-known parrot species, is a popular choice among pet owners due to its appealing qualities, close bond with its owners, and ease of training. *Pyrrhura Molinae* exhibits four distinct behavioral traits: foraging, remaining, communication, and fear of strangers. The Parrot Optimizer mimics a parrot's natural foraging strategy, in which they investigate various food sources and learn from their surroundings.
For staying behavior, parrots maintain their current place rather than exploring new locations, allowing them to reward great solutions while reducing unnecessary movements, resulting in improved convergence. Parrots' communicative activity enables them to share knowledge about better areas, allowing lesser competitors to learn from stronger ones. This behaviour enhances exploitation, speeds up convergence, and ensures that the best solutions reach the whole population, resulting in superior optimization results.
Pyrrhura Molinae's natural fear of strangers, which is common among birds, causes them to escape from unexpected humans and seek refuge with their owners for safety. This exemplifies how parrots react to unfamiliar or terrifying situations by fleeing to specific spots. By encouraging search agents to travel to new locations, this improves exploration by assisting them in escaping local o finding better solutions.

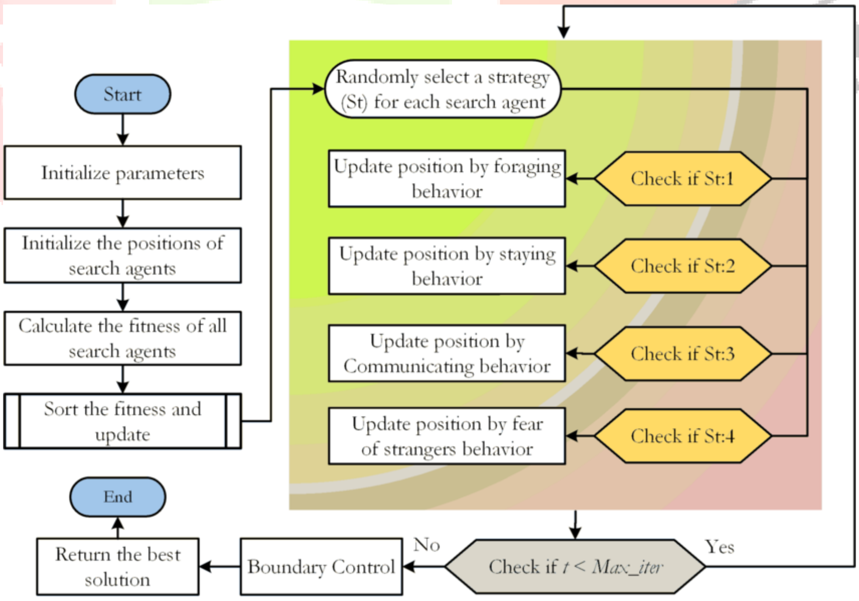

Start
Initialize parameters
Initialize the positions of search agents
Calculate the fitness of all search agents
Sort the fitness and update
Randomly select a strategy (St) for each search agent
Update position by foraging behavior
Check if St:1
Update position by staying behavior
Check if St:2
Update position by Communicating behavior
Check if St:3
Update position by fear of strangers behavior
Check if St:4
Check if t < Max_iter
No
Yes
Boundary Control
Return the best solution
End

Fig:6 Flow diagram of the Parrot Optimization algorithm for optimization problem solving

Nature-inspired optimization algorithms have gained significant attention due to their simplicity, flexibility, and ability to avoid local optima. Inspired by the foraging and communication behaviour of parrots, the Parrot Optimization Algorithm (POA) mimics how parrots learn from each other and explore their surroundings efficiently. POA integrates memory-based learning and flocking behaviour to balance exploration and exploitation in the search space.

Numerous algorithms have been introduced over the years, such as Genetic Algorithms (GA), Ant Colony Optimization (ACO), and Particle Swarm Optimization (PSO). Each algorithm attempts to solve complex optimization problems through a different aspect of natural behaviour. The POA builds upon this foundation by introducing vocal imitation and social interactions among agents as core principles

Importantly, the reason for our design is shown by the unpredictable behaviour of *Pyrrhura Molinae*, which exhibits all four behaviors at random throughout each cycle in domesticated flocks.

$$X_i^{t+1} = (X_i^t - X_{best}) \,.\, Levy(dim) + rand(0,1) \,.\, (1 - \frac{1}{Max_{iter}})^{\frac{2t}{Max_{iter}}} \,.\, X_{mean}^t \rightarrow$$

