Journal of Information Systems Engineering and Management
2025, 10(36s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

An Adaptive Optimization Algorithm for Features Selection to Enhance the Detection of Intrusion Attacks Over Networks

Ola Ali Obead[1,2], Hakem Beitollahi[2]
[1]*Department of Information Networks, College of Information Technology, University of Babylon. Iraq*
[2]*School of Computer Engineering, Iran University of Science and Technology. Tehran, Iran*
coerresponding auther" "Ola334805@gmail.com"

ARTICLE INFO

Received: 01 Mar 2025

Aceepted: 17 Apr 2025

Published: 20 Apr 2025

ABSTRACT

This study looks at the security issues with DNS over HTTPS (DoH), which hides DNS traffic for better privacy but also allows hackers to hide their activities. Introduction: DDoS attacks harm networks by flooding servers with too much traffic, while regular DNS can be easily intercepted. Objectives: The research wants to build a better system to detect attacks in encrypted DNS by using fewer but more important data features, creating a combined optimization method, and testing how well it identifies attacks. Methods: Using real-world encrypted DNS traffic data, the study combines two search methods (Particle Swarm and Grey Wolf) to find the best features for machine learning models. Results: The combined search method worked better than either method alone in tests, and for attack detection, Random Forest achieved about 95% accuracy, while Naive Bayes improved greatly from 66% to 77% when enhanced with the new technique. Conclusions: The combined approach helps find better features for detecting attacks, especially improving weaker detection methods when looking for threats in encrypted DNS traffic.

Keywords: Feature Selection, Wrapper Model, Feature Engineering, Intrusion Detection System, Metaheuristic Algorithms.

INTRODUCTION

Distributed Denial-of-Service (DDoS) attacks represent a significant threat to network systems by overwhelming targeted servers with vast amounts of traffic[1]. In a typical DDoS attack, an attacker, known as the botmaster, controls a command-and-control server to send instructions to a network of compromised devices (botnet)[2]. These devices, which may include hundreds or even thousands of infected hosts, collectively transmit malicious traffic to the victim's server[3]. The immense volume of traffic overloads the server, making it incapable of processing legitimate requests. This disruption can severely affect online services, resulting in downtime, financial losses, and reputational harm. Figure 1 illustrates the mechanism employed by DDoS attacks against victims.

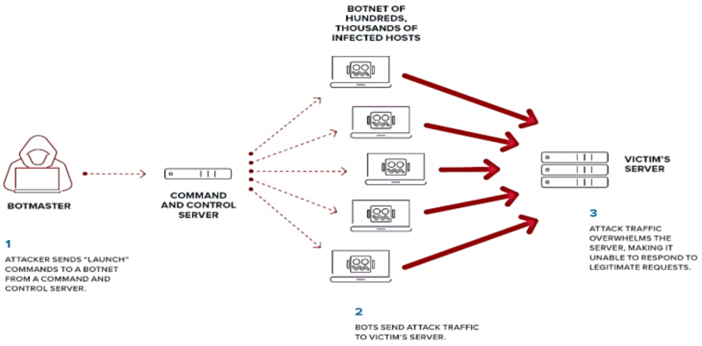

BOTNET OF HUNDREDS, THOUSANDS OF INFECTED HOSTS
BOTMASTER
COMMAND AND CONTROL SERVER
VICTIM'S SERVER
1
ATTACKER SENDS "LAUNCH" COMMANDS TO A BOTNET FROM A COMMAND AND CONTROL SERVER.
2
BOTS SEND ATTACK TRAFFIC TO VICTIM'S SERVER.
3
ATTACK TRAFFIC OVERWHELMS THE SERVER, MAKING IT UNABLE TO RESPOND TO LEGITIMATE REQUESTS.

Figure 1: Mechanism of DDoS attack

The Domain Name System (DNS) plays a foundational role in Internet communication by translating human-readable domain names into machine-recognizable IP addresses [4]. Unfortunately, traditional DNS works over unencrypted channels and is extremely vulnerable to interception, manipulation, and monitoring by malicious actors or unauthorized intermediaries. To counter the increased concern for user privacy and security, a new protocol has been introduced called DNS over HTTPS (DoH), which encapsulates DNS queries in HTTPS traffic and subsequently encrypts them, preventing eavesdropping tampering [5]. DoH massively improves privacy by hiding DNS traffic inside regular HTTPS flows, but at the same time presents a significant problem for network defenders. Encrypted Angel: Over the covering in the encrypted DNS over HTTPS (DoH) sessions, and command-and-control (C&C) communications, data exfiltration, Botnet coordination, etc., malicious acts are likely performed in covers [6]. The encryption that protects users from surveillance also prevents security systems from performing deep packet inspection or traditional signature-based detection. As a result, attackers can exploit encrypted DNS traffic to bypass detection mechanisms, posing a serious threat to both individual users and corporate networks [3].

To address DoH's dual-edged nature, intelligent detection systems that do not rely on decrypting packet contents but instead utilize traffic-level features and behavioral patterns are critical[4-7]. This calls for robust machine learning (ML)- -based solutions that can learn to differentiate between benign and malicious encrypted traffic based on statistical and time-based characteristics. However, these solutions face the challenge of high-dimensional feature spaces, where irrelevant or redundant features can degrade performance, increase computational costs, and reduce model generalizability[7].

Feature selection, a critical preprocessing step within ML pipelines, involves identifying and keeping only the most important relevant attributes, removing noise. Metaheuristic optimization algorithms have emerged as a viable solution capable of effectively traversing the complex and nonlinear search space to identify near-optimal feature subsets. Out of many metaheuristics, Particle Swarm Optimization (PSO) and Grey Wolf Optimiser (GWO) are the most popular as they are simple and effective. While PSO mimics the flocking behavior of birds or schools of fishes, the leadership hierarchy and hunting mechanisms of grey wolf's help form GWO. Both previous algorithms have their limitations; PSO leads to premature convergence, while GWO can experience slow convergence or stagnation in local optima.

This paper proposes a hybrid feature selection framework that combines the exploration strength of PSO and the exploitation capability of GWO to enhance search efficiency and reliability. By integrating these algorithms, the hybrid model seeks to overcome stagnation, improve convergence, and identify high-quality feature subsets for training machine learning classifiers. The proposed system is evaluated using the CIRA-CIC-DoHBrw-2020 dataset, which contains labeled samples of encrypted DNS traffic captured from real-world browsing scenarios across multiple browsers.

OBJECTIVES

The primary objective of this research is to develop an intrusion detection model that classifies traffic over DNS. It is designed to remain lightweight perfromace to avoid incurring significant overhead. The proposed model integrates hybrid optimization techniques (PSO and GWO) with machine learning algorithms in the classification process. The study's objectives can be summarized as follows:

- To reduce data complexity by selecting only the most important features.
- To design a hybrid feature selection method that combines PSO and GWO.
- To train and test a machine learning model that uses the selected features.
- To evaluate the system's ability to detect and classify different types of attacks accurately.

RELATED WORKS

This section provides an overview of cybersecurity risks and the procedures used to detect them.

Dwivedi et al. [8] proposed an Information Gain-based Intrusion Detection System (IGIDS) that integrates a filter-based feature selection technique with machine learning algorithms. IGIDS selects relevant features from IDS datasets to distinguish low-speed DDoS attacks. Filter-based methods assume features are independent and are often inaccurate in real-world datasets, especially for complex tasks like IDS. Ignoring feature dependencies can lead to sub-optimal feature selection and reduced detection accuracy.

