Journal of Robotics and Control (JRC)
Volume 6, Issue 1, 2025
ISSN: 2715-5072, DOI: 10.18196/jrc.v6i1.23649

Two-Level Feature Selection for Enhanced Accuracy and Reduced Computational Overhead in Intrusion Detection Systems Using Rough Set Theory and Binary Particle Swarm Optimization

Moaad Almania [1*], Anazida Zainal [2], Fuad A Ghaleb [3], Ahmad Alnawasrah [4], Mahmoud Al Qerom [5]
[1, 2] Faculty of Computing, Universiti Teknologi Malaysia, Johor Bahru, Johor, 81310, Malaysia
[1] College of Computing and Information Technology, Shaqra University, Shaqra, Kingdom of Saudi Arabia
[3] College of Computing and Digital Technology, Birmingham City University, Birmingham, B47XG, United Kingdom
[4, 5] Department of Information Communication Technology, British University of Bahrain, Bahrain
Email: [1] malmane3@su.edu.sa
*Corresponding Author

***Abstract*—Intrusion Detection Systems (IDS) are essential for safeguarding network infrastructures by detecting and mitigating malicious activities. This study introduces a two-level feature selection approach (TLFSA) designed to enhance classification accuracy and reduce computational overhead. The first phase employs Rough Set Theory (RST) to filter out irrelevant features, while the second phase uses Binary Particle Swarm Optimization (BPSO) to refine the feature subset based on their discriminative power. Experiments conducted on the NSL-KDD dataset show that the TLFSA approach outperforms traditional algorithms such as Genetic Algorithm (GA) and Gravitational Search Algorithm (GSA), achieving a notable improvement of 0.99% in classification accuracy. Furthermore, class-specific feature subsets produced by the method demonstrate superior detection rates across all network traffic classes, with an average accuracy of 97.22%, compared to 91.11% for alternative methods. The proposed method effectively reduces the feature set to approximately 15% of the original features, streamlining the IDS model and improving both operational efficiency and real-time applicability.**

Keywords—Feature Selection; Rough Set Theory; PSO; BPSO.

I. INTRODUCTION

Intrusion Detection Systems are fundamental to securing modern network infrastructures, where they serve as a defense mechanism against malicious activities [1]–[7]. As networks continue to grow in complexity and the volume of traffic increases, IDS face significant challenges in maintaining high detection accuracy while minimizing computational overhead [61][62][63]. One of the primary issues is handling high-dimensional data generated by network traffic, which can lead to increased false positives and higher computational costs. Feature selection (FS) has become a crucial process to mitigate these challenges by reducing the data to the most relevant features, thereby improving the system's efficiency and accuracy [8]–[16].

Despite the extensive research on feature selection, many existing methods have limitations related to scalability, computational efficiency, and adaptability. Approaches such as Genetic Algorithms and Particle Swarm Optimization have been widely used for feature selection in IDS [17]–[21], but they frequently struggle with large-scale datasets, encountering issues such as slow convergence, increased computational burden, and poor adaptability to dynamic network conditions. These limitations highlight the need for more efficient and scalable methods that can handle evolving network environments and the growing complexity of cyber threats.

To address these challenges, this paper introduces a Two-Level Feature Selection Approach that integrates Rough Set Theory with Binary Particle Swarm Optimization. The TLFSA consists of two sequential phases: in the first phase, RST is used as a filtering technique to eliminate irrelevant and redundant features, significantly reducing the dimensionality of the dataset [22]–[26]. In the second phase, BPSO acts as a wrapper method to optimize the feature subset by focusing on the most discriminative features for different classes of network traffic, such as normal, denial of service (DoS), and probing attempts [27]–[32]. This two-level approach ensures both computational efficiency and enhanced classification performance.

The novelty of this method lies in its ability to generate class-specific feature subsets, which tailor the feature selection process to the distinct characteristics of different network traffic classes. This not only reduces the computational burden but also enhances IDS detection accuracy for various types of attacks. Moreover, the combination of RST and BPSO addresses the scalability issues often seen in traditional PSO-based methods, making TLFSA suitable for real-time intrusion detection in large-scale network environments [33]–[40].

In this study, we aim to contribute a scalable, adaptable, and efficient feature selection strategy that enhances IDS performance. The remainder of the paper is organized as follows: Section 2 provides a review of existing feature selection techniques for IDS. Section 3 outlines the proposed methodology, including the dataset and experimental setup. Section 4 presents the results and comparative analysis, while

Section 5 concludes with discussions on limitations and future research directions.

II. LITERATURE REVIEW

The literature on Intrusion Detection Systems highlights the critical role of feature selection in improving both the accuracy and efficiency of IDS models [1], [70], [75]-[78], [80]. Numerous approaches have been proposed, typically categorized into filter methods, wrapper methods, embedded methods, and hybrid methods. Filter methods, such as Information Gain, Chi-Squared, and Correlation-based Feature Selection [30]–[33], evaluate feature relevance without considering the performance of the learning algorithm itself, making them computationally efficient but potentially less accurate. Wrapper methods, such as Recursive Feature Elimination (RFE) and Genetic Algorithm, assess features based on their impact on the learning algorithm's performance [34]–[35], offering higher accuracy but at a much greater computational cost. Embedded methods, which incorporate feature selection during the model training process, are commonly used in decision trees and deep learning approaches [36]. Hybrid methods, which combine filter and wrapper techniques, attempt to balance computational cost and detection performance [37].

In the context of IDS, Particle Swarm Optimization has been widely adopted due to its ability to optimize feature subsets by mimicking the social behavior of swarms [22], [23], [71]. However, despite its advantages, PSO faces challenges such as slow convergence and difficulty scaling to high-dimensional datasets, often leading to suboptimal feature subsets for IDS [6], [7], [64]-[68]. Similar problems have been observed with Genetic Algorithm methods, where the complexity of searching large solution spaces results in high computational costs and potential convergence to local optima [9]. Existing PSO-based feature selection methods are also limited by their static nature, which fails to account for evolving network conditions and cyber threats [10].

Several studies have proposed dynamic feature selection methods to adapt to changing network conditions and evolving threats [38]–[40], [69], [72]-[74]. For example, a sliding window mechanism was suggested in [13] to allow the IDS to adjust its feature set based on recent network activity. While this approach improves adaptability, it still relies on static optimization techniques like PSO, which struggle with large-scale datasets [14]. Similarly, hybrid approaches that combine evolutionary algorithms, such as the Gravitational Search Algorithm [44], with support vector machines have demonstrated improved detection accuracy but often add complexity and require extensive tuning [16]. In the context of big data and real-time processing, studies like [43] have discussed the limitations of traditional methods in handling high-dimensional data streams, suggesting the need for scalable and efficient solutions for IDS [17].

A key limitation of existing methods is their inability to handle class-specific feature subsets effectively. Different types of attacks, such as Denial of Service and Probe attacks, often exhibit distinct characteristics [39]–[41]. However, most traditional methods use generic feature sets that fail to optimize detection for specific classes of network traffic [18]. Recent research has explored the use of machine learning models like random forests and deep neural networks to generate class-specific feature subsets, but these methods are computationally expensive and require large amounts of labeled data [20]. Additionally, while some studies focus on big data, the unique challenges of real-time IDS deployments, including computational efficiency and model interpretability, are often overlooked [21].