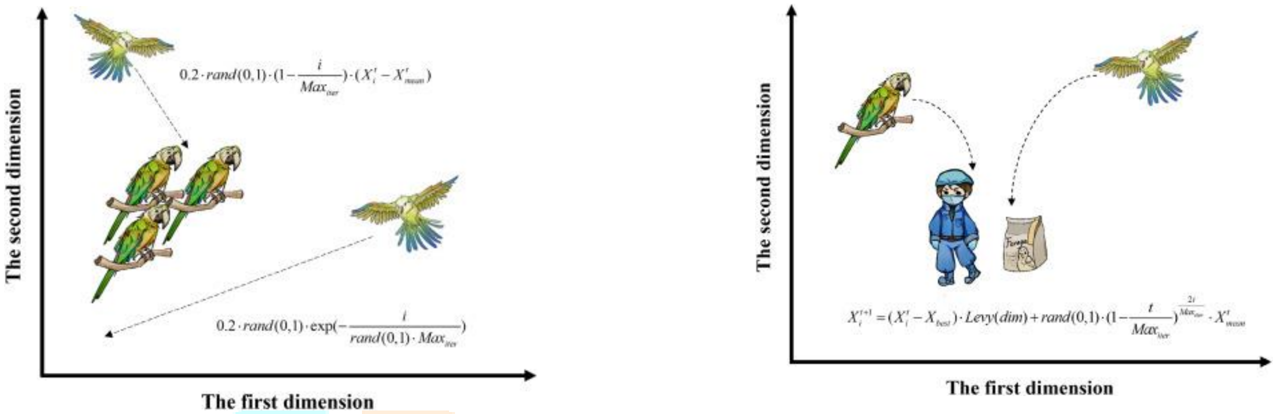
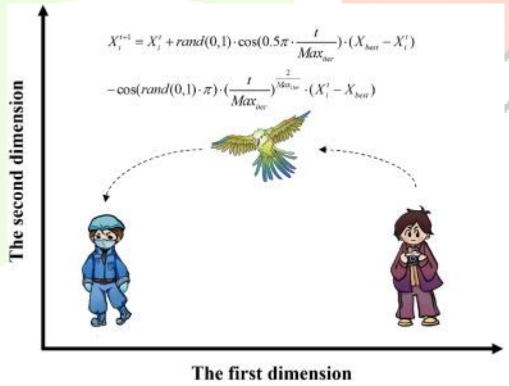

$0.2 \cdot rand(0,1) \cdot (1 - \frac{i}{Max_{iter}}) \cdot (X_i^t - X_{mean}^t)$
$0.2 \cdot rand(0,1) \cdot \exp(-\frac{i}{rand(0,1) \cdot Max_{iter}})$
The second dimension
The first dimension
$X_i^{t+1} = (X_i^t - X_{best}) \cdot Levy(dim) + rand(0,1) \cdot (1 - \frac{t}{Max_{iter}})^{\frac{2t}{Max_{iter}}} \cdot X_{mean}^t$
The second dimension
The first dimension

Fig 7:Stranger Avoidance and Owner Orientation Behavior in PO

4.3.2.1 Communication Mechanism

In natural ecosystems, parrots tend to form flocks where individuals share information through vocalizations and gestures. This behaviour is crucial for food localization and predator awareness. To simulate this process, the optimizer uses two dynamic strategies: group joining and immediate departure after communication. The position update rule is given as:

$$X_i^{t+1} = \begin{cases} 0.2 \cdot rand(0,1) \cdot \left(1 - \frac{t}{Max_{iter}}\right) \cdot (X_i^t - X_{mean}^t),\ P \leq 0.5 \\ 0.2 \cdot rand(0,1) \cdot exp\left(-\frac{t}{rand(01,) \cdot Max_{iter}}\right),\ P > 0.5 \end{cases}$$


$X_i^{t+1} = X_i^t + rand(0,1) \cdot \cos(0.5\pi \cdot \frac{t}{Max_{iter}}) \cdot (X_{best} - X_i^t)$
$- \cos(rand(0,1) \cdot \pi) \cdot (\frac{t}{Max_{iter}})^{\frac{2}{Max_{iter}}} \cdot (X_i^t - X_{best})$
The second dimension
The first dimension

Where,

X_i^{t+1} → denotes the location of the succeeding update,
X_i^t → denotes the present location,
X_{mean}^t → average location inside the present population,
Levy(D) → denotes the Levy distribution,
X_{best} → best position that has been searched from initialization to the current,

The first term facilitates social cohesion and knowledge sharing, improving the exploitation capability. The second term allows random exploration by escaping the current group, which helps maintain population diversity.

5.3.2.2 Stranger Avoidance and Owner Orientation

Parrots exhibit cautious behaviour when confronted with unfamiliar or threatening entities. This avoidance is balanced by their inherent desire to return to the safety of their owner or known locations. The optimizer models this dual behaviour by using trigonometric functions to control motion direction and step size

$$\boldsymbol{X_i^{t+1} = X_i^t + \text{rand}(0{,}1) \cdot \cos\left(0.5\pi \cdot \frac{t}{Max_{iter}}\right) \cdot (X_{best} - X_i^t)\text{-} \;\; \cos(rand(0,1) \cdot \pi) \cdot \left(\frac{t}{Max_{iter}}\right)^{\frac{2}{Max_{iter}}} \cdot (X_i^t - X_{best})}$$

Here:

- Xbest is the best-known solution up to iteration ttt,
- The cosine-based first term encourages convergence to better areas (i.e., flying towards the owner),
- The second cosine-based term simulates deviation from risk zones (stranger avoidance).
- $0.2 \cdot rand(0{,}1) \cdot \left(1 - \frac{t}{Max_{iter}}\right) \cdot (X_i^t - X_{mean}^t)$ → denotes the process of an individual joining a parrot's group to communicate,
- $0.2 \cdot rand(0{,}1) \cdot exp\left(-\frac{t}{rand(01,) \cdot Max_{iter}}\right)$ → denotes the process of an individual flying away immediately after communicating,
- $\cos\left(0.5\pi \cdot \frac{t}{Max_{iter}}\right) \cdot (X_{best} - X_i^t)$ → shows the process of reorientating to fly towards the owner,
- $\cos(rand(0{,}1) \cdot \pi) \cdot \left(\frac{t}{Max_{iter}}\right)^{\frac{2}{Max_{iter}}} \cdot (X_i^t - X_{best})$ → shows the process of moving away from the strangers.

