STAP
Smart Technologies Academic Press

Journal of Cyber Security and Risk Auditing

https://www.jcsra.thestap.com/

Leveraging ACO, GA, and GWO for Enhancing Port Scan Attack Detection Using Machine Learning

Mohammed Amin Almaiah[1], **Rajan Kadel**[2]

[1]King Abdullah the II IT School, The University of Jordan, Amman 11942, Jordan
[2]School of IT and Engineering (SITE), Melbourne Institute of Technology (MIT), Melbourne, VIC 3000, Australia

ARTICLE INFO

Article History

Received: 17-05-2025
Revised: 30-07-2025
Accepted: 20-08-2025
Published: 27-08-2025

Vol.2025, No.4

DOI:
https://doi.org/10.63180/jcsra.thestap.2025.4.9

Corresponding author.
Email:
m.almaiah@ju.edu.jo

Orcid:
https://orcid.org/0000-0003-0773-1597

ABSTRACT

Port scan attacks are commonly employed by malicious actors or automated tools to probe a system's network ports in search of open ports and potential vulnerabilities. These ports function as communication endpoints that allow services and applications to exchange data. While port scanning is often associated with malicious intent—such as mapping network structures, identifying running services, or preparing for subsequent attacks—it is not always harmful. In fact, cybersecurity professionals and system administrators regularly use port scanning as a diagnostic tool to identify and address system weaknesses. To protect against port scan attacks, organizations typically deploy a combination of firewalls, intrusion detection systems (IDS), and network monitoring tools to detect and block unauthorized scanning activities. Detecting port scans is a vital part of cybersecurity defense, enabling organizations to identify points of vulnerability, respond swiftly to incidents, and implement appropriate security measures. This proactive approach significantly reduces the risk of successful cyber intrusions. In our research, we propose a machine learning-based approach for detecting port scan attacks. The process begins with data collection, where network traffic data containing behavioral indicators of scanning activity is gathered. From this data, relevant features are extracted to train the model. Feature selection is then performed using metaheuristic algorithms such as Ant Colony Optimization (ACO), Genetic Algorithm (GA), and Gray Wolf Optimization (GWO), which help reduce computational complexity by selecting the most informative features. These selected features are then used to train machine learning models, including classifiers like Support Vector Machine (SVM) and K-Nearest Neighbors (KNN), to differentiate between benign and malicious activity. Finally, the performance of the trained models is assessed using evaluation metrics such as precision, recall, F1-score, and accuracy. The results of our experiments indicate that the proposed models are highly effective, achieving accuracy rates exceeding 99% across all tested configurations. In summary, port scan detection is essential for strengthening network defenses. By leveraging machine learning techniques and optimization-based feature selection, it is possible to detect and respond to port scanning behaviors with greater accuracy and efficiency.

Keywords: Internet of Things (IoT), Authentication, Blockchain, Security of IoT networks, Homomorphic Encryption Privacy.

How to cite the article
Almaiah, M. A., & Kadel, R. (2025). Leveraging ACO, GA, and GWO for Enhancing Port Scan Attack Detection Using Machine Learning. Journal of Cyber Security and Risk Auditing, 2025(4), 306–326.

1. Introduction

Port scan attacks are techniques used to probe computer systems or networks in order to identify open ports and associated services. By sending a series of data packets to various ports, malicious actors attempt to determine the status of each port and uncover potential system vulnerabilities [1]. These scans can take multiple forms, including TCP port scanning, UDP port scanning, covert scanning, and SYN flood attacks, all of which aim to gather intelligence that could be used for unauthorized access or more sophisticated future exploits [5]. The early detection and prevention of port scan attacks are essential components of robust network security strategies. Tools such as intrusion detection systems (IDS), firewalls, and continuous network monitoring play a pivotal role in identifying and halting suspicious activities before they escalate [4]. Timely detection of port scans contributes to proactive threat mitigation, allowing organizations to address vulnerabilities before they can be exploited [2]. This proactive stance not only limits potential damage but also enhances compliance with cybersecurity regulations, improves overall network efficiency, and supports a culture of security awareness and preparedness [3].

Port scanning, however, is not inherently malicious and can serve legitimate purposes. It is commonly used by cybersecurity professionals for reconnaissance, vulnerability assessment, security auditing, penetration testing, network mapping, and research [16]. Such scans help security teams evaluate existing protections, simulate potential threats, and improve defensive measures. Nonetheless, conducting port scans without proper authorization is considered both illegal and unethical, as it infringes on privacy and system integrity [17]. Therefore, while port scanning remains a critical tool in the cybersecurity arsenal, it must be employed responsibly and within legal boundaries to ensure it supports, rather than threatens, digital security.