The research gap, therefore, lies in developing a scalable, adaptable feature selection approach that reduces computational overhead while improving detection accuracy for class-specific network traffic. The current study addresses these gaps by proposing a two-level feature selection approach that integrates Rough Set Theory with Binary Particle Swarm Optimization. RST is employed as a filter to eliminate redundant and irrelevant features, significantly reducing the dimensionality of the data. BPSO is then applied to refine the remaining feature subset, optimizing it for class-specific detection performance. This approach not only reduces the computational complexity associated with traditional methods but also enhances the overall detection accuracy by focusing on class-specific feature subsets [45][46].

In conclusion, while existing feature selection techniques have contributed significantly to the improvement of IDS, they often fall short in scalability, adaptability, and computational efficiency. The proposed two-level approach offers a novel solution to these challenges by providing a more balanced, efficient, and accurate feature selection process, which is better suited to the unique demands of intrusion detection in dynamic network environments.

III. METHODOLOGY

The proposed feature extraction technique in this study consists of two phases: initial and final phases. The initial and final phases of the two-level feature selection are presented in Fig. 1. Rough Set Theory was used at the entry-level. RST employs a filtering strategy. Its goal was to search the feature space and remove unnecessary and irrelevant features. The second level deployed Binary Particle Swarm Optimization in the interim. The network traffic is classified into the following categories: Normal, Probe, DoS, U2R, and R2L using this method, which is categorized under the wrapper approach. If the data dimension is large, PSO might encounter a local optimization issue [22]. A greater dimension denotes a broader search area. Like other stochastic algorithms, the PSO process takes longer as the search space expands [23]. Therefore, when dealing with large search spaces, combining the RST (filtering approach) and BPSO (wrapper approach) should complement the time-consuming aspect of PSO. The search time should also be reduced by reducing the search space. The purpose of feature selection is to remove unnecessary and redundant features while preserving the approximation quality of the initial set of features [43]. Feature selection aims to increase detection accuracy while reducing the processing volume of work. The feature selection process used in this study is depicted in Fig. 1. A fitness function was used to assess the effectiveness of the reduced feature subsets.

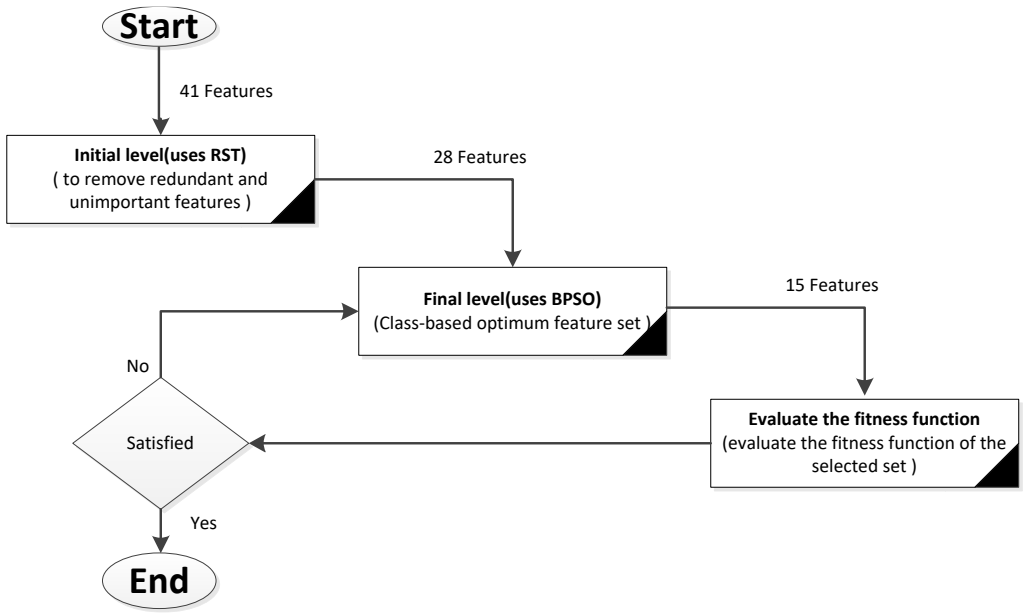

Fig. 1. TLFSA approach

Rough Set Theory and Binary Particle Swarm Optimization are employed for feature selection due to their specific advantages in handling the complexities of Intrusion Detection Systems. RST is chosen because it is an effective filtering technique that excels at handling uncertainty and redundancy in datasets. Its ability to reduce features while maintaining classification accuracy is well-documented, making it a suitable candidate for the initial phase of feature selection where irrelevant features need to be discarded. BPSO, on the other hand, is utilized in the second phase because of its optimization capabilities. It refines the reduced feature subset by focusing on the discriminative power of the features. This combination allows for a balance between computational efficiency (RST's strength) and the fine-tuning of features for classification (BPSO's strength). While other techniques such as Genetic Algorithms and Gravitational Search Algorithms are commonly used, BPSO has been selected for its lower computational cost and ability to avoid the stagnation issues often seen in GA. Furthermore, BPSO allows for an adjustable trade-off between exploration and exploitation in the search space, which is crucial when working with IDS data characterized by high dimensionality and variability.

Scalability is a critical issue when dealing with high-dimensional IDS datasets, and while BPSO offers many advantages, it can suffer from local optimization and longer search times as the data's dimensionality increases. To mitigate this, we propose a two-level feature selection approach where the initial phase (RST) reduces the dimensionality by filtering out irrelevant features, significantly decreasing the search space before applying BPSO. Additionally, empirical evidence from our experiments shows that the computational complexity of BPSO is reduced after applying RST, making the method scalable even when handling large datasets. However, we acknowledge that in extreme cases of high-dimensional data, advanced strategies like dimensionality reduction techniques (e.g., Principal Component Analysis) could be considered to further enhance scalability.

While GAs have been used for finding reducts in RST, their computational cost and tendency to converge on suboptimal solutions make them less ideal for this application. In our approach, BPSO replaces GA due to its more efficient convergence properties and better handling of high-dimensional search spaces. This mitigates over-reliance on GA by introducing a more computationally feasible alternative that still maintains a high degree of accuracy in feature selection.

A key limitation of the current methodology is the assumption of a static dataset. While it improves feature selection for IDS, real-time adaptability to evolving threats is crucial. Future work could integrate adaptive learning to update feature subsets dynamically, ensuring high accuracy against new or evolving attacks in real-world deployments.

The fitness function used in BPSO is a key component of the feature selection process, as it evaluates the quality of feature subsets based on classification accuracy and subset length. The criteria for evaluating fitness include maximizing detection accuracy and minimizing the number of selected features to ensure computational efficiency. A balanced trade-off between these two metrics is essential to avoid overfitting and unnecessary complexity in the model.

Pseudo code of the two-level procedure:

1. Choose random training and testing datasets
2. Rough Set Theory generates classification rules based on a training dataset.
3. Choose the most critical reducts/features that appear in RST rules.
4. BPSO algorithm should be used.
5. Determine each particle's fitness. Revisit Step 4 if the fitness value is below the predetermined fitness value; otherwise, exit.