4.3.2.3 Fitness Evaluation and Solution Selection

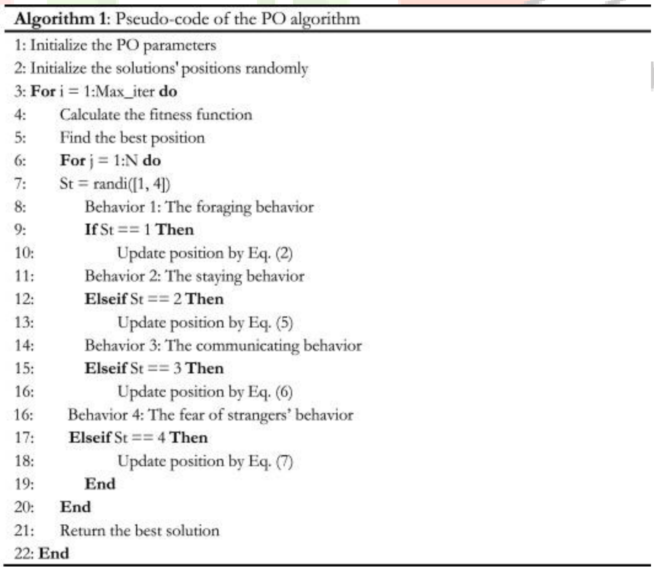
To evaluate the quality of candidate solutions, the optimizer employs a wrapper-based feature selection strategy. Each individual XiX_iXi represents a binary vector that selects a subset of features. The selected subset is then evaluated using a **Random Forest (RF)** classifier to determine classification accuracy. The fitness function is defined as:

$$Fitness(X_i^t) = Accuracy_{RF}(X_i^t)$$

Where:

- AccuracyRF is the classification accuracy of the selected feature subset using RF,
- The higher the accuracy, the better the subset.

Only solutions with higher accuracy are preserved in subsequent generations, ensuring convergence toward high-quality feature subsets. Additionally, elitism may be applied to retain the best-performing individuals across generations.



```
Algorithm 1: Pseudo-code of the PO algorithm
Initialize the PO parameters
Initialize the solutions' positions randomly
For i = 1:Max_iter do
    Calculate the fitness function
    Find the best position
    For j = 1:N do
    St = randi([1, 4])
        Behavior 1: The foraging behavior
        If St == 1 Then
            Update position by Eq. (2)
        Behavior 2: The staying behavior
        Elseif St == 2 Then
            Update position by Eq. (5)
        Behavior 3: The communicating behavior
        Elseif St == 3 Then
            Update position by Eq. (6)
      Behavior 4: The fear of strangers' behavior
      Elseif St == 4 Then
            Update position by Eq. (7)
        End
    End
    Return the best solution
End
```

Fig. 8: Pseudocode of the Parrot Optimizer

4.3.3 Grey Wolf Optimizer (GWO)

The Grey Wolf Optimizer (GWO) algorithm is inspired by the social hierarchy and hunting behaviour of grey wolves (*Canis lupus*), a species belonging to the Canidae family and classified as apex predators. Grey wolves typically live in structured packs, averaging between 5 to 12 members. Their social structure is strictly hierarchical and is generally composed of four primary roles: alpha, beta, delta (or subordinate), and omega.

At the top of the hierarchy are the **alpha wolves**, usually a dominant male and female pair, responsible for making crucial decisions such as hunting times, sleeping locations, and general group movement. The pack abides by the decisions made by the alphas, although democratic behaviour has also been observed in some scenarios where the alpha follows the will of the group.
Alphas are not necessarily the strongest wolves but are often the most capable leaders, illustrating that leadership and control are more critical than sheer strength.

The **beta wolves** serve as second-in-command and act as advisors to the alphas. They reinforce the alpha's decisions and ensure discipline within the pack. A beta can be male or female and is often the most likely successor to the alpha role in the event of absence or death.

At the bottom of the hierarchy lies the **omega wolf**, which functions as a scapegoat and plays a crucial role in diffusing tension and aggression within the pack. Omegas are submissive to all other wolves and are the last to feed. Despite their low status, the loss of an omega has been shown to disrupt pack harmony due to the lack of an outlet for collective frustration.

Delta wolves, or subordinates, rank above the omega but below the alpha and beta. This group encompasses scouts, sentinels, hunters, elders, and caretakers. Each role contributes to the well-being of the pack: scouts monitor territory boundaries, sentinels ensure safety, hunters assist in obtaining food, elders offer wisdom, and caretakers look after weak or injured wolves.