The primary goals of identifying port scan attacks using feature selection and machine learning techniques are to accurately detect threats, categorize network activities, optimize resource usage, adapt to evolving attack strategies, enable real-time detection, reduce false positives, and ensure scalability [7]. Feature selection algorithms, when integrated with machine learning models, significantly enhance the accuracy and efficiency of port scan detection. This advancement allows organizations to respond more swiftly and effectively, minimizing the risk of security breaches [6]. However, detecting port scan attacks poses several challenges. Adversaries often use evasive tactics, such as encrypted traffic or obfuscation, to avoid detection, resulting in increased occurrences of false positives or false negatives [9]. The complexity and dynamism of modern networks further complicate the differentiation between benign and malicious activities, and the continuous evolution of attack vectors necessitates constant updates to detection methods. Additionally, deep inspection of traffic can impact system performance, while privacy concerns must also be considered. In the absence of granular data, the accuracy of detection mechanisms may be compromised [10]. Zero-day attacks and system overloads present further obstacles, potentially delaying detection or causing it to fail altogether. Despite these limitations, techniques such as behavioral analysis and continuous network monitoring can strengthen detection capabilities and overall system security [12] [14].

The impetus for detecting port scan activity is rooted in the critical need to preserve network security and defend IT infrastructure from unauthorized access [13]. Port scan detection helps identify threats, reveal vulnerabilities, and enable corrective actions to be taken. It also supports compliance with cybersecurity regulations, improves operational efficiency, and enhances incident response and forensic investigations, thereby increasing organizational awareness and resilience [4] [15]. Machine learning and deep learning approaches to port scan detection typically involve feature selection and model training to classify network traffic as either normal or indicative of a scan. These approaches include supervised learning using labeled data, unsupervised learning for anomaly detection, neural networks such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), ensemble methods that aggregate multiple models, and online learning techniques for real-time detection [18]. The success of these approaches depends heavily on the quality of training data, the relevance of the selected features, and the suitability of the algorithms used. Continuous monitoring and periodic reevaluation are essential to maintain high accuracy and adaptability to emerging threats [19].

In our study, port scan detection is implemented by employing feature selection algorithms—such as Ant Colony Optimization (ACO), Genetic Algorithm (GA), and Gray Wolf Optimization (GWO)—alongside machine learning classifiers like Support Vector Machine (SVM) and k-Nearest Neighbors (KNN). The process begins with collecting raw network traffic data and extracting key features, including IP addresses, port numbers, and packet-level attributes. These features are then refined through selection algorithms to prepare the data for modeling. Subsequently, machine learning models are trained using SVM and KNN, and their performance is evaluated using relevant metrics. The models are iteratively refined to enhance their effectiveness and then deployed for real-time detection. Sustained monitoring and updates are vital to ensure the ongoing reliability of the system, with success hinging on high-quality training datasets, robust feature selection, appropriate model choice, and continuous performance assessment.

When applying optimization techniques such as Ant Colony Optimization (ACO), Genetic Algorithm (GA), and Grey Wolf Optimizer (GWO) in conjunction with machine learning (ML) methods for port scan detection, several critical research questions emerge. RQ1: How can ACO be leveraged to enhance feature selection in the context of port scan detection? RQ2: What are the specific advantages of GA in optimizing feature selection for identifying port scan attacks? RQ3: In what ways does GWO contribute to improving feature selection performance for port scan detection? RQ4: How can machine learning models be effectively integrated with these optimization algorithms to boost the accuracy and efficiency of detecting port scans? RQ5: What is the overall impact of this integration on performance within complex feature spaces?

This research presents several key contributions. First, a comparative analysis is conducted to evaluate the performance of selected feature selection algorithms (ACO, GA, GWO) and machine learning classifiers (SVM, KNN) in terms of their accuracy, precision, and detection capabilities. This comparison identifies the most effective combinations and informs the selection of techniques for enhanced detection performance. Second, the study explores innovative combinations of feature selection methods with machine learning algorithms to uncover synergistic effects, aiming to outperform the accuracy and efficiency of standalone approaches. Third, it focuses on evaluation metrics specifically tailored to the nuances of port scan detection, such as precision, recall, F1-score, and overall accuracy, thereby enabling a nuanced performance assessment. Fourth, the research employs a real-world dataset derived from actual network environments containing both benign and port scan traffic. This ensures the relevance, credibility, and applicability of the findings, especially in dynamic network contexts with evolving attack patterns. Finally, the study proposes enhanced solutions—in terms of accuracy, computational efficiency, and adaptability providing valuable tools for network administrators and cybersecurity professionals to detect and respond to port scanning threats effectively.