In [9], the authors introduced a new approach to spot DRDoS attacks by analyzing DNS traffic patterns. Instead of passively sifting through data, their system actively identifies the most telling clues (features) upfront. To do this, they upgraded existing optimization methods to pinpoint critical signals while trimming away unnecessary noise in the data. They then tested well-known machine learning tools—like k-NN, random forest, and SVM—to classify whether traffic was malicious based on these streamlined features.

Krishna et al. [10] introduced a hybrid optimization approach combining the Metaheuristic Lion Optimization Algorithm and the Firefly Optimization Algorithm (ML-F) for detecting Distributed Denial of Service (DDoS) and other attacks on IoT devices. In preprocessing data to remove noise and missing data, it applies recursive feature elimination for feature extraction and employs a random forest classifier for attack classification. The algorithm improved the number of features but did not resolve the overlap and imbalance data.

The authors in [11] proposed a clustering method for classifying the DNS traffic to intrusion and normal. The proposed model reduces the number of producing clustering from the first stage based on frequent Euclidean distance and threshold. The limitation of this work is that the authors reused the same metrics across distributed data on clusters and, in an evolving process, made the clustering algorithm rely heavily on the threshold.

METHODS

This section discusses the materials and methods employed that are related to the proposed model.

A- Dataset Description

The CIRA-CIC-DoHBrw-2020 dataset is a publicly available dataset used for research in network intrusion detection, especially focusing on encrypted DNS traffic, such as DNS over HTTPS (DoH) [12]. The dataset contains traffic collected from different web browsers, such as Firefox and Chrome. Table 1 describes the CIRA-CIC-DoHBrw-2020 dataset

Table 1: Dataset description [12]

Feature	Description
Total Samples	Over 600,000 records
Traffic Type	DNS over HTTPS (DoH) and regular DNS
Browsers Used	Firefox, Chrome
Labeling	DoH and Normal
Number of Features	Over 80 network-related features

B- Prerequisite:

This section introduces the prerequisites for understanding the proposed model and its related methodologies.

1- Distributed Denial-of-Service (DDoS)

A DDoS attack occurs when numerous compromised hosts direct excessive or malicious traffic toward a single target, exhausting its resources[13]. Unlike a denial of service (DoS) attack, which originates from a sole source, DdoS attacks intensify their effects by employing multiple geographically distributed agents[14]. This distributed nature significantly complicates detecting, containing, and tracing the source.

Key elements of DDoS attacks include[15]:

- Botnets: Collections of infected devices (computers, IoT devices, servers) under an attacker's control. These devices can be activated concurrently to launch coordinated attacks.
- Command and Control (C&C) Servers: Attackers use centralized systems or hidden communication layers to issue commands and coordinate botnet activity.
- Attack Vectors: Common approaches include volumetric (flood) attacks that saturate bandwidth, protocol attacks targeting connection state tables (e.g., SYN floods), and application-layer attacks designed to exploit specific features of web or application servers (e.g., HTTP GET/POST floods)

2- Metaheuristic Algorithms

Metaheuristic utilization encompasses methods designed to address complex utilization problems[16]. Optimization is extensively employed across various fields, including engineering design[17], economics[18], travel planning[18], and internet routing[19], to efficiently allocate limited resources such as time, materials, and finances. Real-world utilization challenges are often present in nonlinear, multimodal conditions with intricate constraints [16]. These issues typically involve conflicting objectives, making pursuing an ideal or even near-optimal solution difficult. Consequently, effective utilization methods are crucial. Optimization algorithms are utilization into two groups based on their utilization of derivative information[20]: gradient-based and derivative-free. Gradient-based methods, including hill-climbing, employ derivatives to identify solutions swiftly and are utilized for efficiency. Derivative-free methods, like the Nelder-Mead simplex algorithm, depend solely on evaluating function values, making them apt for functions that are discontinuous or costly to differentiate. Another method of classifying utilization algorithms is distinguishing between trajectory-based and population-based approaches[21]. Trajectory-based algorithms utilize a single solution that progresses step-by-step through the search process.

Examples of Metaheuristic Algorithms are genetic Algorithms, Particle Swarm Optimization (PSO), Gray Wolf Optimization (GWO), Bat Algorithms (BA), and Ant Colony Optimization (ACO).

Figure 2 illustrates the search mechanism used in metaheuristic algorithms.

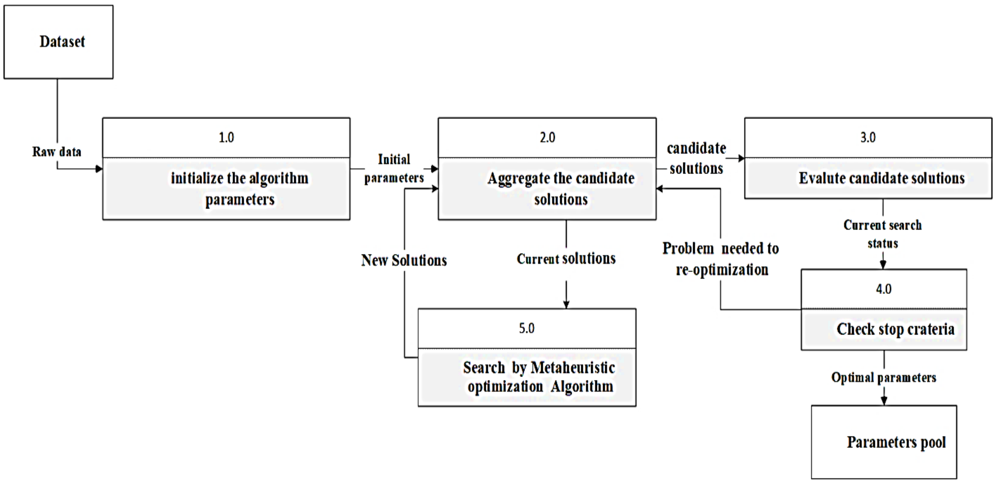

Dataset
Raw data
1.0
initialize the algorithm parameters
Initial parameters
2.0
Aggregate the candidate solutions
candidate solutions
3.0
Evalute candidate solutions
Current search status
4.0
Check stop crateria
Problem needed to re-optimization
New Solutions
Current solutions
5.0
Search by Metaheuristic optimization Algorithm
Optimal parameters
Parameters pool

Figure 2: Searching engine of metaheuristics algorithms

Metaheuristic optimization methods face several limitations that can hinder their effectiveness. One key issue is their tendency to converge to local optima, especially in the case of local search algorithms, which often require external mechanisms like random restarts to escape such traps [19].

Another challenge of metaheuristics is stagnation. It means the algorithm stops making meaningful progress toward better solutions. The stagnation offends happen dute lake in the exploration and exploitation of algorithms.

Furthermore, metaheuristics do not always ensure consistent success or optimal results across different problem domains. The hybrid approaches potentially reduce the probability of happens in stagnation during searching prograss [16].

Figure 3 illustrates the stagnation in searching progress of metaheuristics algorithms.