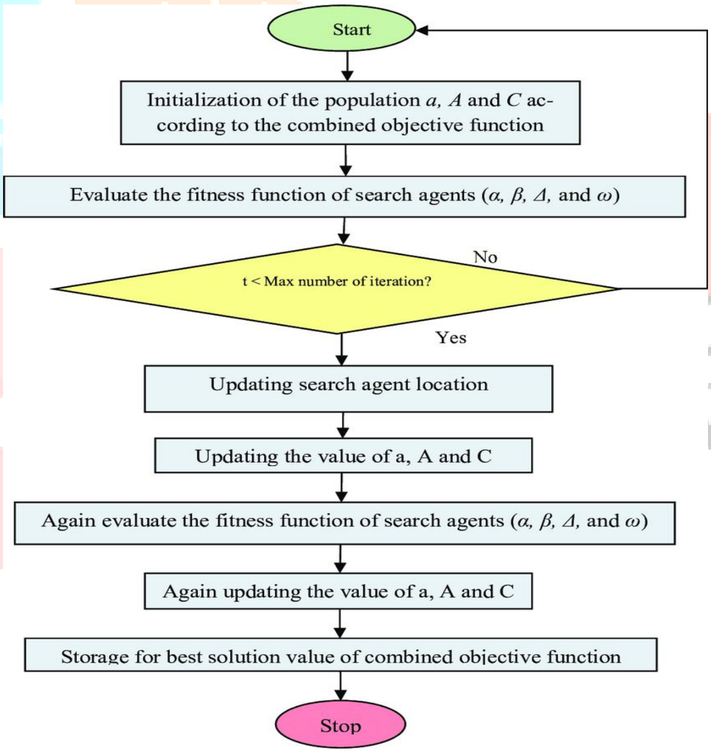

Start
Initialization of the population a, A and C according to the combined objective function
Evaluate the fitness function of search agents (α, β, Δ, and ω)
t < Max number of iteration?
No
Yes
Updating search agent location
Updating the value of a, A and C
Again evaluate the fitness function of search agents (α, β, Δ, and ω)
Again updating the value of a, A and C
Storage for best solution value of combined objective function
Stop

Fig:9 Flow diagram of the Grey Wolf algorithm for optimization problem solving

This hierarchical structure is directly mapped onto the GWO algorithm. The **alpha (α\alphaα)** represents the best solution found so far, followed by the **beta (β\betaβ)** and **delta (δ\deltaδ)** as the second and third best solutions, respectively. The remaining candidate solutions are classified as **omega (ω\omegaω)** wolves.

The GWO algorithm simulates the grey wolves' hunting process through three main phases:

1. **Tracking**: Wolves estimate the location of the prey (optimal solution) and adjust their positions based on the top solutions found so far.
2. **Encircling**: Wolves encircle the prey by updating their positions relative to the alpha, beta, and delta wolves.
3. **Attacking**: Once the prey is encircled, wolves begin to attack by reducing the randomness of their movements, allowing for intensified exploitation and convergence toward the best solution.

This combination of exploration and exploitation, guided by the alpha, beta, and delta positions, enables the GWO algorithm to effectively solve complex optimization problems.

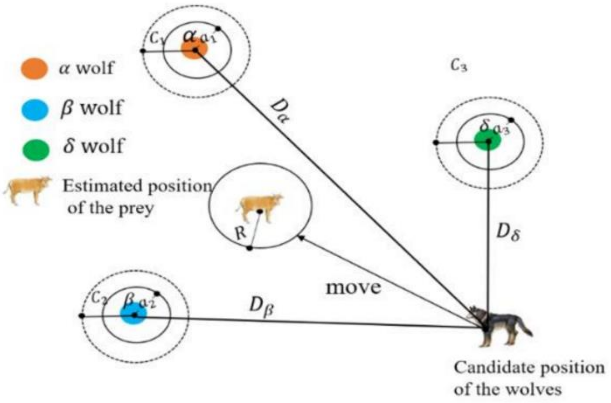


Fig 10 ;Role Hierarchy in GWO: Alpha, Beta, Delta, Omega

4.3.3.1 Encircling Prey

Grey wolves encircle prey prior to an attack. To simulate this behavior, the algorithm estimates the distance between a search agent and the prey, and updates the agent's position accordingly. The mathematical modelling of the encircling process is defined as:

$$\vec{D} = |\vec{C} \cdot \overrightarrow{X_p}(t) - \vec{X}(t)|$$

$$\vec{X}(t+1) = \overrightarrow{X_p}(t) - \vec{A} \cdot \vec{D}$$

where:

t → indicates the current iteration,

$\vec{A}$ and $\vec{C}$ are coefficient vectors,

$\overrightarrow{X_p}(t)$ → position vector of the prey,

$$\vec{A} = 2a \cdot \vec{r}_1 - a, \quad \vec{C} = 2 \cdot \vec{r}_2$$

$\vec{X}$ → indicates the position vector of a grey wolf.

4.3.3.2 Hunting:

Hunting involves cooperation among the pack's leading wolves. In GWO, the three best candidate solutions are labelled as alpha (α\alphaα), beta (β\betaβ), and delta (δ\deltaδ). All other wolves (omegas) update their positions based on the positions of these top three solutions.

The hunting behaviour is mathematically modelled as:

$$\vec{D}_\alpha = \left|\vec{C}_1 \cdot \vec{X}_\alpha - \vec{X}\right|, \quad \vec{X}_1 = \vec{X}_\alpha - \vec{A}_1 \cdot \vec{D}_\alpha$$

$$\vec{D}_\beta = \left|\vec{C}_2 \cdot \vec{X}_\beta - \vec{X}\right|, \quad \vec{X}_2 = \vec{X}_\beta - \vec{A}_2 \cdot \vec{D}_\beta$$

$$\vec{D}_\delta = \left|\vec{C}_3 \cdot \vec{X}_\delta - \vec{X}\right|, \quad \vec{X}_3 = \vec{X}_\delta - \vec{A}_3 \cdot \vec{D}_\delta$$

The updated position of a grey wolf is then computed as:

$$\vec{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3}$$

This collective influence guides the population toward promising regions of the search space, balancing exploration and exploitation.