2. Literature Review

Several researchers have explored the detection of port scan attacks by proposing various machine learning and artificial intelligence models to enhance network security. For example, the study by [11] focused on distinguishing between port scanning and Distributed Denial of Service (DDoS) attacks using machine learning techniques. The researchers compared multiple algorithms—including decision trees, k-nearest neighbors (KNN), support vector machines (SVM), and random forests—to evaluate their effectiveness in identifying and differentiating between these types of attacks. The study began by outlining the characteristics of port scans and DDoS attacks, emphasizing their potential impact on network security and the importance of robust detection mechanisms. A labeled dataset comprising normal traffic, port scanning activities, and DDoS instances was used for training and testing. Feature extraction was performed to isolate relevant attributes, and each algorithm was evaluated based on accuracy, precision, recall, and F1 score. In a related study, [12] proposed the use of artificial intelligence (AI) algorithms to detect and classify network traffic patterns associated with port scanning. The authors highlighted the importance of early detection in mitigating security risks and discussed their methodology, which involved collecting network traffic, extracting features such as source and destination ports, timing, and packet size, and applying AI techniques for classification. The study employed algorithms including decision trees, SVM, and neural networks, trained to recognize behavioral patterns indicative of port scans. Experimental results demonstrated the models' effectiveness based on standard performance metrics, and the authors also addressed practical concerns such as system scalability and adaptability to evolving attack techniques.

Another work by [1] examined the use of machine learning to detect probe attacks—specific forms of network intrusions aimed at gathering information without exploiting vulnerabilities. Their approach leveraged classification algorithms trained on datasets containing both benign and probe attack traffic to distinguish between normal and malicious behavior. The findings underscored the capability of machine learning models to accurately identify such reconnaissance activities, thereby reinforcing network defenses. Similarly, [2] introduced an enhanced intrusion detection system (IDS) targeting probe attacks. This system integrated both signature-based and anomaly-based detection techniques, utilizing a comprehensive set of features derived from packet headers and traffic behavior. Machine learning was employed to improve classification accuracy, and the system demonstrated a high detection rate with a reduced false positive rate. The study concluded that the hybrid IDS model significantly improved the ability to detect and respond to probe-based threats in computer networks. A study [3] conducted a comparative study on two side-channel attack techniques—Flush+Reload and Prime+Probe—targeting the Advanced Encryption Standard (AES) cryptographic algorithm. These attacks exploit cache access pattern leaks to infer secret keys used during AES encryption. The study employed machine learning strategies to analyze collected cache access patterns and train classification models to recover cryptographic keys. Their findings evaluated the effectiveness of both attack methods in terms of accuracy, efficiency, and robustness. The results revealed critical vulnerabilities in AES implementations and demonstrated how machine learning can support the identification and mitigation of such side-channel threats. In another study, researchers proposed an intrusion detection system (IDS) combining Particle Swarm Optimization (PSO) and Grey Wolf Optimization (GWO) with a Support Vector Machine (SVM) classifier [4]. This hybrid approach was designed to optimize the classifier's parameters, improving detection accuracy and reducing false positives. The system was validated using a dataset comprising network traffic features, and results showed significant improvements in identifying various intrusion types, confirming the efficacy of the PSO-GWO hybrid technique in enhancing SVM-based IDS performance. A recent research [5] explored the identification and remediation of zero-day vulnerabilities in software systems. These vulnerabilities, being unknown to vendors, pose significant risks as they remain unpatched until discovered. The paper outlined challenges in data acquisition, accurate assessment, and rapid mitigation. It highlighted the importance of collaboration among developers, researchers, and cybersecurity professionals to address these vulnerabilities proactively and improve software security.

A further study proposed a novel detection method for unauthorized access in wireless sensor networks (WSNs) using a hybrid approach of Whale Optimization Algorithm (WOA) and Artificial Bee Colony (ABC) to optimize a Convolutional Neural Network (CNN) [6]. This method aimed to address resource constraints in WSNs by fine-tuning CNN parameters, thereby improving classification accuracy. Experimental results confirmed the model's effectiveness in distinguishing between normal and malicious network traffic, demonstrating the advantages of integrating optimization algorithms with machine learning for WSN security. [7] Focused on detecting port scanning attacks using supervised machine learning classifiers. Their proposed technique analyzed network traffic to identify patterns indicative of scanning behavior. By training classifiers on labeled datasets containing benign and malicious traffic, the system effectively distinguished port scans, thereby offering a practical solution for early detection and mitigation of reconnaissance attacks. In another study, researchers evaluated an intrusion detection system for the Internet of Things (IoT) by analyzing the impact of various feature selection techniques on detection accuracy and computational efficiency [8]. Given the resource limitations in IoT environments, selecting relevant features is critical. The study found that proper feature selection not only enhanced IDS performance but also reduced processing overhead, emphasizing its importance in practical IoT deployments. Another work by [9] presented a comprehensive survey on the application of machine learning techniques for intrusion detection in wireless sensor networks. The paper reviewed numerous algorithms used to counter threats such as data injection, node compromise, and routing attacks. It also discussed inherent WSN challenges, including limited computational resources and the need for real-time processing. The survey provided critical insights into the suitability of various machine learning methods and identified potential areas for future exploration in this domain. Lastly, a study investigated the vulnerability of network attack detection systems based on the Random Forest algorithm to adversarial attacks [10]. These attacks aim to deceive machine learning models by manipulating input data. The paper examined the strategies used to subvert detection accuracy and assessed countermeasures to enhance system robustness. The findings illuminated existing weaknesses and proposed improvements for building resilient detection frameworks against such sophisticated threats.