A. Initial Level (Rough Set Theory)

Entire reducts were used to obtain just attributes (reducts) that maintain the indiscernibility relationship with all instances in the training set. Prior to proceeding with the computation of minimal attribute subsets that effectively differentiate the first object within X (specifically, benign HTTP connections in this instance) from all other pertinent objects within U (representing the universe or other data in the training set), we first computed the minimal attribute subsets that distinguish the second object within X from all other relevant objects in U. The Genetic Algorithm was employed in this study to find reducts because it is widely used and is claimed to be the most effective algorithm for significant system reduction computation in practice [45]. Rosetta's software's full reducts are shown in Table I.

Based on training data, 28 Reducts are produced, as shown in Table I. These Reducts were made up of features; duration(1), src_byte (f5), dst_byte (f6), hot(10), num_failed_login(11), logged_in (12), num_compromised (13), su_attempted (15), num_root (16), num_file_creations (17), num_access_files (19), is_guest_login (22) , count (23), srv_count(24), serror_rate (f25), srv_serror_rate (26), srv_rerror_rate (f28), diff_srv_rate (30), srv_diff_host_rate (31), dst_host_count (32), dst_host_srv_count (33), dst_hostsame_srv_rate (34), dst_host_diff_srv_rate (35), dst_host_same_src_port_rate (36), dst_host_srv_diff_host_rate (37), dst_host_serror_rate(38), dst_host_srv_serror_ate (39), dst_host_rerror_rate (f40), and dst_host_srv_rerror_rate (41). Protocol type (f2), service (f3), and Flag (4) were added as new features. F2 was selected because HTTP is a service available via the TCP protocol, so this study attempted to exclude only the HTTP service, which was indicated by feature f3. These characteristics were plotted individually, and their discriminative powers were assessed. Three data subsets from the dataset were randomly chosen for testing and training. The data distribution across each of the utilized data sets is listed in Table II.

TABLE I. FULL REDUCTS

Reduces	Support	length	Reduces	support	length	Reduces	support	length	Reduces	support	length
[f35]	100	1	[f37]	100	1	[f32, f33]	100	2	[f5, f24]	100	2
[f36]	100	2	[f12]	100	1	[f32, f34]	100	2	[f5, f37]	100	2
[f23]	100	1	[f24]	100	1	[f33, f37]	100	2	[f5, f35]	100	2
[f5]	100	1	[f41]	100	1	[f32, f35]	100	2	[f5, f36]	100	2
[f30]	100	1	[f39]	100	1	[f35, f37]	100	2	[f5, f34]	100	2
[f33]	100	1	[f33, f38]	100	2	[f1]	100	1	[f6, f24]	100	2
[f34]	100	1	[f35, f38]	100	2	[f24, f40]	100	2	[f12, f33]	100	2
[f29]	100	1	[f38, f40]	100	2	[f23, f24]	100	2	[f5, f33]	100	2
[f24, f37]	100	2	[f34, f38]	100	2	[f24, f31]	100	2	[f12, f24]	100	2
[f24, f32]	100	2	[f31, f33]	100	2	[f31, f37]	100	2	[f1, f33]	100	2
[f26]	100	1	[f31]	100	2	[f24, f35]	100	2	[f31, f36]	100	2
[f32]	100	1	[f40]	100	1	[f24, f38]	100	2	[f5, f32]	100	2
[f6]	100	1	[f24, f33]	100	2	[f23, f37]	100	2	[f6, f33]	100	2
[f25]	100	1	[f34, f37]	100	2	[f23, f35]	100	2	[f5, f6]	100	2
[f19]	100	1	[f23, f40]	100	2	[f12, f31, f34]	100	3	[f31, f32, f33]	100	3
[f10]	100	1	[f23, f38]	100	2	[f24, f26]	100	2	[f12, f32, f34]	100	3
[f15]	100	1	[f31, f32, f34]	100	3	[f24, f25]	100	2	[f6, f32, f34]	100	3
[f11]	100	1	[f23, f33]	100	2	[f37, f41]	100	2	[f23, f32, f33]	100	3
[f16]	100	1	[f38]	100	1	[f36, f37]	100	2	[f6, f32, f35]	100	3
[f13]	100	1	[f23, f31]	100	2	[f6, f37]	100	2	[f24, f41]	100	2
[f34, f36]	100	2	[f28]	100	1	[f30, f33]	100	2	[f23, f33, f34]	100	3
[f33, f36]	100	2	[f17]	100	1	[f12, f33, f36]	100	3	[f6, f33, f34]	100	3
[f24, f36]	100	2	[f33, f34]	100	2	[f12, f31, f33]	100	3	[f24, f33, f34]	100	3
[f22]	100	1	[f32, f37]	100	2	[f12, f32, f33]	100	3	[f6, f33, f41]	100	3
[f31, f34, f37]	100	3	[f31, f32]	100	2	[f32, f36]	100	2	[f1, f6, f33]	100	3
[f31, f33, f35]	100	3	[f23, f32]	100	2	[f31, f32, f35]	100	3	[f6, f33, f35]	100	3
[f31, f33, f38]	100	3	[f33, f35]	100	2	[f12, f23, f33]	100	3	[f25, f38]	100	2
[f5, f31]	100	2	[f24, f34]	100	2	[f12, f32, f35]	100	3	[f37, f38]	100	2

TABLE II. FEATURE SELECTION CLASSES DISTRIBUTED ACROSS TRAINING AND TEST DATASETS

Data	Classes					
	Normal	Probe	DoS	U2R	R2L	Total
Training	4,000	3000	700	11	65	7,776
Testing 1. Set 1	4,000	3000	700	11	65	7,776
2. Set 2	4,000	3000	700	11	65	7,776

There is no data redundancy and an equal amount for each of the two data subsets being used for testing, except the U2R class, whose volume in the original dataset was modest. Ten different significant features were put forth by [46], and a maximum of 17 features were suggested by [47]. The fifteen best reducts with the highest scores in the generated rules were chosen due to the difficulty posed by the giant search space and to speed up the optimization in Particle Swarm. Table III includes a list of these 15 features. Using similar algorithms, the training and testing datasets were first discretized and reduced using Naive and Genetic Algorithms, respectively. Since most published works used fewer than fifteen significant features in their IDS works, the feature set was restricted to fifteen [48][49]. The selection of 15 features as the optimal number was based on extensive experimentation, where we observed that reducing the feature set to around 15 yielded the best balance between accuracy and computational efficiency. This number was not arbitrarily chosen but rather empirically derived through iterative testing, ensuring that it represents the most critical features for IDS performance while keeping the feature set manageable for real-time applications.

To further strengthen the methodology, a detailed computational complexity analysis is recommended. While we have observed that the two-level approach improves the efficiency of the feature selection process, particularly in reducing the dimensionality and thus the search space for BPSO, a formal analysis would provide valuable insights into the scalability of the approach when applied to larger datasets.