4.3.3.3 Attacking Prey (Exploitation Phase)

As the search progresses and the prey becomes trapped, wolves converge to attack. Mathematically, this is achieved by reducing the value of parameter aaa, which in turn controls the magnitude of vector $A^{\to}$

When $|A^{\to}|<1$, wolves are drawn closer to the prey, favoring exploitation and convergence. Conversely, $|A^{\to}|>1$ leads to divergence, promoting exploration.

The controlled decrease of aaa from 2 to 0 during iterations ensures a smooth transition from exploration to exploitation, aiding in avoiding local optima and achieving global convergence.

4.3.3.4 Searching for Prey (Exploration Phase)

In the initial iterations, wolves are encouraged to search a broad space to discover optimal solutions. This is achieved when $|\vec{A}| > 1$, which causes the wolves to move away from the prey.
Additionally, the vector $\vec{C} \in [0,2]$ introduces randomness in the prey's perceived position, either emphasizing ($\vec{C}>1$) or deemphasizing ($\vec{C}<1$) its influence.

This stochastic behavior encourages a diverse and global search, which is critical in avoiding premature convergence and improving the algorithm's robustness.

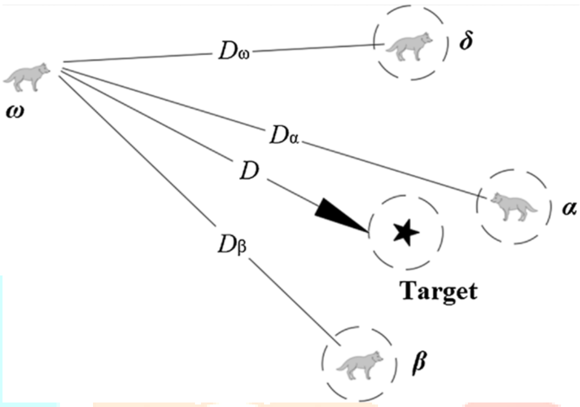
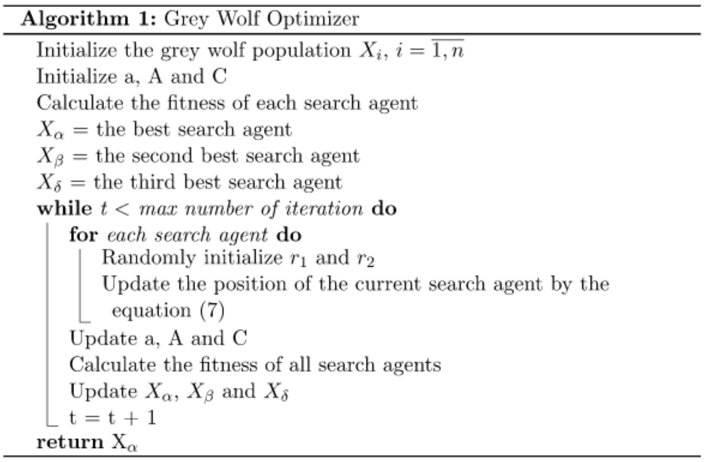


Fig 11: Prey Encircling and Hunting Mechanism in GWO



```
Algorithm 1: Grey Wolf Optimizer
Initialize the grey wolf population $X_i$, $i = \overline{1,n}$
Initialize a, A and C
Calculate the fitness of each search agent
$X_\alpha$ = the best search agent
$X_\beta$ = the second best search agent
$X_\delta$ = the third best search agent
while $t <$ max number of iteration do
    for each search agent do
        Randomly initialize $r_1$ and $r_2$
        Update the position of the current search agent by the
         equation (7)
    Update a, A and C
    Calculate the fitness of all search agents
    Update $X_\alpha$, $X_\beta$ and $X_\delta$
    t = t + 1
return $X_\alpha$
```

Fig. 12: Final Optimization and Convergence in GWO

5. Machine Learning model Training and Evaluation

In this study, the filtered feature set obtained from the proposed Ensemble Feature Selection technique is employed to train and evaluate the performance of five widely-used machine learning (ML) classifiers: Decision Tree (DT), Random Forest (RF), Support Vector Machine (SVM), eXtreme Gradient Boosting (XGBoost), and a shallow Neural Network (NN). The primary objective of this stage is to assess the classification accuracy and robustness of different ML models in distinguishing between benign and malicious network traffic, using the most relevant subset of features identified through ensemble-based selection.
The utilization of multiple ML algorithms ensures a comprehensive evaluation of the proposed feature selection method, enabling the analysis of its performance across a diverse range of learning paradigms. This approach guarantees that the effectiveness of the ensemble-selected features is not biased towards or dependent on any single classifier architecture, thereby validating the generalizability of the selection process.

Each classifier is trained using a fixed set of thirty features, significantly reduced from the original high-dimensional feature space. This dimensionality reduction not only enhances the computational efficiency but also mitigates the risk of overfitting, thereby improving the overall generalization capability of the models.

Additionally, training on a reduced yet informative feature set results in lightweight models, which are especially beneficial in real-time intrusion detection scenarios where processing time and resource consumption are critical factors.

The performance of each model is evaluated using standard classification metrics such as accuracy, precision, recall, F1-score, and Area Under the Receiver Operating Characteristic Curve (AUC-ROC). The results of these evaluations provide insights into the impact of the ensemble feature selection on model training and classification performance, offering empirical evidence in support of its utility in cybersecurity applications.