3. Methodology

To detect port scan attacks using feature selection techniques and machine learning (ML) algorithms, the process involves several key steps. First, data collection is performed by gathering a dataset that comprises network traffic data, including both normal traffic instances and instances of port scan attacks. Second, feature selection is applied to identify the most relevant features that effectively distinguish between benign and malicious traffic. Third, feature extraction is carried out by retrieving selected attributes from the preprocessed dataset—such as IP addresses, port numbers, packet timing, packet size, and protocol type—that are critical for modeling network behavior. Fourth, appropriate machine learning algorithms are selected based on the problem requirements and the nature of the dataset. Fifth, model training is conducted by using the extracted features and labeled data to train the ML models to learn the underlying patterns associated with port scanning behavior. Finally, in the evaluation and validation phase, the performance of the trained models is assessed using metrics such as accuracy, precision, recall, F1 score, and AUC-ROC. The models are validated using separate testing datasets to determine their generalization ability and effectiveness in real-world scenarios. Figure 1 presents the research methodology steps.

3.1 Data Acquisition Description

In the ongoing efforts to defend against evolving network threats, Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS) serve as critical security mechanisms. However, the effectiveness of anomaly-based intrusion detection approaches is often limited by the lack of reliable datasets for rigorous testing and validation. A comparative review of eleven prominent datasets developed since 1998 indicates that many of them are now outdated, lacking in diversity, and fail to comprehensively cover a wide range of known attack types. Furthermore, many datasets anonymize payload data, reducing their utility for practical evaluation.