TABLE III. RST'S CHOICE OF THE TOP 15 FEATURES FOR EACH TRAFFIC CLASS

Classes	Significant Features
Normal	F2, F3, F4, F5, F11, F13, F15, F16, F17, F19, F22, F25, F26, F28, F38
Probe	F2, F5, F6, F8, F10, F12, F13, F23, F24, F30, F31, F32, F35, F36, F37
DoS	F1, F5, F6, F10, F12, F13, F16, F17, F23, F24, F25, F26, F37, F38, F39
U2R	F1, F2, F3, F5, F6, F9, F12, F16, F23, F24, F29, F30, F32, F33, F37
R2L	F1, F4, F5, F6, F11, F12, F27, F28, F29, F30, F35, F38, F39, F40, F41

B. Final Level Feature Selection Using Binary Particle Swarm Optimization

As previously mentioned, Binary Particle Swarm Optimization was implemented to select the final level feature. Like PSO, BPSO uses Equations (1) and (2) to determine each particle's velocity (V_{id}) and position (X_{id}). This section describes PSO.

$$V_{id} = wV_{id} + C_1 rand()(P_{id} - X_{id}) + C_2 Rand()(P_{gd} - X_{id}) \quad (1)$$

$$X_{id} = X_{id} + V_{id} \quad (2)$$

Positive constants C_1 and C_2 are used to represent learning rates. These show how the stochastic acceleration terms, which force every particle toward its best positions, are weighted. High values cause sudden moves toward target regions, whereas low values allow particles to move far from the target regions before being pulled back.

w is the inertia weight, and $rand()$ and $Rand()$ are two random functions with a range of [0, 1]. With the right choice of inertia weight, global and local exploration are balanced, and it takes fewer iterations to arrive at an optimal solution on average. The i^{th} particle is defined by $X_i = (x_{i1}, x_{i2}, \ldots, x_{id})$, and the ith particle's best previous position is shown by $P_i = (p_{i1}, p_{i2}, \ldots, p_{id})$. $V_i = (v_{i1}, v_{i2}, \ldots, v_{id})$ denotes the velocity or rate of position change for particle i.

Typically, one fragment of a particle signifies one feature for feature representation. The bit is set to "1" if the feature is selected, and "0" is anything other than that. A particle's features were chosen using a few different methods. Some studies randomly selected these features, while others used a roulette wheel to choose them [50, 51]. A few published works used selection pressure to limit the likelihood of choosing highly fitting features [52]. [53] used the sigmoid function to squash Vid and used velocity as a probability to predict whether Xid will be in state "1" or state "0". The Particle Swarm Optimization implementation is shown in Fig. 2, and Table IV provides the critical parameter values.

TABLE IV. KEY PARAMETERS VALUES USED IN BPSO

Parameter	Values	Description
N	15	This is the length of initial level features
max_fitness	85%	The highest fitness value that satisfies accuracy and feature subset length
m	5	Number of particles
particles best	Variable length	P local best
Gbest	Variable length	P global best. The best feature subset

function BPSO (*reducts*) **return** proposed_features *(G_{best})*
Inputs: *x,* is training data where $\Sigma^n x$ of *X* data from the training dataset with *y* attributes
Begin
max_fitness › *y*
for i=1 ***to*** *m*
particle › randomly initialize possible position. (1 feature is chosen, 0 otherwise)
$particle_{i\_lbest}$ › *$particle_i$*
end
while (curr_fitness < max_fitness) ***do***
Read data with respective feature subset (as represented by a particle) from input, *X*
for i=1 **to** *m* do
Evaluate fitness for *particles* and *$particle_{i\_lbest}$* according to Equation (4.6)
if *particlei_fitness > particlei_lbest_fitness*
then $particle_{lbest} = particle_i$
End

Fig. 2. PSO pseudo code

The parameter settings for TLFSA and BBA were tuned through grid search, ensuring optimal performance across all datasets. The following parameters were tuned: the number of particles in BPSO, inertia weight, and the mutation rate for GA.

The algorithm will stop iterating whenever the proposed feature subset's fitness exceeds the predetermined fitness value. A fitness function is typically defined as the correct classification rate using the features chosen by each particle in most feature selection works. The fitness function shown in Equation (3) was used in this study because it considers both the length of the fitness function and the significance of the features. Bae, et al. [56] employed the same fitness function.

$$Fitness\ of\ the\ proposed\ features(R) = a \times YR(D) + B \times \frac{|C| - |R|}{|C|} \quad (3)$$

The classification rate for feature subset R concerning decision D is denoted by the term $\gamma R\ (D)$. $|R|$ is the position's "1" number or the length of the chosen feature subset. The overall number of features is $|C|$. The parameters denoted by the symbols α and β stand for the significance of classification quality and the length of the feature subset, respectively. They have values of 0 and 1, respectively. The length of the feature subset is less significant than the classification quality. This fitness function assesses the goodness of each particle position.

IV. Results and Discussion of Feature Selection

PSO is a random walk algorithm, and it was tested against two testing datasets using multiple runs to find the best feature subset. In Table V, the first column lists the class-specific features proposed by the TLFSA approach, and the second column lists the features chosen by the Binary Bat Algorithm (BBA) [57]. So, every class consists of the same number of features.

TABLE V. The BPSO and BBA Proposed Feature Subsets

Class	Class-specific features	
	TLFSA	BBA [18]
Normal	F5, F19, F22 AND F25	F3, F5, F16, F25, F28 AND F38
Probe	F8, F12, F32 AND F36	F5, F6, F8, F13 AND F35
DoS	F6, F10, F16 AND F26	F5, F6, F8, F24, F31, F32 AND F32
U2R	F6, F9, F23 AND F24	F5, F9, F12 AND F29
R2L	F6, F11, F35, F39 AND F41	F1, F6, F11, F12, F27, F35 AND F38

The accuracy with which the TLFSA and BBA proposed class-specific feature subsets could distinguish between the five classes of network traffic is shown in Table VI. From the NSL-KDD, two testing datasets with 7776 connections each were chosen at random. Table II provides information about the datasets used for training and testing. This comparative evaluation aimed to find the best minimal feature subsets that can be applied by any classifier at random. The BBA classifier was chosen for the comparison because many other researchers have used this method [57]-[59]. The radial basis function was the kernel type used for classification, and 10 cross-validations were carried out. The experiments were conducted using 10-fold cross-validation, and the dataset was split into training and testing subsets. The cross-validation ensured the robustness of the results, minimizing overfitting.

TABLE VI. TLFSA and BBA's Proposed Feature Subsets' Accuracy Rates (%)

Data	Normal		Probe		DoS		U2R		R2L	
	TLFSA	BBA	TLFSA	BBA	TLFSA	BBA	TLFSA	BBA	TLFSA	BBA
Set1	97.77	96.66	94.44	92	98.88	93.33	93.33	92.66	96.66	92
Set2	97.77	92.66	94.44	92.22	97.77	92.66	93.33	92.66	96.66	92.66
Average	97.77	94.66	94.44	92.11	98.33	92.99	93.33	92.66	96.66	92.33

To validate the statistical significance of these results, a t-test was conducted, showing that the differences between the accuracy rates of TLFSA and other methods were statistically significant ($p < 0.05$), confirming the superiority of the TLFSA approach. While this study compares TLFSA with GA, GSA, and BBA, future work should consider a broader set of feature selection algorithms, including recent advancements in machine learning-based feature selection methods.

Overall, the Normal, Probe U2R, R2L, and DoS classes have higher average accuracy rates once using the feature subsets proposed by TLFSA than when using the feature subsets proposed by BBA. The datasets used for the training and testing feature subsets are displayed in Table VII.