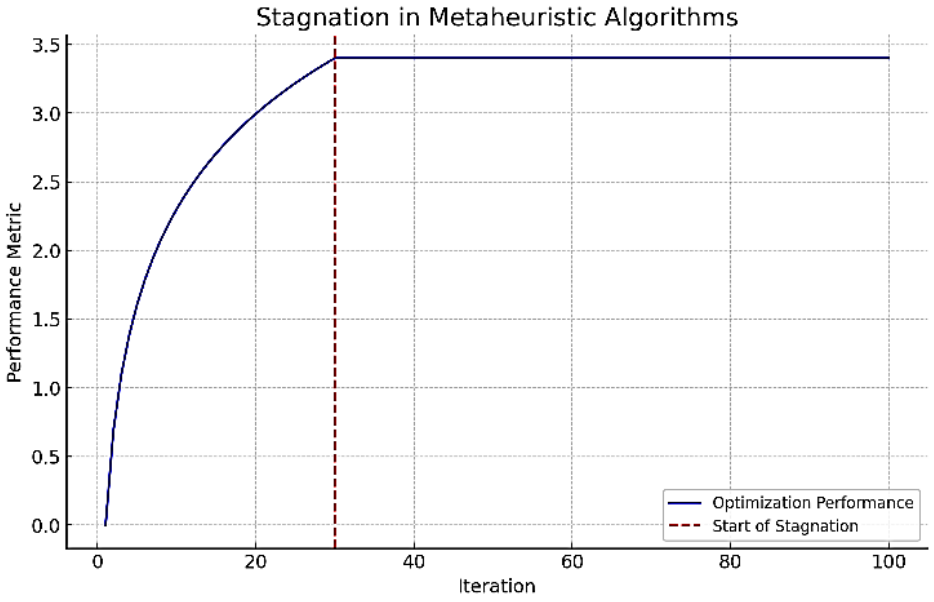


Figure 3: Stagnation in metaheuristics algorithms

3- Particle Swarm Optimization (PSO) algorithm

Particle Swarm Optimization (PSO) is a powerful metaheuristic algorithm inspired by the collective behavior of natural swarms like bird flocks and fish schools[18]. Imagine flocks of birds searching for food; they share information about promising areas, gradually converging on the best location as individuals follow each other and react to their discoveries. PSO mimics this behavior to solve optimization problems, guiding a swarm of “particles” toward the optimum solution within a search space[22].

Inspired by bird flocks, Particle Swarm Optimization (PSO) mimics their collective search behavior to solve optimization problems[21]. A swarm of particles (solutions) moves through a search space, guided by their own best positions and the best position found by the entire swarm. Each particle adjusts its direction based on these, gradually converging towards the optimal solution. This iterative process balances exploring new areas and exploiting promising paths, leading to effective optimization[16].

The advantage of PSO is its ease of implementation, requiring minimal setup and coding[23]. This efficiency extends to its search process, as PSO often converges quickly to good solutions compared to other optimization methods. Its robustness allows it to tackle complex problems with many variables, unfazed by noise or high dimensionality. Additionally, PSO seamlessly leverages parallel computing systems, further boosting its power for demanding tasks[24].

4- Grey Wolf Optimizer (GWO) algorithm

Grey Wolf Optimizer (GWO) is a bio-inspired metaheuristic algorithm based on grey wolves’ social hierarchy and hunting strategies[16]. Inspired by how wolves collaborate to hunt prey, GWO effectively searches for optimal solutions to complex optimization problems. Grey Wolf Optimizer (GWO) mimics the collaborative hunting strategies of wolves to tackle complex optimization problems. Alpha, Beta, and Delta wolves represent potential solutions, guiding their pack (Omega wolves) toward the optimal prey. Figure 4 shows the wolves’ hierarchy dominant strict society from the best particle (alpha) to the worst particle (Omega).

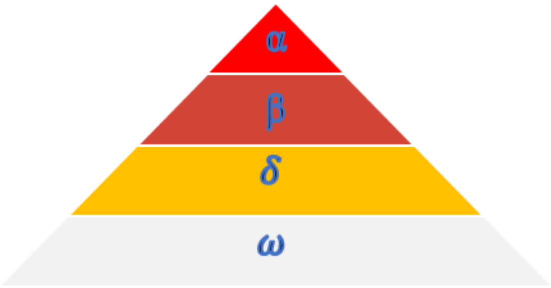


Figure 4: Hierarchy of grey wolf based on getting the best solutions from top to down

By encircling, attacking, and exploring the search space, GWO efficiently converges on good solutions. This translates to real-world benefits: easy implementation, fast convergence, the ability to escape local traps, and handling complex problems with ease. Like wolves working together, GWO demonstrates the power of collaboration and adaptation in finding optimal solutions.

5- Feature selection by metaheuristic algorithms

Feature selection is crucial in data analytics, helping identify the most informative features from a dataset to improve model performance and interpretability [25-27]. Wrapper methods approach this by directly evaluating the impact of feature subsets on a chosen learning model. By employing optimization algorithms, these methods can efficiently navigate the vast space of possible feature combinations to select the optimal subset [21]. Figure 5 shows the Wrapper model framework used to select features.

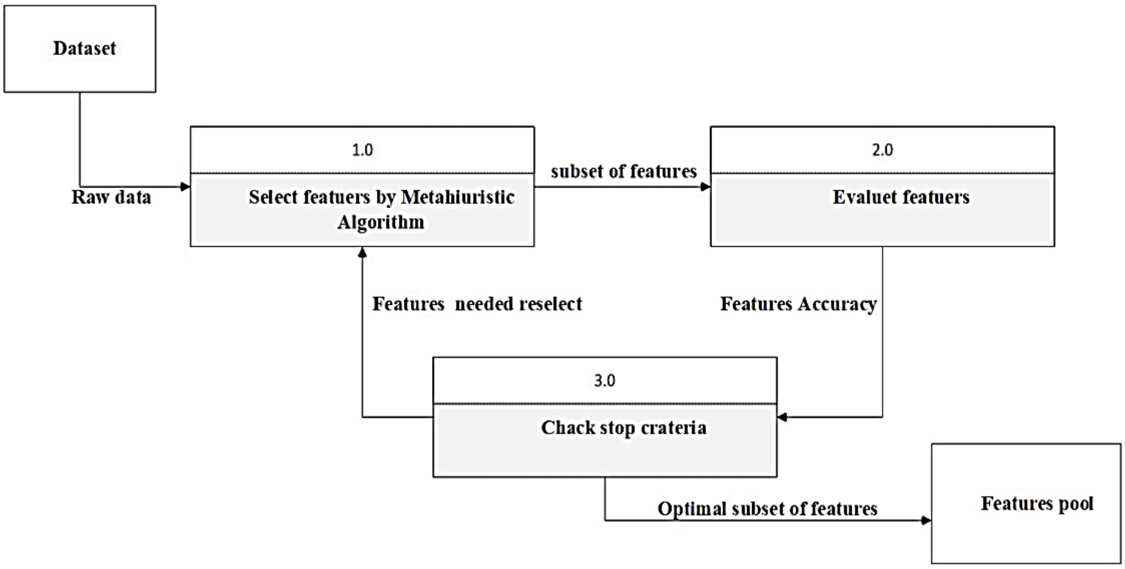


Figure 5: Feature selection based-Wrapper Framework [16]

6- Time Complexity of Machine Learning with Feature Selection

Feature selection is a crucial preprocessing step in machine learning, where a subset of the most relevant features (variables, predictors) is selected for model construction[28]. By reducing the dimensionality of the dataset, feature selection aims to improve model performance, enhance interpretability, and reduce computational costs [29-31]. This section explores how feature selection impacts the time complexity of machine learning algorithms, supported by mathematical evidence and proofs.

The time complexity of a machine learning algorithm, which refers to the computational resources required for training and predicting, is typically expressed as a function of the number of samples n and the number of features d. High-dimensional datasets (large d) increase computational demands due to the “curse of dimensionality.” By applying feature selection to reduce d to d' (where $d' < d$), computational requirements of machine learning algorithms can be significantly decreased, impacting both training and prediction times.