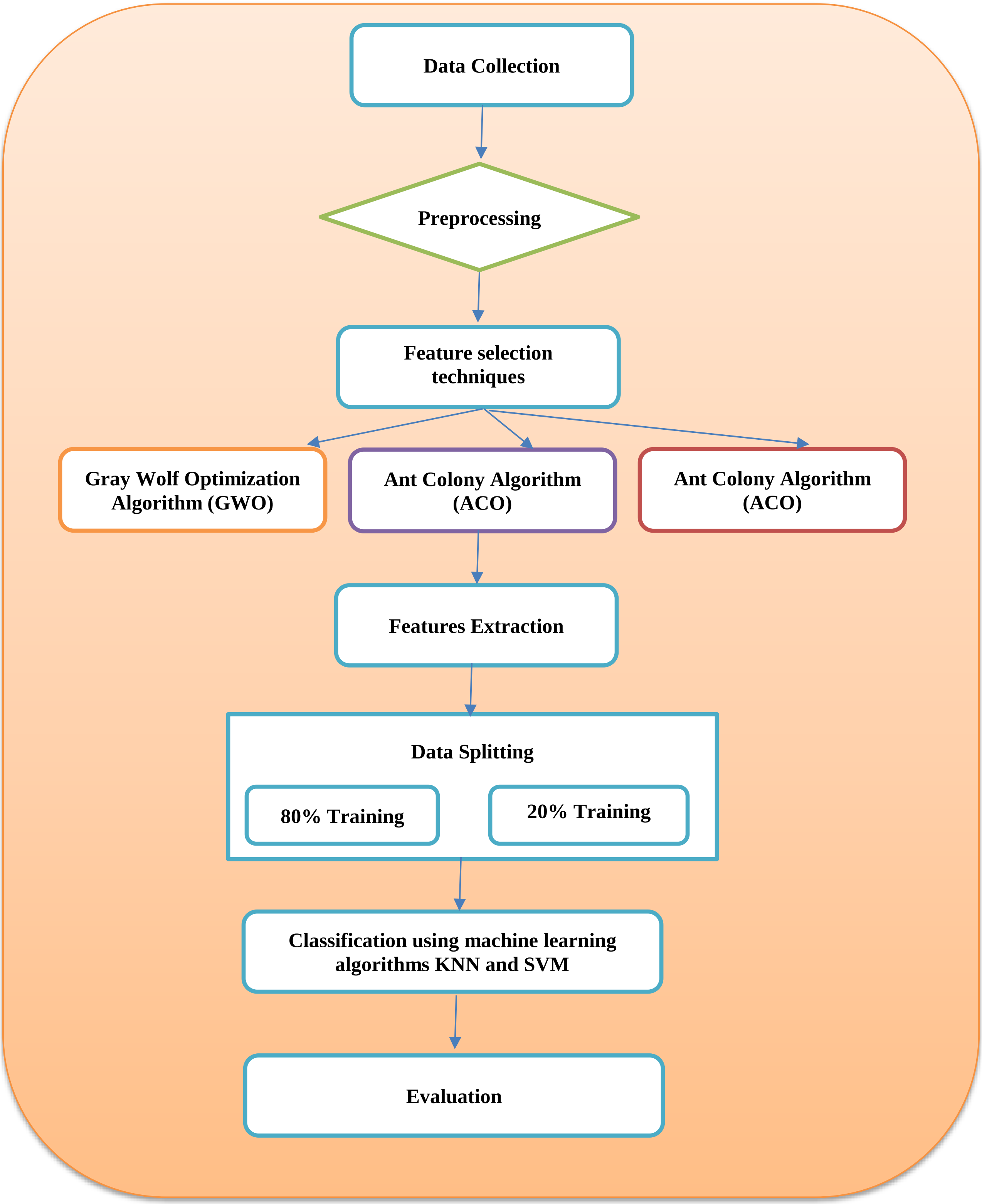

Data Collection
Preprocessing
Feature selection techniques
Gray Wolf Optimization Algorithm (GWO)
Ant Colony Algorithm (ACO)
Ant Colony Algorithm (ACO)
Features Extraction
Data Splitting
80% Training
20% Training
Classification using machine learning algorithms KNN and SVM
Evaluation

Figure 1. Research Methodology Steps.

In contrast, the CICIDS2017 dataset stands out as a benchmark for its realistic representation of contemporary network traffic and attack scenarios. It includes detailed labeled traffic flows with attributes such as timestamps, source and destination IP addresses, ports, protocols, and attack classifications—all organized in CSV format. A notable feature of this dataset is the use of the B-Profile system, which generates realistic background traffic based on human behavioral patterns.

The dataset simulates the behavior of 25 users across various network protocols and incorporates a wide range of attack types including Brute Force (FTP and SSH), Denial of Service (DoS), Web-based attacks, infiltration attempts, and Botnet activity. It adheres to eleven established criteria for benchmark datasets, providing a comprehensive network setup and multiple traffic sources. During the attack simulations, network traffic data were collected alongside memory dumps and system call traces from compromised hosts. Over 80 distinct flow-based features were extracted using the CICFlowMeter tool. Table 1 presents several types of datasets.

Table1. Datasets.

Datasets	Description
NSL-KDD	Refined version of KDD99dataset after removal of duplicate records
ISCX	Traffic from real world physical test environment
UNSW-NB15	55 features covering 10 types of attacks
CICIDS2(our dataset) 017	79 features with normal traffic and attacks

3.2 Feature Selection

Feature selection is a fundamental process in the detection of port scan attacks, as it helps isolate the most informative attributes that distinguish normal traffic from malicious scanning behavior. By automating the selection of relevant features, this process enhances model performance, reduces computational complexity, and mitigates overfitting. Several widely-used feature selection methods include: Mutual Information (MI): Measures the dependency between each feature and the target variable, quantifying their information gain. Information Gain (IG): Assesses how much information a feature contributes to reducing uncertainty in the classification outcome. Chi-Square Test: Evaluates the statistical independence between categorical variables, helping identify significant relationships between features and the class label. Recursive Feature Elimination (RFE): Iteratively removes the least important features based on model-specific criteria until an optimal subset is achieved. Wrapper Methods: Utilize predictive models to evaluate subsets of features, relying on search algorithms to identify the best-performing combinations. L1 Regularization (Lasso): Applies penalties to feature weights to encourage sparsity, thereby retaining only features with non-zero coefficients. Correlation-Based Feature Selection (CFS): Selects features that are highly correlated with the target variable while minimizing redundancy among selected features.

The choice of feature selection technique is influenced by factors such as dataset characteristics, feature dimensionality, and the desired balance between computational efficiency and model accuracy. In this study, we utilize three metaheuristic optimization algorithms for feature selection: Genetic Algorithm (GA), Ant Colony Optimization (ACO), and Grey Wolf Optimization (GWO). These algorithms are employed to identify the most relevant feature subsets that enhance the detection performance of the machine learning models.

3.3 Feature Selection Algorithms

3.3.1 Gray Wolf Optimization (GWO) algorithm

The Gray Wolf Optimization (GWO) algorithm is a metaheuristic technique inspired by the social hierarchy and hunting strategies of gray wolves in nature. While GWO is traditionally employed for solving complex optimization problems, it has proven highly effective in feature selection tasks, such as identifying the most relevant features for detecting port scan attacks in network traffic. In this context, each gray wolf in the population represents a potential subset of features. The algorithm begins by initializing key parameters including population size and the number of iterations. Wolves are randomly positioned in the multidimensional feature space, each encoding a unique feature subset. Figure 2 presents the Gray Wolf Optimization (GWO) algorithm steps.