The experiment then compared the performance of feature subsets proposed by [44][60]. The genetic algorithm GA was proposed by [60], who claimed that 32 significant features were sufficient to classify a network connection. While Gauthama Raman, et al. [44] proposed the gravitational search algorithm GSA and claimed that 20 significant features were sufficient to categorize a network connection.

TABLE VII. Performance Evaluation Concerning other Feature Subsets

Class	Data	TLFSA	GA	GSA
Normal	Set1	97.77	91.11	95.55
	Set2	97.77	90	96.66
	Average	97.77	90.56	96.11
Probe	Set1	94.44	91.11	93.33
	Set2	94.44	90	92.22
	Average	94.44	90.56	92.76
DoS	Set1	98.88	90	97.77
	Set2	97.77	91.11	96.66
	Average	98.33	90.56	97.22
U2R	Set1	93.33	94	92.22
	Set2	93.33	92.11	94.44
	Average	93.3	93.06	93.33
R2L	Set1	96.66	93.33	92.22
	Set2	96.66	92.11	96
	Average	96.66	92.72	94.11

As shown in Table VIII and Fig. 3, the TLFSA method achieved consistently higher classification accuracy compared to BBA across all traffic classes, with an average accuracy of 97.22% compared to BBA's 91.11%.

Specifically, TLFSA outperformed BBA by a significant margin in the Probe and DoS classes, reflecting its robustness in distinguishing between different types of network intrusions.

The results presented in Table VIII indicate that the TLFSA approach is superior to BBA in terms of classification accuracy for network intrusion detection across multiple traffic classes. The most notable improvement was observed in the DoS class, where TLFSA achieved an accuracy of 98.88%, while BBA only reached 93.33%. This suggests that TLFSA is particularly effective at identifying DoS attacks, likely due to its ability to reduce the feature set to the most relevant attributes, which enhances detection precision. Similarly, for the Probe class, TLFSA provided a significant accuracy boost (94.44% vs. 92.00%), highlighting its ability to handle various types of malicious network activities. Overall, the performance improvements demonstrated by TLFSA across all classes confirm its potential as a more reliable feature selection method compared to BBA.

Feature subsets created by GA and feature subsets created by GSA were compared with the suggested method. The classification performance of these feature subsets is shown in Table VII and Fig. 3 based on their experiment using 7776 randomly selected data points from five classes. Table VIII's performance comparison demonstrates that, overall, the class-specific feature subsets proposed in this study's discrimination capability is superior to that of the feature subsets proposed by [44][60]. Reducing the feature set has direct practical implications, improving computational efficiency and enabling real-time analysis in large-scale IDS deployments. The reduced model size also enhances interpretability, making the approach more suitable for operational environments.

Fig. 4 compares the performance of TLFSA and BBA in terms of Precision, Recall, and F1-score across five traffic classes: Normal, Probe, DoS, U2R, and R2L. The results align with the accuracy data from Table VIII, showing that TLFSA consistently outperforms BBA across all metrics, especially in classes like DoS and Probe, where the differences are most pronounced.

- **Precision and Recall**: TLFSA shows higher precision and recall for all classes, particularly in **DoS** (Precision: 0.99 vs. 0.94, Recall: 0.98 vs. 0.93) and **Normal** traffic. This suggests TLFSA is better at minimizing false positives and capturing true positives.
- **F1-Score**: TLFSA maintains higher F1-scores across the board, indicating a better balance between precision and recall. The greatest advantage is seen in **DoS** traffic (F1: 0.98 vs. 0.93), aligning with the higher accuracy seen in Table VIII.

Overall, **TLFSA** provides superior performance compared to BBA, particularly in detecting frequent attacks like DoS and Probe, while also slightly outperforming BBA in harder-to-detect classes like U2R and R2L.

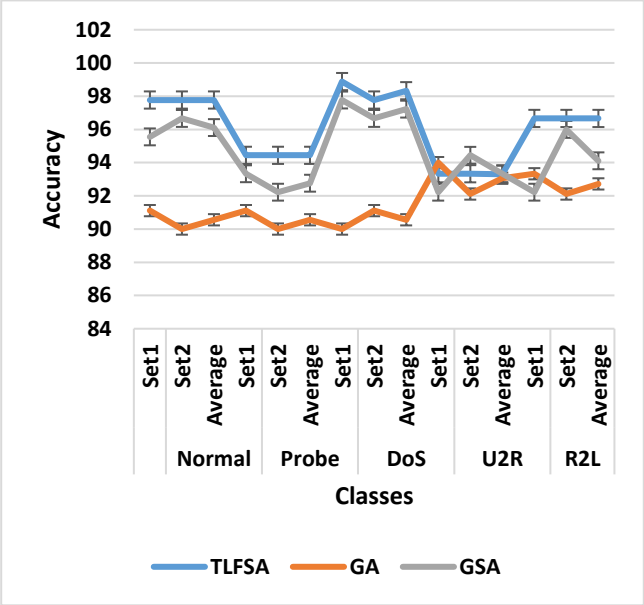

Fig. 3. The classification performance of the two subsets

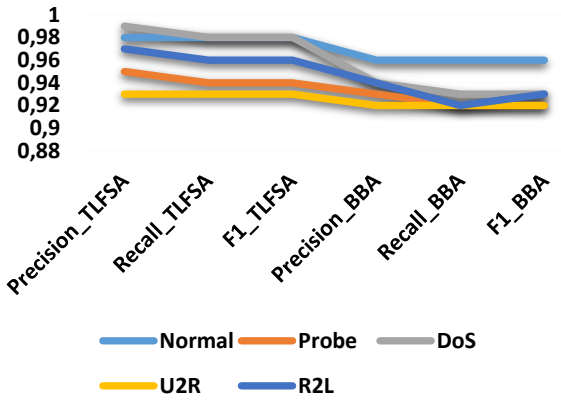

Fig. 4. Comparison of Precision, Recall, and F1-Score Between TLFSA and BBA Methods Across Network Traffic Classes (Normal, Probe, DoS, U2R, and R2L)

TABLE VIII. COMPARISON OF CLASSIFICATION ACCURACY (%) BETWEEN TLFSA AND BBA METHODS ACROSS DIFFERENT TRAFFIC CLASSES (NORMAL, PROBE, DOS, U2R, AND R2L)

Class	Precision TLFSA	Recall TLFSA	F1 TLFSA	Precision BBA	Recall BBA	F1 BBA
Normal	0.98	0.98	0.98	0.96	0.96	0.96
Probe	0.95	0.94	0.94	0.93	0.92	0.92
DoS	0.99	0.98	0.98	0.94	0.93	0.93
U2R	0.93	0.93	0.93	0.92	0.92	0.92
R2L	0.97	0.96	0.96	0.94	0.92	0.93

The NSL-KDD dataset was split into Set1 and Set2. These subsets reflect typical network traffic patterns, and the distribution of attack types, including the low-frequency U2R and R2L attacks, mirrors the real-world imbalances found in network traffic. The 41 potential feature candidates were successfully reduced to Four and Five features using the feature selection methodology used in this study. The selection of four to five features was based on an empirical analysis showing that these features provide sufficient discriminatory power across all traffic classes. The reduction in computational complexity further supports this without compromising accuracy. Four features are necessary for the Normal, Probe, U2R, and DoS classes to reveal their characteristics. Meanwhile, five features can be used to represent network traffic that falls under the R2L class. The comparative study has proven that classification performance is significantly better when using the suggested class-specific feature sets as input than when using a generic feature set.