6. Performance Evaluation and Analysis

The performance of machine learning (ML) models must be rigorously assessed to ensure their effectiveness and reliability. In this research, we evaluate the performance of the proposed ML-based Ensemble Feature Selection approach using a comprehensive set of evaluation metrics. These include: **accuracy**, **precision**, **recall**, **F1-score**, **AUC-ROC score**, and the **confusion matrix**.

The **confusion matrix** is a foundational tool in binary classification tasks, providing insight into the model's prediction outcomes. It consists of the following components:

- **True Positive (TP):** The number of instances correctly predicted as positive.
- **True Negative (TN):** The number of instances correctly predicted as negative.
- **False Positive (FP):** The number of negative instances incorrectly predicted as positive.
- **False Negative (FN):** The number of positive instances incorrectly predicted as negative.

6.1 Accuracy:

Accuracy represents the proportion of correctly predicted instances among the total number of predictions. In the context of network traffic classification, it denotes the percentage of network traffic samples that are correctly identified by the model. It's a abecedarian metric used to estimate the overall performance of a classifier. The mathematical expression for accuracy is given by:

$$Accuracy = \frac{True\ Postive\ (TP) + True\ Negative(TN)}{Total\ Instances\ (TP + TN + FP + FN)}$$

6.2 Recall:

Recall, also referred to as **Sensitivity** or **True Positive Rate**, measures the proportion of actual positive instances that are correctly identified by the classifier. It is especially important in scenarios where failing to detect a positive instance (i.e., a **False Negative**) carries a significant cost or risk, such as in intrusion detection or medical diagnosis.

The mathematical expression for recall is given by:

$$Recall = \frac{True\ Positives\ (TP)}{True\ Positives\ (TP) + False\ Negatives\ (FN)}$$

6.3 Precision:

Precision quantifies the proportion of predicted positive instances that are truly positive. It evaluates the model's ability to avoid false positives by measuring how many of the predicted positive outcomes are actually correct. Precision is particularly important in applications where false alarms must be minimized, such as in spam detection or fraud detection.

$$Precision = \frac{True\ Positives\ (TP)}{True\ Positives\ (TP) + False\ Positives\ (FP)}$$

6.4 F1-Score:

The F1-score is the **harmonic mean** of perfection and recall. It provides a single metric that balances the trade-off between the two, especially in cases of **imbalanced datasets** where accuracy alone can be misleading. F1-score is particularly useful when both false positives and false negatives carry significant consequences.

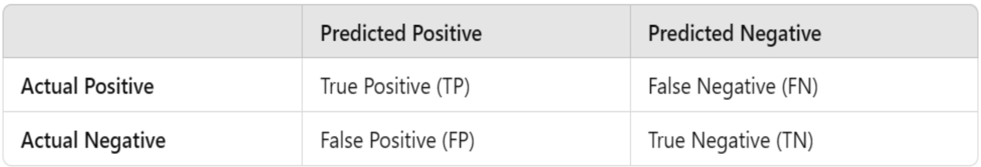
The F1-score is calculated using the following formula:

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

6.5 Confusion Matrix:

A confusion matrix is a powerful tool used in the field of machine learning to evaluate the performance of classification models. It is a tabular representation that provides insight into the number of correct and incorrect predictions made by the model, allowing for a more comprehensive understanding of its behavior and performance

Table-2 Confusion Matrix +ve and -ve Prediction



	Predicted Positive	Predicted Negative
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

6.6 Training and Testing Time

Training and testing times are critical metrics for evaluating the performance of machine learning models, particularly in time-sensitive applications such as intrusion detection systems.

- **Training Time:** This refers to the duration required to train the model on a given dataset. It is typically measured in seconds or minutes, depending on the model complexity and dataset size.
  $$\text{Training Time} = \text{End\_Time}_{\text{train}} - \text{Start\_Time}_{\text{train}}$$
- **Testing Time:** This represents the time taken to evaluate the model on unseen data, which includes data processing and prediction generation. It is also measured in seconds or minutes.
  $$\text{Testing Time} = \text{End\_Time}_{\text{test}} - \text{Start\_Time}_{\text{test}}$$

Efficiency is indicated by a reduced training time, allowing for faster model updates and improved adaptability to emerging threats. A rapid testing time is crucial for ensuring timely responses to potential risks without introducing significant latency.

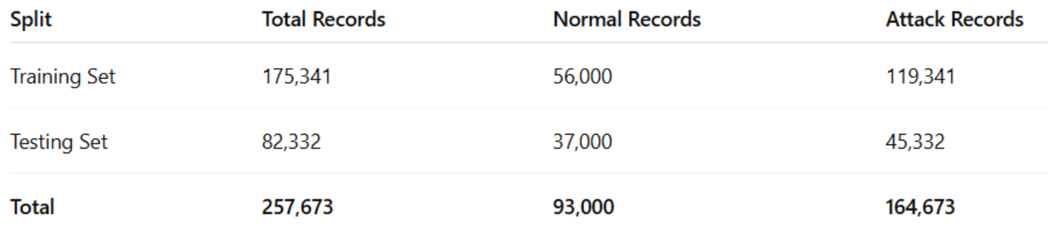
7. NB-15 Dataset:

The **NB-15 dataset** is a comprehensive network traffic dataset primarily designed for anomaly detection and intrusion detection system (IDS) evaluation. It is an extension and modification of the older KDDCup99 dataset, providing a more realistic representation of modern network traffic. The dataset includes a wide range of network features, such as protocol types (TCP, UDP, ICMP), connection duration, source and destination IP addresses and ports, traffic flow statistics, and various flags. These features are essential for distinguishing between normal and malicious network traffic. The dataset contains numerous attack categories, including Denial of Service (DoS), Distributed Denial of Service (DDoS), port scanning (Probe), and attacks such as Remote to Local (R2L) and User to Root (U2R). It provides millions of records, making it a suitable choice for developing and testing machine learning models for network security. The NB-15 dataset is valuable for researchers working on intrusion detection, as it offers a diverse set of both normal traffic and attack scenarios that help in training and evaluating detection algorithms. Preprocessing techniques like normalization and categorical data conversion are often necessary to prepare the data for machine learning applications. The NB-15 dataset is widely used in the cybersecurity community for benchmarking algorithms used in detecting network-based attacks.

Attribute	**Description**
Dataset Name	NB-15
Purpose	Anomaly detection and intrusion detection system (IDS) evaluation.
Size	Large dataset with millions of records, including both normal and malicious network traffic.
Features	Includes features such as IP addresses, protocol type, connection duration, source/destination ports, flags, and traffic count.
Attack Categories	Denial of Service (DoS), Distributed Denial of Service (DDoS), Probe (port scanning), Remote to Local (R2L), User to Root (U2R).
Preprocessing Needs	Data normalization, handling of missing values, and conversion of categorical data to numerical format.

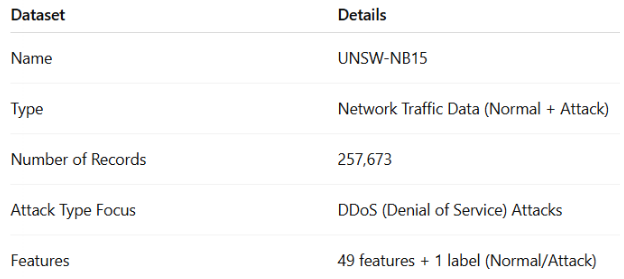
Attribute	Description
Applications	Useful for evaluating machine learning models for network anomaly detection and intrusion detection.
Data Type	Network traffic data with labeled attack types.

Table 3: UNSW-NB15 Dataset Set Records



Split	Total Records	Normal Records	Attack Records
Training Set	175,341	56,000	119,341
Testing Set	82,332	37,000	45,332
Total	**257,673**	**93,000**	**164,673**

Table 4 : Dataset Selection for DoS, DDoS, and Benign Traffic



Dataset	Details
Name	UNSW-NB15
Type	Network Traffic Data (Normal + Attack)
Number of Records	257,673
Attack Type Focus	DDoS (Denial of Service) Attacks
Features	49 features + 1 label (Normal/Attack)

Results:

Selected Features using SHO

[‘proto’, ‘spkt’, ‘dpkts’, ‘sbytes’, ‘dloss’, ‘sinpkt’, ‘response_body_len’, ‘ct_dst_ltm’, ‘ct_dst_sport_ltm’, ‘ct_src_ltm’, ‘ct_srv_dst’, ‘attack_cat’]

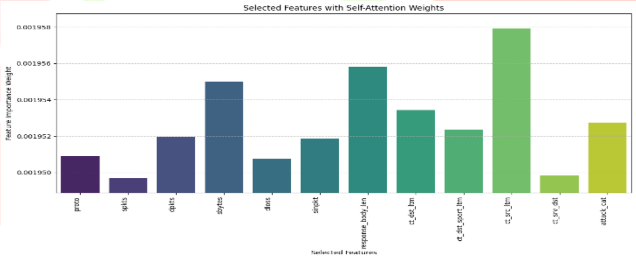


Fig13: Selected Features of SHO

Selected Features using PO

['spkts', 'dpkts', 'sbytes', 'sttl', 'sloss', 'djit', 'stcpb', 'dwin', 'tcprtt', 'synack', 'dmean', 'response_body_len', 'ct_srv_src', 'ct_dst_ltm', 'ct_src_dport_ltm', 'ct_dst_sport_ltm', 'ct_ftp_cmd', 'ct_src_ltm']

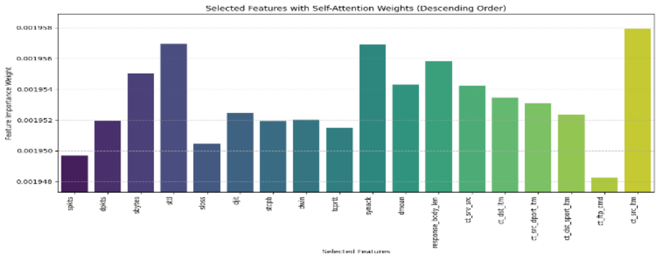


Fig14 : Selected Features of PO

Selected Features using GWO

['dur', 'state', 'dpkts', 'sbytes', 'dttl', 'sload', 'sloss', 'dinpkt', 'sjit', 'stcpb', 'dtcpb', 'dwin', 'smean', 'dmean', 'response_body_len', 'ct_state_ttl', 'ct_src_dport_ltm', 'ct_dst_sport_ltm', 'ct_ftp_cmd', 'ct_flw_http_mthd', 'ct_srv_dst']