During the hunting phase, wolves update their positions based on the behavior of the leading wolves—namely the alpha (best solution), beta (second-best), and delta (third-best) wolves. These leaders guide the search process by influencing the movement of the remaining wolves using mathematically modeled behaviors that simulate encircling prey and attacking it. This dynamic allows the algorithm to strike a balance between exploration (searching new areas of the feature space) and exploitation (refining known good solutions). At each iteration, a fitness function is used to evaluate the quality of each wolf's feature subset. This function measures how well the subset distinguishes between normal and malicious network traffic, helping to assess the effectiveness of the features in detecting port scans.

As the algorithm progresses, wolves adapt their positions with respect to the alpha, beta, and delta wolves, gradually converging towards the most promising regions of the feature space. Once a stopping criterion is met—typically a fixed number of iterations or convergence—the algorithm selects the feature subset associated with the alpha wolf as the optimal solution. This subset is considered the most effective for classifying port scan activities. Through its adaptive behavior and biologically inspired optimization process, GWO effectively identifies key features, thereby improving the performance and accuracy of machine learning models used in port scan detection.

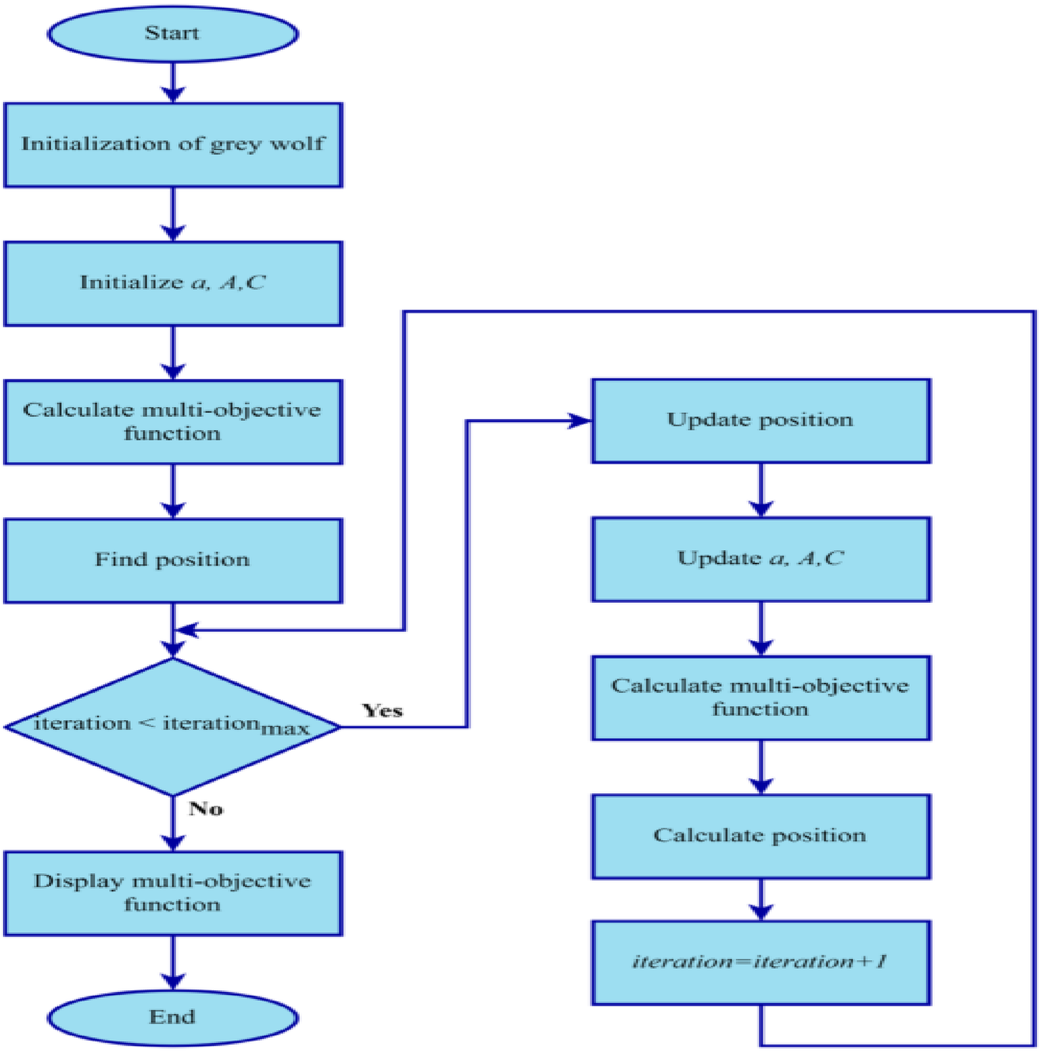

Start
Initialization of grey wolf
Initialize a, A, C
Calculate multi-objective function
Find position
iteration < iteration_max
Yes
No
Display multi-objective function
End
Update position
Update a, A, C
Calculate multi-objective function
Calculate position
iteration=iteration+1