Several common machine learning algorithms illustrate how feature selection reduces time complexity. For linear models (e.g., linear regression, logistic regression) trained using gradient descent, each iteration involves computing gradients for each feature, with a time complexity per iteration of $T_{\text{linear}} = O(n \times d)$. When feature selection reduces the number of features to d', the time complexity becomes $T'_{\text{linear}} = O(n \times d')$, and the computational cost decreases proportionally with the number of selected features: $\frac{T'_{\text{linear}}}{T_{\text{linear}}} = \frac{d'}{d}$. Consider an intrusion detection dataset with $n = 100{,}000$ samples and $d = 100$ features. If feature selection reduces features to $d' = 20$, training time for logistic regression was initially calculated as $T_{\text{original}} = O(100{,}000 \times 100) = O(10{,}000{,}000)$ reduces to $T_{\text{reduced}} = O(100{,}000 \times 20) = O(2{,}000{,}000)$, achieving an 80% reduction. Similarly, for a neural network with 50 hidden neurons, feature selection changes time complexity from $T_{\text{original}} = O(100{,}000 \times (100 \times 50 + 50 \times o))$ to $T_{\text{reduced}} = O(100{,}000 \times (20 \times 50 + 50 \times o))$, approximating a reduction factor of 0.2 and an 80% computational cost decrease. While feature selection offers significant benefits, it is essential to consider its overhead costs, potential trade-offs in model accuracy, and scalability, as some algorithms may struggle with very high-dimensional data.

C- Proposed model

The proposed model for intelligent detection the malicious traffic over DNS. By carefully selecting features using Particle Swarm Optimization (PSO) and Grey Wolf Optimization (GWO), the model effectively reduces the dimensionality of the dataset. The flow diagram in Figure 6 illustrates the pipeline, starting from raw data acquisition to final attack type prediction.

1. Preprocessing

This step consists of four basic steps that result in preparing the data to be worked on using artificial intelligence algorithms

- **Remove redundancy**

Identifying and removing duplicate or near-duplicate records within the dataset is essential. Redundant entries can distort the model's understanding of the data by overemphasizing specific patterns, leading to skewed or biased results. When duplicates remain, the model may seem to perform better than it does, as it effectively "learns" the same information multiple times. By systematically eliminating these redundancies, we ensure that each observation carries equal weight, thereby providing a cleaner, more representative dataset and supporting a fairer, more accurate model.

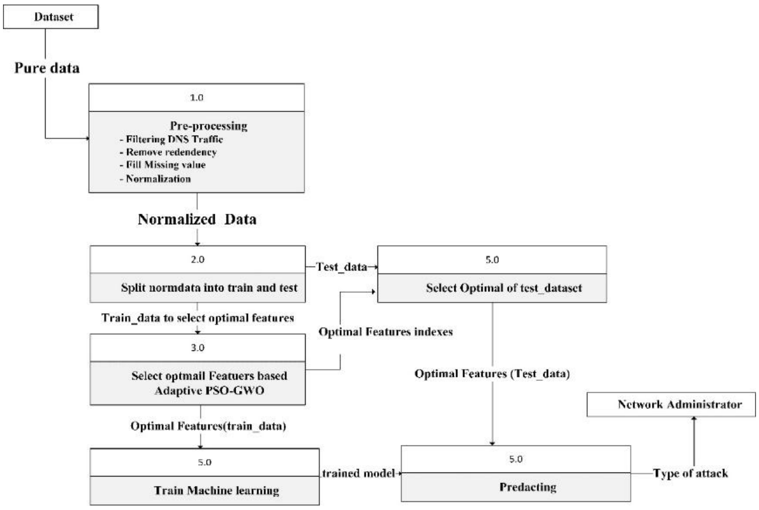


Figure 6: The main architecture of the proposed model

- **Fill Missing value**

Addressing missing values is crucial for maintaining the integrity of the dataset. Depending on the nature and extent of the missing data, you may choose to either remove records with large portions of missing information or impute (estimate) those values. A common and straightforward imputation method is mean-based imputation. Suppose a feature ***X*** has ***n*** valid (non-missing) observations: $x_1, x_2, \dots, x_n$. The mean $\bar{x}$ of these valid observations is computed as shown in equation 3.1:

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

Error! No text of specified style in document.1

For each missing entry x_m in the same feature, you then replace it with $\bar{x}$ as shown in equation 2

$$x_m = \bar{x} \qquad 2$$

This ensures the dataset remains the same size while providing a reasonable central value estimate for missing observations. However, alternative methods (e.g., median or mode imputation, regression-based imputation, or advanced model-based approaches) may be more appropriate if missing values comprise a substantial portion of the dataset or if the data distribution is skewed. In cases where the missing data is extreme and may introduce bias even after imputation, removing those records entirely can be more effective, provided the removal does not compromise the dataset's representativeness.

- **Normalization**

Normalization using the standard deviation method is essential in preparing numerical features for machine learning models. This method, often called z-score normalization, standardizes the data by centering it around zero with a standard deviation of one. The equation 3 used for z-score normalization:

$$x_{norm} = \frac{x - \mu}{\sigma} \qquad 3$$

Where x is the original value of the feature, μ is the mean of the feature values, and σ is the standard deviation of the feature values.

2. Split into train and test

The dataset must be divided into training, validation, and test sets. To assess the model's performance. The proposed model divides the dataset into:

- 70% training.
- 10% validation.
- 20% testing.

3. Select features

In the feature selection step, features are selected based on the proposed Adptieve_PSO_GWO optimization algorithm. Figure 7 presents the overall flow of the proposed hybrid feature selection method.

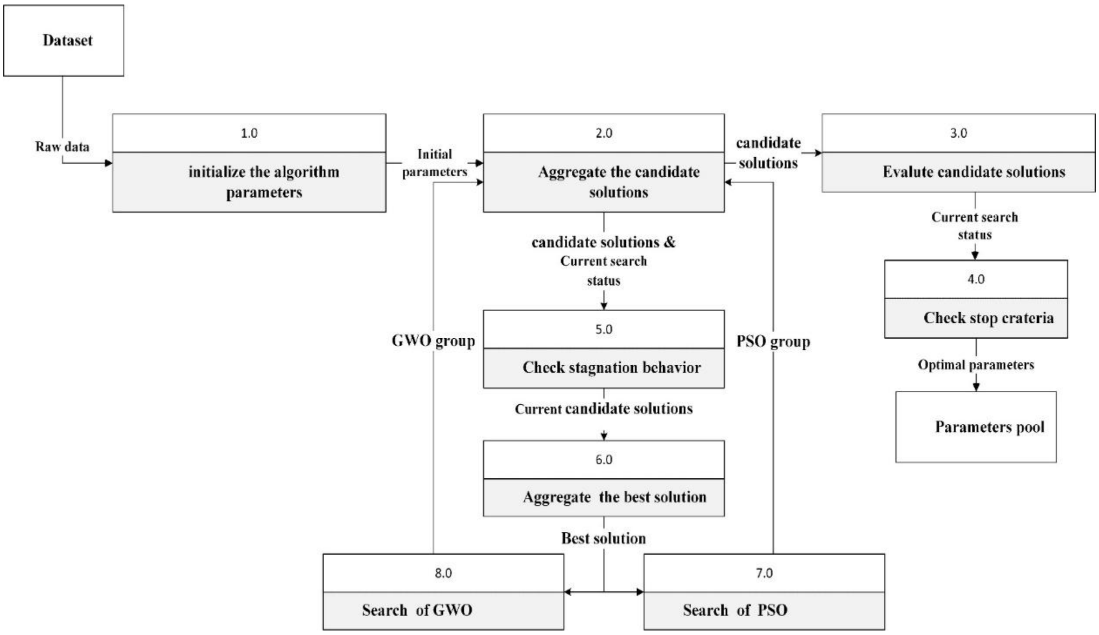

Dataset
Raw data
1.0
initialize the algorithm parameters
Initial parameters
2.0
Aggregate the candidate solutions
candidate solutions
3.0
Evalute candidate solutions
Current search status
4.0
Check stop crateria
Optimal parameters
Parameters pool
candidate solutions & Current search status
5.0
Check stagnation behavior
GWO group
PSO group
Current candidate solutions
6.0
Aggregate the best solution
Best solution
8.0
Search of GWO
7.0
Search of PSO

Figure 7: the proposed Adaptive_PSO_GWO framework

The goal is to search effectively for an optimal subset of features while avoiding stagnation or premature convergence. Each step of the diagram is described below.