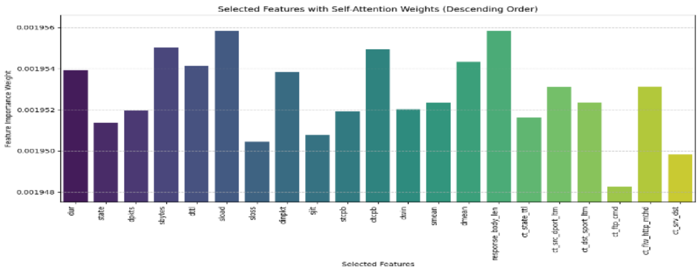


Fig 15: Selected Features Of GWO

References :

[1] Pooja Kumari, Ankit Kumar Jain .,"A comprehensive study of DDoS attacks over IoT network and their countermeasures" .,Computers and Security 127 (2023) 103096.

[2] Bindu Bala and Sunny Behal .,"AI techniques for IoT-based DDoS attack detection: Taxonomies, comprehensive review and research challenges" .,Computer Science Review 52 (2024) 100631.

[3] Saurav Kumar, Ajit kumar Keshri .,"An effective DDoS attack mitigation strategy for IoT using an optimization-based adaptive security model" .,Knowledge-Based Systems 299 (2024) 112052.

[4] Monika Roopak, Gui Yun Tian, Jonathon Chambers .,"An Intrusion Detection System Against DDoS Attacks in IoT Networks" .,IEEE Xplore (2020).

[5] Vanlalruata Hnamte, Ashfaq Ahmad Najar, Hong Nhung-Nguyen, Jamal Hussain, Manohar Naik Sugali .,"DDoS attack detection and mitigation using deep neural network in SDNenvironment" .,Computers and Security 138 (2024) 103661.

[6] Kavitha D, Ramalakshmi R .,"Machine learning-based DDOS attack detection and mitigation in SDNs for IoT environments" .,Journal of the Franklin Institute 361 (2024) 107197.

[7] Usman Haruna Garba, Adel N. Toosi, Muhammad Fermi Pasha and Suleman Khan .,"SDN-based detection and mitigation of DDoS attacks on smart homes" .,Computer Communications 221 (2024) (pp 29–41).

[8] Gaurav Dhiman and Vijay Kumar .,"Spotted hyena optimizer: A novel bio-inspired based metaheuristic technique for engineering applications" ., Advances in Engineering Software 114 (2017) (pp. 48–70).

[9] Youfa Fu1, Dan Liu1, Jiadui Chen1, Ling He1.,"Secretary bird optimization algorithm: a new metaheuristic for solving global optimization problems"., Artifcial Intelligence Review (2024) (pp. 57-123).

[10] Hema Banati, Richa Sharma, and Asha Yadav .,"Binary Peacock Algorithm: A Novel Metaheuristic Approach for Feature Selection"., SpringerLink (2024), Vol 41,(pp. 216–244).

[11] Junbo Lian, Guohua Hui, Ling Ma, Ting Zhu, Xincan Wu, Ali Asghar Heidari, Yi Chen, Huiling Chen .,"Parrot optimizer: Algorithm and applications to medical problems" .,Computers in Biology and Medicine 172 (2024) 108064.

[12] Anushiya and V.S. Lavanya .,"A new deep-learning with swarm based feature selection for intelligent intrusion detection for the Internet of things" .,Measurement: Sensors 26 (2023) 100700.

[13] Fizza Rizvi, Ravi Sharma, Nonita Sharma, Manik Rakhra, Arwa N. Aledaily, Wattana Viriyasitavat, Kusum Yadav, Gaurav Dhiman and Amandeep Kaur .,"An evolutionary KNN model for DDoS assault detection using genetic algorithm based optimization" .,SpringerNature Link (2024) Volume 83, (pp 83005–83028).

[14] O. Pandithurai, C. Venkataiah, Shrikant Tiwari and N. Ramanjaneyulu .,"DDoS attack prediction using a honey badger optimization algorithm based feature selection and Bi-LSTM in cloud environment" .,Expert Systems With Applications 241 (2024) 122544.

[15] Ahmed Ahmim, Faiz Maazouzi, Marwa Ahmim, Sarra Namane, Imed Ben Dhaou .,"Distributed Denial of Service Attack Detection for the Internet of Things Using Hybrid Deep Learning Model" .,IEEE Xplore (2023) Volume 11 119862.

[16] Xuan-Ha Nguyen, Kim-Hung Le .,"Robust detection of unknown DoS/DDoS attacks in IoT networks using a hybrid learning model" .,Internet of Things 23 (2023) 100851.

[17] Mohamed Aly Bouke, Azizol Abdullah, Sameer Hamoud ALshatebi, Mohd Taufik Abdullah, Hayate El Atigh .,"An intelligent DDoS attack detection tree-based model using Gini index feature selection method" .,Elsevier : Microprocessors and Microsystems 98 (2023) 104823.

[18] Md. Alamgir Hossain, Md. Saiful Islam .,"Enhancing DDoS attack detection with hybrid feature selection and ensemble-based classifier: A promising solution for robust cybersecurity" ., Elsevier : Measurement: Sensors 32 (2024) 101037.

[19]Anupama Mishra, Neena Gupta and Brij B. Gupta .,"Defensive mechanism against DDoS attack based on feature selection and multi-classifier algorithms" .,SpringerNature Link (2023) Volume 82, Volume 82, pages 229–244