Figure 2. Gray Wolf Optimization (GWO) algorithm [4]

The parameters commonly employed in our implementation of the Gray Wolf Optimization (GWO) algorithm include several key elements that guide the search process. Firstly, the Population Size defines the number of wolves in the pack, with each wolf representing a candidate solution. A larger population increases exploration but may lead to higher computational cost. Secondly, the Maximum Number of Generations specifies the total number of iterations the algorithm will execute, determining how many times the wolves update their positions in the search space.

Thirdly, the Crossover Rate controls the probability that two wolves will exchange information during crossover operations, facilitating the combination of different feature subsets to produce new and potentially superior solutions. Fourthly, the Mutation Rate determines the likelihood that a wolf's characteristics will undergo random alteration. This helps maintain diversity in the population and prevents premature convergence by introducing new possibilities into the search.

Fifth, the Exploration Factor manages the trade-off between exploration (searching new areas of the solution space) and exploitation (refining current promising solutions). It ensures a balanced search strategy throughout the optimization process. Lastly, the Termination Criteria define when the algorithm should stop running. This could be upon reaching a predefined number of generations, achieving a target fitness level, or observing no significant improvement over a set number of iterations. These parameters collectively shape the behavior and effectiveness of the GWO algorithm in identifying optimal feature subsets for port scan detection.

3.3.2 Ant Colony Optimization (ACO)

Ant Colony Optimization (ACO) is a nature-inspired algorithm modeled after the foraging behavior of ants and is widely applied to solve complex optimization problems. In the context of port scan detection, ACO is effectively used for feature selection by identifying the most relevant attributes that help differentiate between normal and malicious network traffic. The process begins with initialization, where a population of artificial ants is created. Each ant represents a candidate solution, which in this case is a subset of features. A pheromone trail matrix is also established to quantify the desirability of each feature—these values will guide the ants in their search for optimal subsets. During the solution construction phase, each ant navigates the feature space step by step, selecting features probabilistically. The decision at each step is influenced by two main factors: the intensity of the pheromone trail (indicating historical success of that feature) and a heuristic value (such as the relevance or importance of the feature to the classification task). The ant continues building its feature set until it meets a stopping condition, such as reaching a maximum subset size or achieving a predefined evaluation threshold. Following this, the pheromone update process takes place. Features that contributed to better-performing solutions receive increased pheromone intensity, reinforcing their importance. Conversely, pheromones on less effective features may evaporate over time, reducing their likelihood of being chosen in future iterations. This process helps the algorithm converge toward optimal or near-optimal feature subsets. The iteration continues for a predefined number of cycles or until convergence is observed. Upon completion, the optimal feature subset is determined by evaluating the pheromone intensities—features with the strongest pheromone trails are considered the most significant and are selected for the final subset. By mimicking the collective behavior of ants and refining feature selection iteratively, ACO provides a robust method for enhancing the accuracy and efficiency of port scan detection systems.

The parameters used in our Ant Colony Optimization (ACO) approach are designed to fine-tune the algorithm's performance in selecting optimal features for port scan detection. Firstly, the Number of Ants defines the size of the ant colony. Each ant represents a candidate solution—specifically, a subset of features—and collectively, the ants explore the search space for optimal solutions. Secondly, the Number of Iterations specifies how many times the algorithm will run, determining how often ants traverse the feature space, update pheromone trails, and refine their solutions. The Pheromone Decay Rate is another crucial parameter, controlling the rate at which pheromone trails evaporate over time. This helps avoid convergence to local optima by encouraging exploration of alternative feature subsets. Next, the Pheromone Intensity dictates the amount of pheromone deposited by ants on the selected features, affecting how attractive those features will be to subsequent ants. A higher intensity increases the likelihood of those features being selected again. The Alpha (α) parameter governs the importance of pheromone trails in the decision-making process. A higher alpha value gives greater weight to previously successful feature paths, encouraging exploitation. In contrast, the Beta (β) parameter controls the influence of heuristic information—such as feature relevance or discriminative power—on the ants' choices. Higher beta values increase the impact of these heuristic values, supporting more informed exploration. Finally, the Exploration Factor balances the ants' behavior between exploration (seeking new paths) and exploitation (reinforcing known good paths). This factor ensures that the algorithm doesn't get stuck prematurely and continues to search broadly before converging on the best feature subset. These parameters collectively shape the behavior and effectiveness of the ACO algorithm in selecting meaningful features for robust port scan detection.