- **Initialize the Algorithm Parameters (Process 1.0)**: The initialization parameters include population sizes (for PSO or GWO), learning coefficients (for PSO), and control parameters (for GWO).
- **Aggregate the Candidate Solutions (Process 2.0)**: Aggregate each algorithm (PSO and GWO) and propose a pool of candidate solutions—i.e., potential subsets of features.
- **Evaluate Candidate Solutions (Process 3.0)**: In this step, each candidate feature subset is evaluated using a chosen fitness function. The fitness function typically involves training a classifier on the selected features (for example, using a decision tree or support vector machine) and measuring the classification performance (accuracy, F1-score, etc.). The fitness values guide both algorithms in identifying more promising feature subsets.
- **Check Stop Criteria (Process 4.0)**: After evaluating the current generation of candidate subsets, the system checks whether a stopping criterion has been met. Common stopping criteria may include a maximum number of iterations, a convergence threshold in fitness values, or a plateau in improvement over several iterations. If the criterion is satisfied, the process halts and exports the best feature subset to the parameters pool (the final solution). If not, the method proceeds to refine the search.
- **Check Stagnation Behavior (Process 5.0)**: One unique aspect of this approach is its explicit monitoring of stagnation. If either the PSO or GWO subprocess shows little or no improvement for a specified number of iterations, the system recognizes that it may be stuck in a local optimum. Consequently, the method can trigger a re-initialization or parameter adjustment step, which helps guide the search away from local traps and maintain diverse candidate solutions.
- **Aggregate the Best Solution (Process 6.0)**: At this stage, the best current solution (i.e., feature subset with the highest fitness so far) is identified and stored. This step helps the framework keep track of the top-performing subset across iterations, ensuring that the best-known features are never lost, even if the algorithms diverge temporarily when escaping local optima.
- **Search of PSO (Step 7.0) and Search of GWO (Process 8.0)**: Next, the Particle Swarm Optimization and Grey Wolf Optimization groups continue their respective searches.

  ✓ **PSO:** Particles (candidate solutions) update their positions (feature subsets) based on global and local best solutions, potentially leading to better subsets in each iteration.

- ✓ **GWO:** Wolves (candidate solutions) mimic the social hierarchy of a grey wolf pack to move toward the current “alpha” (best solution), refining the subset of features in search of further performance gains.

- **Parameters Pool**: a pool as the final feature subset.

4. Applying Feature Selection to Test Data

The selected optimal feature indices from the training data are applied to the test dataset.

5. Model Training

A machine learning classifier is trained using the optimal features from the training set.

6. Prediction

The trained model is then used to predict the type of attack present in the test data. The output of this step is an accurate classification of DNS traffic, which can be forwarded to a network administrator for further action or alerting.

RESULTS

This section discusses the experiments regarding the proposed optimization algorithm, clustering, and classification models.

1. Evaluate the proposed optimization model in CEC2005

The proposed Adaptive_PSO_GWO is compared with two standard metaheuristic optimization algorithms (PSO and GWO).

Table 4.1 summarizes the performance of the proposed Adaptive_PSO_GWO algorithm compared to standard GWO and PSO for 24 benchmark functions from the CEC2005 suite. The algorithms implement 30 independent runs and find a benchmark function, the best (minimum) value, the worst (maximum) value, the mean, and the standard deviation (Std Dev) over multiple runs. Ideally, a good optimizer will achieve lower (or higher if the function is a maximization problem) best, mean, and worst values with smaller standard deviations, indicating both accuracy and stability.

In Function F1, the proposed Adaptive_PSO_GWO achieves the best result over 30 independent runs (Best = 20.08) and a favourable mean (65.24). It clearly outperforms GWO (Best = 57.76, Mean = 237.04) and PSO (Best = 284.60, Mean = 1326.72). The lower standard deviation (33.44 compared to 145.20 for GWO and 756.29 for PSO) reflects the higher consistency of the hybrid algorithm.

Function F2, the Adaptive_PSO_GWO again yields the best (1.27) and Mean (2.30) values, surpassing GWO (Best = 3.31, Mean = 6.03) and PSO (Best = 23.67, Mean = 46.55).

Function F3. For this more complex multimodal function, GWO achieves the lowest best value (2796.95) and a lower mean value (9810.89) compared to Adaptive_PSO_GWO (best = 4840.76, mean = 12544.64), indicating better searchability for this problem. However, the large standard deviations for all three methods (in the thousands) emphasize the difficulty and robustness of F3.

Function F4. GWO slightly outperforms both Adaptive_PSO_GWO and PSO, with a best value of 10.77 and a mean value of 20.86. Nevertheless, the new hybrid remains comparable (Best = 13.55, Mean = 28.61). The best values of PSO (11.79) are close to each other but suffer from a relatively high mean value (18.17).

le **2: Result of test competition optimization algorithm Adaptive_PSO_GWO, GWO, and PSO over 30 independent ru**

Adaptive_PSO_GWO				GWO				PSO		
Be (Mi	**Worst (Max)**	**Mean**	**Std Dev**	**Best (Min)**	**Worst (Max)**	**Mean**	**Std Dev**	**Best (Min)**	**Worst (Max)**	**Mean**
08	148.81	**65.24**	33.44	57.76	832.01	237.04	145.20	284.60	3259.36	1326.72
27	3.65	**2.30**	0.64	3.31	11.22	6.03	1.75	23.67	87.34	46.55
.76	20904.44	12544.64	4417.16	2796.95	19134.74	**9810.89**	3807.16	3643.56	57816.41	14692.05
55	47.76	28.61	9.24	10.77	32.24	20.86	5.18	11.79	27.22	**18.17**
.95	78117.18	**11584.32**	16825.84	1746.56	80897.60	21033.73	19894.75	16555.27	400958.67	123081.42
37	129.72	**57.77**	27.73	81.20	612.27	230.44	111.43	249.64	2296.16	1040.51
06	0.42	0.24	0.11	0.08	0.58	**0.23**	0.12	11.76	75.46	32.29
.27	-4174.39	**-5256.36**	662.62	-6418.42	-2112.99	-3673.65	1304.18	-3854.84	-1629.51	-2319.01
52	193.65	**138.97**	35.34	80.41	298.21	159.59	65.25	240.10	381.71	316.31
07	12.50	**4.58**	1.78	3.71	8.59	5.09	1.20	5.18	10.11	7.65
0	2.74	**1.67**	0.43	1.61	6.53	2.97	0.99	205.42	334.54	262.16
31	42.18	**12.32**	8.21	4.42	41.02	14.66	8.70	10.05	45.48	25.05
34	13826.35	**501.29**	2516.76	15.35	15614.83	640.64	2854.28	35.23	292063.44	19812.06
00	19.23	9.13	5.56	1.99	20.86	10.39	5.79	1.00	17.37	**9.11**
00	0.06	**0.01**	0.01	0.00	0.02	0.01	0.01	0.00	0.06	0.01
03	-1.00	-1.03	0.01	-1.03	-1.02	-1.03	0.00	-1.03	-1.03	-1.03
40	2.71	0.48	0.42	0.40	2.17	**0.46**	0.32	0.40	2.71	0.63
00	84.30	13.82	28.03	3.00	30.00	**5.75**	8.22	3.00	84.00	8.40
86	-3.84	-3.86	0.01	-3.86	-3.81	-3.85	0.01	-3.86	-3.09	-3.82
32	-2.63	-3.16	0.19	-3.32	-2.30	**-3.17**	0.24	-3.32	-1.71	-3.10
01	-2.49	**-5.73**	3.40	-10.09	-1.60	-5.04	3.54	-10.15	-2.56	-4.96
32	-1.79	-6.48	3.28	-10.33	-2.50	-6.42	3.54	-10.40	-1.84	**-6.09**
49	-2.39	**-7.79**	3.16	-10.52	-1.70	-6.26	3.72	-10.53	-1.67	-4.91
05	5.74	**4.24**	0.87	4.00	6.72	5.25	0.74	5.02	10.18	7.47