This result is in line with what was noticed with GA and GSA. In addition, a feature-based recognizer requires less training time than a classifier that uses all features, as shown in Table IX.

TABLE IX. GENERIC FEATURE-BASED CLASSIFICATION PERFORMANCE

	TLFSA	**GA**	**GSA**
Set1	97.77	91.11	97.77
Set2	96.66	91.11	95.55
Average	**97.22**	**91.11**	**96.66**

Throughout this study, the duplicative and clashing features were removed using the Rough Set technique, and they were then refined using the Binary Particle Swarm Optimization technique. Class-specific feature subsets have been produced by BPSO using this hierarchical structure. This small set of features could reduce the data size from 41 to just four or five features based on the specific traffic classes. According to the experimental finding, the application of feature selection provides a significant improvement of 0.99% in classification accuracy compared to GSA. Only 15% of the features are necessary to represent any network traffic connection accurately. This small feature set has led to a classifier design that is easier to train and is compact.

V. CONCLUSIONS

This study proposes a two-level feature selection strategy combining Rough Set Theory and Binary Particle Swarm Optimization to enhance the accuracy and efficiency of Intrusion Detection Systems. The results show measurable improvements in classification accuracy and computational efficiency compared to traditional methods like Genetic Algorithm and Gravitational Search Algorithm. However, further validation on diverse datasets and with statistical significance testing is necessary to confirm these improvements and avoid overgeneralization.

While the TLFSA method demonstrated strong performance on the NSL-KDD dataset, future work should apply the method to more diverse datasets, such as modern network traffic datasets, to validate its scalability and generalizability across different network environments. The practical benefits of reduced computational overhead and improved scalability make it promising for real-world IDS applications, but the method's adaptability to dynamic network conditions and integration challenges requires further exploration.

Moreover, while feature reduction to about 15% of the original set was effective, more detailed comparisons with recent feature selection techniques would provide stronger validation.

In conclusion, while the proposed method offers significant potential for improving IDS performance, future research should focus on testing its applicability across diverse datasets and addressing limitations related to scalability and adaptability to evolving security threats.

ACKNOWLEDGMENT

I acknowledge the initial support received from Shaqra University. This support played a vital role in facilitating this research.

REFERENCES

[1] A. Al-Nawasrah, A. A. Almomani, S. Atawneh, and M. Alauthman, "A survey of fast flux botnet detection with fast flux cloud computing," *International Journal of Cloud Applications and Computing (IJCAC)*, vol. 10, no. 3, pp. 17-53, 2020.

[2] W. Alomoush, A. Alrosan, A. Almomani, K. Alissa, O. A. Khashan, and A. Al-Nawasrah, "Spatial information of fuzzy clustering based mean best artificial bee colony algorithm for phantom brain image segmentation," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 11, no. 5, pp. 4050-4058, 2021.

[3] H. A. Al Issa, M. H. Al-Jarah, A. Almomani, and A. Al-Nawasrah, "Encryption and decryption cloud computing data based on XOR and genetic algorithm," *International Journal of Cloud Applications and Computing (IJCAC)*, vol. 12, no. 1, pp. 1-10, 2022.

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

[5] Z. Yu, J. J. Tsai, and T. Weigert, "An adaptive automatically tuning intrusion detection system," *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, vol. 3, no. 3, pp. 1-25, 2008.

[6] E. Anthi, L. Williams, and P. Burnap, "Pulse: an adaptive intrusion detection for the internet of things," *ET Conference Proceedings*, 2018.

[7] G. Fernandes, J. J. Rodrigues, L. F. Carvalho, J. F. Al-Muhtadi, and M. L. Proença, "A comprehensive survey on network anomaly detection," *Telecommunication Systems*, vol. 70, no. 3, pp. 447-489, 2019.

[8] A. Al-Nawasrah, A. Al-Momani, F. Meziane and M. Alauthman, "Fast flux botnet detection framework using adaptive dynamic evolving spiking neural network algorithm," *2018 9th International Conference on Information and Communication Systems (ICICS)*, pp. 7-11, 2018.

[9] A. Al-Nawasrah *et al*., "Botnet Attack Detection Using A Hybrid Supervised Fast-Flux Killer System," in *Journal of Web Engineering*, vol. 21, no. 2, pp. 179-202, 2022.

[10] A. Almomani, A. Al-Nawasrah, M. Alauthman, M. A. Al-Betar, and F. Meziane, "Botnet detection used fast-flux technique, based on adaptive dynamic evolving spiking neural network algorithm," *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 36, no. 1, pp. 50-65, 2021.

[11] A. S. Almogren, "Intrusion detection in Edge-of-Things computing," *Journal of Parallel Distributed Computing*, vol. 137, pp. 259-265, 2020.

[12] R. Vinayakumar, K. P. Soman, and P. Poornachandran, "A comparative analysis of deep learning approaches for network intrusion detection systems (N-IDSs): deep learning for N-IDSs," *International Journal of Digital Crime and Forensics (IJDCF)*, vol. 11, no. 3, pp. 65-89, 2019.

[13] P. Sun *et al*. "DL-IDS: Extracting Features Using CNN-LSTM Hybrid Network for Intrusion Detection System," *Security and communication networks*, vol. 2020, no. 1, 8890306, 2020.

[14] S. Seth, K. K. Chahal, and G. Singh, "Concept Drift–Based Intrusion Detection For Evolving Data Stream Classification In IDS: Approaches And Comparative Study," *The Computer Journal*, 2024.

[15] V. G. Krishnan, P. V. Lakshmi, A. N. Julaiha, S. L. Jemina, A. Sunitha, and V. Divya, "Vortex Search Algorithm based Machine Learning Classification for IDS," *2022 6th International Conference on Trends in Electronics and Informatics (ICOEI)*, pp. 1366-1372, 2022.

[16] M. Eskandari, Z. H. Janjua, M. Vecchio, and F. Antonelli, "Passban IDS: An Intelligent Anomaly-Based Intrusion Detection System for IoT Edge Devices," in *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 6882-6897, 2020.

[17] S. Krishnaveni, S. Sivamohan, S. Sridhar, and S. Prabhakaran, "Network intrusion detection based on ensemble classification and feature selection method for cloud computing," *Concurrency and Computation: Practice and Experience*, vol. 34, no. 11, p. e6838, 2022.

[18] M. Prasad, S. Tripathi, and K. Dahal, "An efficient feature selection based Bayesian and Rough set approach for intrusion detection," *Applied Soft Computing*, vol. 87, p. 105980, 2020.

[19] M. A. Siddiqi and W. Pak, "Optimizing filter-based feature selection method flow for intrusion detection system," *Electronics*, vol. 9, no. 12, p. 2114, 2020.

[20] S. Aljawarneh, M. Aldwairi, and M. B. Yassein, "Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model," *Journal of Computational Science*, vol. 25, pp. 152-160, 2018.

[21] Y. Xuan *et al*., "Multi-Model Fusion Short-Term Load Forecasting Based on Random Forest Feature Selection and Hybrid Neural Network," in *IEEE Access*, vol. 9, pp. 69002-69009, 2021.