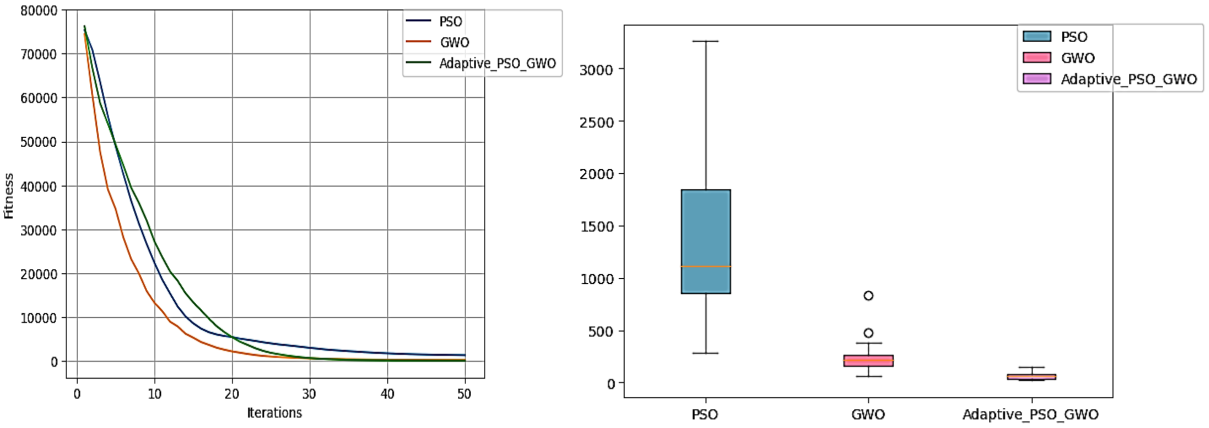
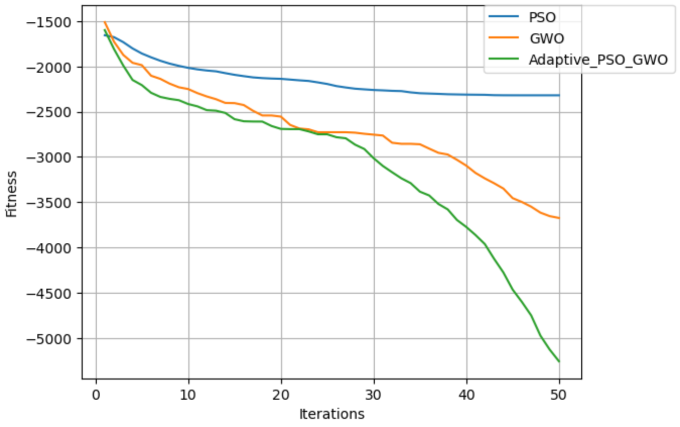
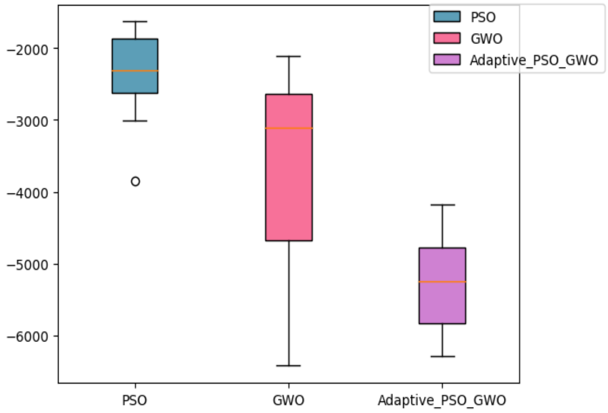


Figure 8: The average convergence of the proposed Adaptive PSO-GOW, GWO, and PSO over 30 independent runs

Figure 9 shows the convergences (Figure 9 -a) of search progress and boxplot (Figure 9 -b) of finding the best of 30 runs over F8 of CEC 2005.



a



B

Figure 9: The average convergence of the proposed Adaptive PSO-GOW, GWO, and PSO over 30 independent runs of F8.

2. Evaluate the proposed optimization model in CEC2005

This section discusses the classification performance of the tested models on the DNS-over-HTTPS (DoH) detection task, focusing on four classifiers: Random Forest (RF), Naive Bayes (NB), and their enhanced versions using the Adaptive Feature Processing Optimization (AFPO) technique.

Figure 10 illustrates the learning behavior of the proposed Adaptive PSO-GWO algorithm in minimizing the classification error when selecting optimal features, using Random Forest (RF) as the objective function. The vertical axis represents the Best Cost, which corresponds to the classification error (or inverse of accuracy), while the horizontal axis shows the number of iterations.

Initially, at iteration 0, the cost is relatively high, indicating a suboptimal feature subset. A sharp decrease occurs by iteration 1, demonstrating a rapid improvement as the algorithm begins to explore better combinations of features. From iterations 1 to 5, the cost plateaus, indicating a period of stagnation or slow improvement, which is typical as the algorithm refines its search around promising regions. A second drop is observed at iteration 6, followed by another plateau until iteration 9, where the best cost stabilizes around 0.049, suggesting that the algorithm has converged to an optimal or near-optimal solution.

This behavior highlights the balance between exploration and exploitation achieved by the hybrid Adaptive PSO-GWO approach. The initial fast convergence followed by steady refinement supports the effectiveness of the proposed model in reducing the classification error of RF through optimized feature selection

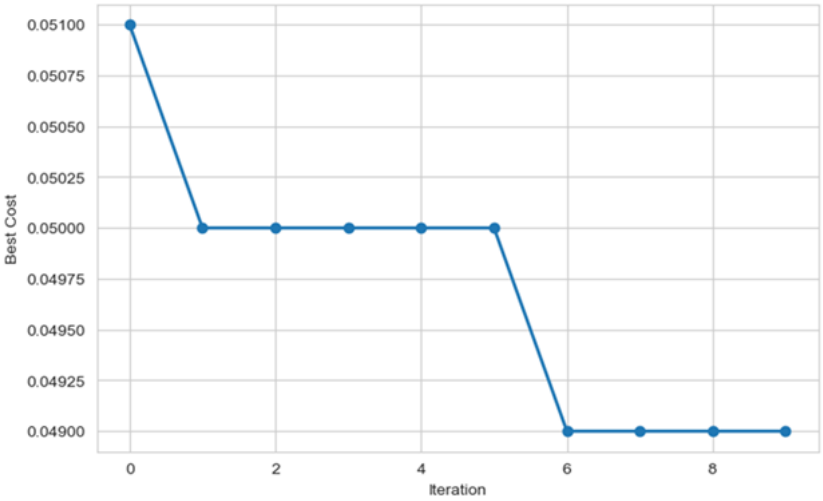


Figure 10: Learning of RF algorithm by proposed Adaptive- PSO-GWO for classification DNS traffic

Table 3 presents the true and false classification rates for both Benign and DoH traffic across four models: RF, RF+AGP, NB, and NB+AGP. The RF model demonstrates high detection accuracy, with 97.57% of Benign and 92.09% of DoH traffic accurately classified and relatively low false rates. When combined with the AGP enhancement, RF+AGP slightly improves the true positive rate for Benign traffic to 98.18%. It reduces its false rate to 1.82% while maintaining the same performance for DoH traffic, suggesting a modest refinement. Conversely, the NB model struggles with accurately identifying DoH traffic, achieving only 37.75% true positives and a high false positive rate of 62.25%. However, following the application of AGP, the NB+AGP model exhibits a substantial improvement in detecting DoH traffic, raising the true rate to 83.30% and decreasing the false rate to 16.70%. This gain comes at the expense of reduced accuracy in detecting Benign traffic, where the true rate drops to 70.25%. Overall, the table indicates that while RF performs consistently well with or without enhancement, the AGP technique significantly aids the weaker NB model, particularly in enhancing DoH traffic detection.

Table 3: confusion matric rate of RF and NB classifier with and without proposed feature selection model

Model	T-Benign	T-DoH	F-Benign	F-DoH
RF	97.57%	92.09%	2.43%	7.91%
RF+AGP	98.18%	92.09%	1.82%	7.91%
NB	94.53%	37.75%	5.47%	62.25%
NB+AGP	70.25%	83.30%	29.15%	16.70%

Table 4 provides a comparative evaluation of four models—RF, RF+AFPO, NB, and NB+AFPO—based on key classification metrics: precision, recall, and F1-score for both Benign and DoH classes, along with overall accuracy, macro average, and weighted average scores. The Random Forest (RF) model demonstrates balanced and high performance across all metrics, with an overall accuracy of 94.80% and nearly equal F1 scores for both classes (94.88% for Benign and 94.72% for DoH). When combined with the AFPO optimization technique, the RF+AFPO model shows a slight improvement in all metrics, particularly in precision for the DoH class (rising to 98.11%) and an increase in the overall weighted average to 95.28%, confirming the refinement introduced by the optimization.

In contrast, the Naive Bayes (NB) model initially performs poorly, especially in detecting DoH traffic, with a low recall of 37.75% and an imbalanced F1-score distribution (73.20% for Benign vs. 52.76% for DoH), leading to a low weighted average of 62.86%. However, after integrating AFPO, the NB+AFPO model shows notable improvement, balancing the precision and recall across both classes and increasing the F1-score for DoH to 78.67%. The overall accuracy rises to 77.15%, demonstrating the optimization's effectiveness in enhancing a weaker base classifier. These results underline that while RF performs well on its own, the AFPO technique offers significant benefits when applied to more limited models like NB, substantially improving their classification reliability.

Table 4 : Comparative performance of RF and NB classifier with and without proposed feature selection model

Model	Metrics	Benign	DoH	accuracy	macro avg	weighted avg
RF	precision	92.34%	97.49%	94.80%	94.91%	94.94%
	recall	97.57%	92.09%	94.80%	94.83%	94.80%
	f1-score	94.88%	94.72%	94.80%	94.80%	94.80%
RF+AFPO	precision	92.38%	98.11%	95.10%	95.24%	95.28%
	recall	98.18%	92.09%	95.10%	95.14%	95.10%
	f1-score	95.19%	95.01%	95.10%	95.10%	95.10%
NB	precision	59.72%	87.61%	65.80%	73.67%	73.83%
	recall	94.53%	37.75%	65.80%	66.14%	65.80%
	f1-score	73.20%	52.76%	65.80%	62.98%	62.86%
NB+AFPO	precision	80.55%	74.54%	77.15%	77.54%	77.51%
	recall	70.85%	83.30%	77.15%	77.08%	77.15%
	f1-score	75.39%	78.67%	77.15%	77.03%	77.05%

CONCLUSION

This study proposed a hybrid feature selection framework that combines Particle Swarm Optimization (PSO) and Grey Wolf Optimizer (GWO) to enhance the detection of malicious activities in encrypted DNS traffic. It addressed the limitations of individual metaheuristic algorithms by balancing exploration and exploitation capabilities. Experimental results using benchmark functions from CEC2005 demonstrated that the proposed Adaptive_PSO_GWO outperformed standard PSO and GWO algorithms. It achieves the best values, means, and standard deviations. That indicates both higher accuracy and consistency.

The Random Forest already performed well with 94.80% accuracy and saw modest improvements with optimization (95.10%); the most notable enhancement was observed with the Naive Bayes classifier, which improved from 65.80% to 77.15% accuracy when combined with our optimization approach.

This demonstrates that our hybrid method particularly benefits weaker classifiers by selecting more relevant features and reducing dimensionality. The research successfully achieved its objectives of reducing data complexity through intelligent feature selection, designing an effective hybrid optimization method, and improving classification accuracy for encrypted DNS traffic detection. The balance between exploration and exploitation achieved by our hybrid approach enables more efficient navigation of complex search spaces to find near-optimal feature subsets. This work contributes to the ongoing challenge of securing encrypted communications by providing a more accurate and efficient detection system that can identify malicious activities without compromising the privacy benefits of encryption.

To improve trust in detection decisions, future work should incorporate explainable AI techniques that can provide interpretable insights into why specific traffic is classified as malicious. This would help security analysts understand detection rationales and reduce false positive investigations.

REFERENCES

[1] R. R. Nuiaa, S. Manickam, and A. H. Alsaeedi, "A Comprehensive Review of DNS-based Distributed Reflection Denial of Service (DRDoS) Attacks: State-of-the-Art," International Journal on Advanced Science, Engineering and Information, vol. 6, no. 12, pp. 2452-2461, 2022.

[2] S. E. Mathew, Y. S. Vali, and L. Shakkeera, "Botnet Detection Methods: A Review and Classification," in 2025 3rd International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT), 2025: IEEE, pp. 497-502.

[3] S. M. Hadi, A. H. Alsaeedi, R. R. Nuiaa, S. Manickam, and A. S. D. Alfoudi, "Dynamic Evolving Cauchy Possibilistic Clustering Based on the Self-Similarity Principle (DECS) for Enhancing Intrusion Detection System," International Journal of Intelligent Engineering & Systems, vol. 15, no. 5, 2022.

[4] A. H. Alsaeedi, S. M. Hadi, and Y. Alazzawi, "Adaptive Gamma and Color Correction for Enhancing Low-Light Images," International Journal of Intelligent Engineering & Systems, vol. 17, no. 4, 2024.

[5] A. H. Alsaeedi, A. M. Al-juboori, H. H. R. Al-Mahmood, S. M. Hadi, H. J. Mohammed, M. R. Aziz, M. Aljibawi, and R. R. Nuiaa, "Dynamic Clustering Strategies Boosting Deep Learning in Olive Leaf Disease Diagnosis," Sustainability, vol. 15, no. 18, p. 13723, 2023.