[22] X. -F. Song, Y. Zhang, Y. -N. Guo, X. -Y. Sun, and Y. -L. Wang, "Variable-Size Cooperative Coevolutionary Particle Swarm Optimization for Feature Selection on High-Dimensional Data," in *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 5, pp. 882-895, 2020.

[23] A. D. Li, B. Xue, and M. Zhang, "Improved binary particle swarm optimization for feature selection with new initialization and search space reduction strategies," *Applied Soft Computing*, vol. 106, p. 107302, 2021.

[24] A. Shahraki, M. Abbasi, and Ø. Haugen, "Boosting algorithms for network intrusion detection: A comparative evaluation of Real AdaBoost, Gentle AdaBoost and Modest AdaBoost," *Engineering Applications of Artificial Intelligence*, vol. 94, p. 103770, 2020.

[25] L. Ashiku and C. Dagli, "Network intrusion detection system using deep learning," *Procedia Computer Science*, vol. 185, pp. 239-247, 2021.

[26] M. Masdari and H. Khezri, "Towards fuzzy anomaly detection-based security: a comprehensive review," *Fuzzy Optimization and Decision Making*, vol. 20, no. 1, pp. 1-49, 2021.

[27] Z. Halim *et al*., "An effective genetic algorithm-based feature selection method for intrusion detection systems," *Computers & Security*, vol. 110, p. 102448, 2021.

[28] S. Thakur, A. Chakraborty, R. De, N. Kumar, and R. Sarkar, "Intrusion detection in cyber-physical systems using a generic and domain specific deep autoencoder model," *Computers & Electrical Engineering*, vol. 91, p. 107044, 2021.

[29] M. Samadi Bonab, A. Ghaffari, F. Soleimanian Gharehchopogh, and P. Alemi, "A wrapper-based feature selection for improving performance of intrusion detection systems," *International Journal of Communication Systems*, vol. 33, no. 12, p. e4434, 2020.

[30] A. Thakkar and R. Lohiya, "A survey on intrusion detection system: feature selection, model, performance measures, application perspective, challenges, and future research directions," *Artificial Intelligence Review*, vol. 55, no. 1, pp. 453-563, 2022.

[31] S. M. Kasongo and Y. Sun, "Performance analysis of intrusion detection systems using a feature selection method on the UNSW-NB15 dataset," *Journal of Big Data*, vol. 7, pp. 1-20, 2020.

[32] M. Prasad, R. K. Gupta, and S. Tripathi, "A multi-level correlation-based feature selection for intrusion detection," *Arabian Journal for Science and Engineering*, vol. 47, no. 8, pp. 10719-10729, 2022.

[33] O. Osanaiye, O. Ogundile, F. Aina, and A. Periola, "Feature selection for intrusion detection system in a cluster-based heterogeneous wireless sensor network," *Facta Universitatis, Series: Electronics and Energetics*, vol. 32, no. 2, pp. 315-330, 2019.

[34] M. Artur, "Review the performance of the Bernoulli Naïve Bayes Classifier in Intrusion Detection Systems using Recursive Feature Elimination with Cross-validated selection of the best number of features," *Procedia computer science*, vol. 190, pp. 564-570, 2021.

[35] F. Moslehi and A. Haeri, "A novel hybrid wrapper–filter approach based on genetic algorithm, particle swarm optimization for feature subset selection," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, no. 3, pp. 1105-1127, 2020.

[36] J. Gu and S. Lu, "An effective intrusion detection approach using SVM with naïve Bayes feature embedding," *Computers & Security*, vol. 103, p. 102158, 2021.

[37] M. M. Hassan, A. Gumaei, A. Alsanad, M. Alrubaian, and G. Fortino, "A hybrid deep learning model for efficient intrusion detection in big data environment," *Information Sciences*, vol. 513, pp. 386-396, 2020.

[38] K. Albulayhi, Q. Abu Al-Haija, S. A. Alsuhibany, A. A. Jillepalli, M. Ashrafuzzaman, and F. Sheldon, "IoT intrusion detection using machine learning with a novel high performing feature selection method," *Applied Sciences*, vol. 12, no. 10, p. 5015, 2022.

[39] Y. Zhang, X. Shi, S. Zhang, and A. Abraham, "A XGBoost-Based Lane Change Prediction on Time Series Data Using Feature Engineering for Autopilot Vehicles," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 10, pp. 19187-19200, 2022.

[40] M. A. Rahman *et al*., "Effective combining of feature selection techniques for machine learning-enabled IoT intrusion detection," *Multimedia Tools and Applications*, pp. 1-19, 2021.

[41] R. Bhatia, S. Benno, J. Esteban, T. V. Lakshman, and J. Grogan, "Unsupervised machine learning for network-centric anomaly detection in IoT," In *Proceedings of the 3rd acm conext workshop on big data, machine learning and artificial intelligence for data communication networks*, pp. 42-48, 2019.

[42] C. Khammassi and S. Krichen, "A GA-LR wrapper approach for feature selection in network intrusion detection," *computers & security*, vol. 70, pp. 255-277, 2017.

[43] N. AlNuaimi, M. M. Masud, M. A. Serhani, and N. Zaki, "Streaming feature selection algorithms for big data: A survey," *Applied Computing and Informatics*, vol. 18, no. 1/2, pp. 113-135, 2020.

[44] M. Gauthama Raman *et al*., "An efficient intrusion detection technique based on support vector machine and improved binary gravitational search algorithm," *Artificial Intelligence Review*, vol. 53, no. 5, pp. 3255-3286, 2020.

[45] Y. Li, M. Jia, X. Han, and X. S. Bai, "Towards a comprehensive optimization of engine efficiency and emissions by coupling artificial neural network (ANN) with genetic algorithm (GA)," *Energy*, vol. 225, p. 120331, 2021.

[46] B. Selvakumar and K. Muneeswaran, and Security, "Firefly algorithm based feature selection for network intrusion detection," *Computers & Security*, vol. 81, pp. 148-155, 2019.

[47] K. A. Taher, B. M. Y. Jisan, and M. M. Rahman, "Network Intrusion Detection using Supervised Machine Learning Technique with Feature Selection," *2019 International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST)*, pp. 643-646, 2019.

[48] S. Mohammadi, H. Mirvaziri, M. Ghazizadeh-Ahsaee, and H. Karimipour, "Cyber intrusion detection by combined feature selection algorithm," *Journal of information security and applications*, vol. 44, pp. 80-88, 2019.

[49] F. H. Almasoudy, W. L. Al-Yaseen, and A. K. Idrees, "Differential evolution wrapper feature selection for intrusion detection system," *Procedia Computer Science*, vol. 167, pp. 1230-1239, 2020.

[50] K. Asghari, M. Masdari, F. S. Gharehchopogh, and R. Saneifard, "Multi-swarm and chaotic whale-particle swarm optimization algorithm with a selection method based on roulette wheel," *Expert Systems*, vol. 38, no. 8, p. e12779, 2021.

[51] H. Li, X. Hu, X. Zhang, S. Wei, and Q. Luo, "Kinematic parameters calibration of industrial robot based on RWS-PSO algorithm," *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, vol. 237, no. 14, pp. 3210-3220, 2023.