[6] M. Jeong, J. Park, and S. H. Oh, "Cyber Environment Test Framework for Simulating Command and Control Attack Methods with Reinforcement Learning," Applied Sciences, vol. 15, no. 4, p. 2120, 2025.

[7] R. Flood, M. Casadio, and D. Aspinall¹, "Generating Traffic-Level Adversarial Examples from Feature-Level," in Computer Security. ESORICS 2024 International Workshops: SECAI, DisA, CPS4CIP, and SecAssure, Bydgoszcz, Poland, September 16–20, 2024, Revised Selected Papers, Part II, 2025, vol. 15264: Springer Nature, p. 118.

[8] S. Dwivedi, M. Vardhan, and S. Tripathi, "Building an efficient intrusion detection system using grasshopper optimization algorithm for anomaly detection," Cluster Computing, vol. 24, no. 3, pp. 1881-1900, 2021/09/01 2021, doi: 10.1007/s10586-020-03229-5.

[9] R. R. Nuiaa, S. Manickam, A. H. Alsaeedi, and E. S. Alomari, "A new proactive feature selection model based on the enhanced optimization algorithms to detect DRDoS attacks," Int. J. Electr. Comput. Eng, vol. 12, no. 2, pp. 1869-1880, 2022.

[10] A.-K.-S. Sabonchi, "Hybrid Metaheuristic Lion and Firefly Optimization Algorithm with Chaotic Map for Substitution S-Box Design," Journal of Information Hiding and Privacy Protection, vol. 6, no. 1, pp. 21--45, 2024. [Online]. Available: http://www.techscience.com/jihpp/v6n1/59164.

[11] R. R. Nuiaa, S. Manickam, A. H. Alsaeedi, and D. E. J. Al-Shammary, "Evolving Dynamic Fuzzy Clustering (EDFC) to Enhance DRDoS_DNS Attacks Detection Mechnism," International Journal of Intelligent Engineering and Systems, vol. 15, no. 1, pp. 509-519, 2022.

[12] M. H. M. Yusof, A. A. Almohammedi, V. Shepelev, and O. Ahmed, "Visualizing realistic benchmarked IDS dataset: CIRA-CIC-DoHBrw-2020," IEEE Access, vol. 10, pp. 94624-94642, 2022.

[13] B. Bala and S. Behal, "AI techniques for IoT-based DDoS attack detection: Taxonomies, comprehensive review and research challenges," Computer science review, vol. 52, p. 100631, 2024.

[14] S. M. S. Islam, "Analyzing distributed denial-of-service attacks in SDN architecture," Macquarie University, 2024.

[15] A. A. Najar and S. M. Naik, "Cyber-secure SDN: A CNN-based approach for efficient detection and mitigation of DDoS attacks," Computers & Security, vol. 139, p. 103716, 2024.

[16] A. H. Alsaeedi, D. Al-Shammary, S. M. Hadi, K. Ahmed, A. Ibaida, and N. AlKhazraji, "A proactive grey wolf optimization for improving bioinformatic systems with high dimensional data," International Journal of Information Technology, vol. 16, no. 8, pp. 4797-4814, 2024.

Journal of Information Systems Engineering and Management
2025, 10(36s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

[17] Y. Xiao, H. Cui, A. G. Hussien, and F. A. Hashim, "MSAO: A multi-strategy boosted snow ablation optimizer for global optimization and real-world engineering applications," Advanced Engineering Informatics, vol. 61, p. 102464, 2024.

[18] M. Song, M. An, W. He, and Y. Wu, "Research on land use optimization based on PSO-GA model with the goals of increasing economic benefits and ecosystem services value," Sustainable Cities and Society, vol. 119, p. 106072, 2025.

[19] A. H. Alsaeedi, A. H. Aljanabi, M. E. Manna, and A. L. Albukhnefis, "A proactive metaheuristic model for optimizing weights of artificial neural network," Indones. J. Electr. Eng. Comput. Sci, vol. 20, no. 2, pp. 976-984, 2020.

[20] X. Cai, L. Wu, T. Zhao, D. Wu, W. Zhang, and J. Chen, "Dynamic adaptive multi-objective optimization algorithm based on type detection," Information sciences, vol. 654, p. 119867, 2024.

[21] R. R. Nuiaa, S. A. A. A. Alsaidi, B. K. Mohammed, A. H. Alsaeedi, Z. A. A. Alyasseri, S. Manickam, and M. A. Hussain, "Enhanced PSO Algorithm for Detecting DRDoS Attacks on LDAP Servers," International Journal of Intelligent Engineering & Systems, vol. 16, no. 5, 2023.

[22] A. H. Alsaeedi, M. A. Al-Sharqi, S. S. Alkafagi, R. R. Nuiaa, A. S. D. Alfoudi, S. Manickam, A. M. Mahdi, and A. M. Otebolaku, "Hybrid extend particle swarm optimization (EPSO) model for enhancing the performance of MANET routing protocols," Journal of Al-Qadisiyah for computer science and mathematics, vol. 15, no. 1, pp. Page 127-136, 2023.

[23] H. Benbouhenni, G. Hamza, M. Oproescu, N. Bizon, P. Thounthong, and I. Colak, "Application of fractional-order synergetic-proportional integral controller based on PSO algorithm to improve the output power of the wind turbine power system," Scientific Reports, vol. 14, no. 1, p. 609, 2024.

[24] A. Alfoudi, A. Alsaeedi, M. Abed, A. Otebolaku, and Y. Razooqi, "Palm vein identification based on hybrid feature selection model," International Journal of Intelligent Engineering and Systems, vol. 14, no. 5, pp. 469-478, 2021.

[25] S. M. Ali, A. H. Alsaeedi, D. Al-Shammary, H. H. Alsaeedi, and H. W. Abid, "Efficient intelligent system for diagnosis pneumonia (SARSCOVID19) in X-ray images empowered with initial clustering," Indones. J. Electr. Eng. Comput. Sci, vol. 22, no. 1, pp. 241-251, 2021.

[26] X. Liu, H. Tang, Y. Ding, and D. Yan, "Investigating the performance of machine learning models combined with different feature selection methods to estimate the energy consumption of buildings," Energy and Buildings, vol. 273, p. 112408, 2022.

[27] S. Mishra, P. K. Mallick, H. K. Tripathy, A. K. Bhoi, and A. González-Briones, "Performance evaluation of a proposed machine learning model for chronic disease datasets using an integrated attribute evaluator and an improved decision tree classifier," Applied Sciences, vol. 10, no. 22, p. 8137, 2020.

[28] A. H. Alsaeedi, H. H. R. Al-Mahmood, Z. F. Alnaseri, M. R. Aziz, D. Al-Shammary, A. Ibaida, and K. Ahmed, "Fractal feature selection model for enhancing high-dimensional biological problems," BMC bioinformatics, vol. 25, no. 1, p. 12, 2024.

[29] H. Zouhri and A. Idri, "A Comparative Assessment of Wrappers and Filters for Detecting Cyber Intrusions," in World Conference on Information Systems and Technologies, 2024: Springer, pp. 118-127.

[30] A. H. . Al-Fatlawi, S. S. . Kashef, Y. S. Mezaal, and M. . Valizadeh, “Design of a Compact Microstrip Band Pass Filter for IoT and S-Band Radar Applications”, Data and Metadata, vol. 4, p. 714, Feb. 2025, doi: 10.56294/dm2025714.

[31] A. H. Al-fatlawi and M. A. Taha, "New microstrip filter for MIMO wireless and computer systems," Journal of Theoretical & Applied Information Technology, vol. 96, no. 12, 2018.