[52] B. Nouri-Moghaddam, M. Ghazanfari, and M. Fathian, "A novel multi-objective forest optimization algorithm for wrapper feature selection," *Expert Systems with Applications*, vol. 175, p. 114737, 2021.

[53] Y. He *et al.*, "A Sparse Protocol Parsing Method for IIoT Based on BPSO-vote-HMM Hybrid Model," in *IEEE/ACM Transactions on Networking*, vol. 31, no. 2, pp. 485-496, 2023.

[54] S. Ajibade, "Particle Swarm Optimization with Chaotic Dynamic Weight for Feature Selection Enhancement," *Engineering Technology and Management*, vol. 1, no. 2, pp. 1-5, 2020.

[55] S. -S. M. Ajibade, N. B. Binti Ahmad and A. Zainal, "A Hybrid Chaotic Particle Swarm Optimization with Differential Evolution for feature selection," *2020 IEEE Symposium on Industrial Electronics & Applications (ISIEA)*, pp. 1-6, 2020.

[56] C. Bae, W.-C. Yeh, Y. Y. Chung, and S. L. Liu, "Feature selection with intelligent dynamic swarm and rough set," *Expert Systems with Applications*, vol. 37, no. 10, pp. 7026-7032, 2010.

[57] S. Maheswari and K. Arunesh, " Unsupervised Binary BAT algorithm based Network Intrusion Detection System using enhanced multiple classifiers," in *2020 International Conference on Smart Electronics and Communication (ICOSEC)*, pp. 885-889, 2020.

[58] N. M. Yusof, A. K. Muda, S. F. Pratama, and F. T. K. E. Elektronik, "A New Binary WOA-BAT Feature Selection Approach for Amphetamine-type Stimulants Drug Classification," *Manuscript Editor*, vol. 2021, p. 26, 2021.

[59] W. Z. Al-Dyani, F. K. Ahmad, and S. S. Kamaruddin, "Binary Bat Algorithm for text feature selection in news events detection model using Markov clustering," *Cogent Engineering*, vol. 9, no. 1, p. 2010923, 2022.

[60] N. Kunhare, R. Tiwari, and J. Dhar, "Intrusion detection system using hybrid classifiers with meta-heuristic algorithms for the optimization and feature selection by genetic algorithm," *Computers and Electrical Engineering*, vol. 103, p. 108383, 2022.

[61] A. Al Nawasrah. *Fast flux botnet detection based on adaptive dynamic evolving spiking neural network*. University of Salford (United Kingdom), 2018.

[62] F. Zhao, H. Li, K. Niu, J. Shi, and R. Song, "Application of deep learning-based intrusion detection system (IDS) in network anomaly traffic detection," *Appl. Comput. Eng*, vol. 86, pp. 231-237, 2024.

[63] A. Almomani, A. Al-Nawasrah, W. Alomoush, M. Al-Abweh, A. Alrosan, and B. B. Gupta, "Information management and IoT technology for safety and security of smart home and farm systems," *Journal of Global Information Management (JGIM)*, vol. 29, no. 6, pp. 1-23, 2021.

[64] A. Nazir and R. A. Khan, "A novel combinatorial optimization based feature selection method for network intrusion detection," *Computers & Security*, vol. 102, p. 102164, 2021.

[65] S. Baghirzada. *Feature Selection with Improved Mountain Gazelle Optimizer Algorithm for Intrusion Detection Systems*. (Master's thesis, Khazar University (Azerbaijan)), 2024.

[66] K. Bian and R. Priyadarshi, "Machine learning optimization techniques: a Survey, classification, challenges, and Future Research Issues," *Archives of Computational Methods in Engineering*, pp. 1-25, 2024.

[67] M. Cherrington, D. Airehrour, J. Lu, F. Thabtah, Q. Xu, and S. Madanian, "Particle Swarm Optimization for Feature Selection: A Review of Filter-based Classification to Identify Challenges and Opportunities," *2019 IEEE 10th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, pp. 0523-0529, 2019, doi: 10.1109/IEMCON.2019.8936185.

[68] A. Pawar and N. Tiwari, "A Novel Approach of DDOS Attack Classification with Optimizing the Ensemble Classifier Using A Hybrid Firefly and Particle Swarm Optimization (HFPSO)," *International Journal of Intelligent Engineering & Systems*, vol. 16, no. 4, 2023.

[69] M. A. Shyaa, N. F. Ibrahim, Z. Zainol, R. Abdullah, M. Anbar, and L. Alzubaidi, "Evolving cybersecurity frontiers: A comprehensive survey on concept drift and feature dynamics aware machine and deep learning in intrusion detection systems," *Engineering Applications of Artificial Intelligence*, vol. 137, p. 109143, 2024.

[70] S. He, Q. H. Wu, and J. R. Saunders, "Group Search Optimizer: An Optimization Algorithm Inspired by Animal Searching Behavior," in *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 5, pp. 973-990, Oct. 2009, doi: 10.1109/TEVC.2009.2011992.

[71] E. Cuevas, M. Cienfuegos, D. Zaldívar, and M. Pérez-Cisneros, "A swarm optimization algorithm inspired in the behavior of the social-spider," *Expert Systems with Applications*, vol. 40, no. 16, pp. 6374-6384, 2013.

[72] M. Heigl, E. Weigelt, D. Fiala, and M. Schramm, "Unsupervised feature selection for outlier detection on streaming data to enhance network security," *Applied Sciences*, vol. 11, no. 24, p. 12073, 2021.

[73] Y. Zheng, Z. Li, X. Xu, and Q. Zhao, "Dynamic defenses in cyber security: Techniques, methods and challenges," *Digital Communications and Networks*, vol. 8, no. 4, pp. 422-435, 2022.

[74] I. K. Thajeel, K. Samsudin, S. J. Hashim, and F. Hashim, "Dynamic feature selection model for adaptive cross site scripting attack detection using developed multi-agent deep Q learning model," *Journal of King Saud University-Computer and Information Sciences*, vol. 35, no. 6, p. 101490, 2023.

[75] A. I. Madbouly and T. M. Barakat, "Enhanced relevant feature selection model for intrusion detection systems," *International Journal of Intelligent Engineering Informatics*, vol. 4, no. 1, pp. 21-45, 2016.

[76] M. Torabi, N. I. Udzir, M. T. Abdullah, and R. Yaakob, "A review on feature selection and ensemble techniques for intrusion detection system," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 5, 2021.

[77] S. Alabdulwahab and B. Moon, "Feature selection methods simultaneously improve the detection accuracy and model building time of machine learning classifiers," *Symmetry*, vol. 12, no. 9, p. 1424, 2020.

[78] A. Alazab, M. Hobbs, J. Abawajy, and M. Alazab, "Using feature selection for intrusion detection system," *2012 International Symposium on Communications and Information Technologies (ISCIT)*, pp. 296-301, 2012, doi: 10.1109/ISCIT.2012.6380910.

[79] L. Brezočnik, I. Fister Jr, and V. Podgorelec, "Swarm intelligence algorithms for feature selection: a review," *Applied Sciences*, vol. 8, no. 9, p. 1521, 2018.

[80] Y. Zhou, G. Cheng, S. Jiang, and M. Dai, "Building an efficient intrusion detection system based on feature selection and ensemble classifier," *Computer networks*, vol. 174, p. 107247, 2020.