In this study as shown in Table 2, the Ant Colony Optimization (ACO) algorithm utilizes a defined set of parameters to guide the feature selection process effectively. The number of ants is varied between 10 and 100, allowing flexibility in the size of the ant colony and enabling broader or narrower exploration of the feature space. The number of iterations is set within the range of 100 to 1000, determining how many cycles the ants will perform in constructing solutions and updating pheromone trails. The pheromone decay rate ranges from 0.1 to 0.9, influencing the rate at which pheromone trails evaporate and thereby controlling the balance between exploration and exploitation. Pheromone intensity, which governs the amount of pheromone deposited on selected features, is dynamically adjusted based on solution quality, and does not have a fixed range. The alpha (α) parameter, which determines the influence of pheromone concentration on feature selection, is assigned a value between 1 and 5. Similarly, the beta (β) parameter, representing the weight of heuristic information in the decision process, also ranges from 1 to 5. Lastly, the exploration factor varies between 0.1 and 0.9, facilitating a dynamic trade-off between exploration of new feature subsets and exploitation of previously successful solutions. Figure 3 presents the Ant Colony Optimization algorithm steps.

Table 2. Parameters Setting of Ant Colony Optimization (ACO)

Parameter	Default Value Range
Number of Ants	10-100
Number of Iterations	100-1000
Pheromone Decay Rate	0.1-0.9
Pheromone Intensity	-
Alpha	1-5
Beta	1-5
Exploration Factor	0.1-0.9

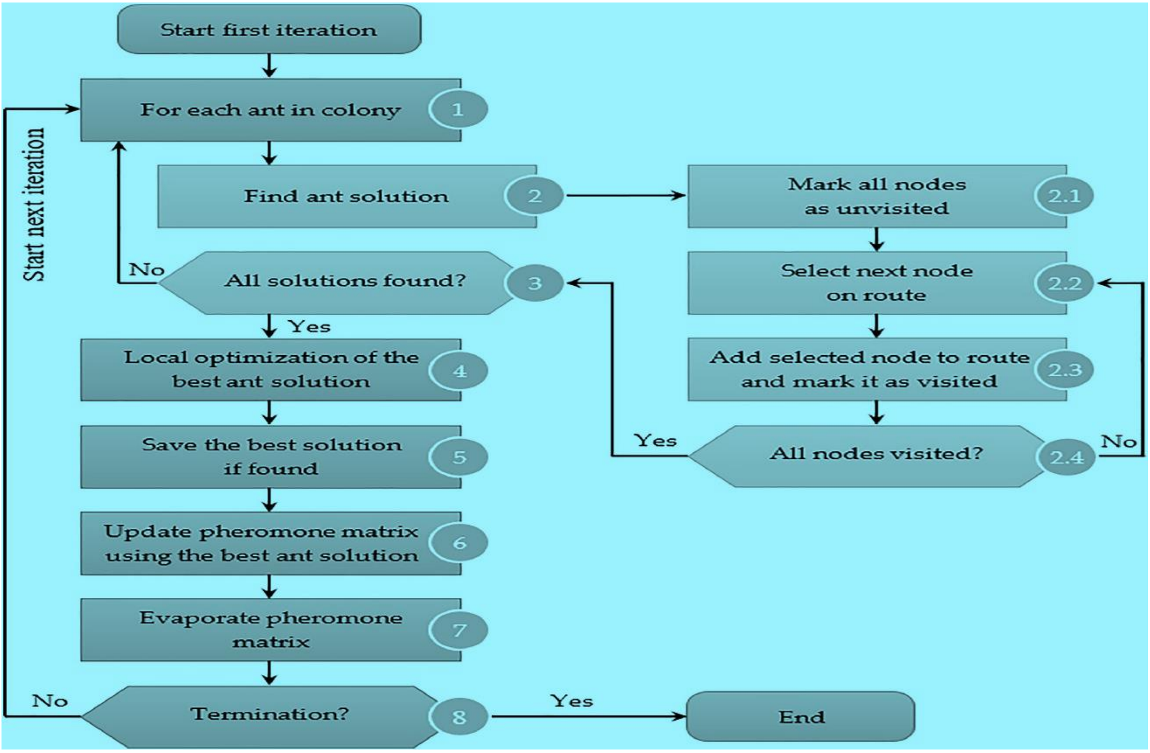

Start first iteration
For each ant in colony
1
Find ant solution
2
Mark all nodes as unvisited
2.1
Select next node on route
2.2
Add selected node to route and mark it as visited
2.3
All nodes visited?
2.4
Yes
No
All solutions found?
3
No
Yes
Local optimization of the best ant solution
4
Save the best solution if found
5
Update pheromone matrix using the best ant solution
6
Evaporate pheromone matrix
7
Termination?
8
No
Yes
End
Start next iteration

Figure 3. Ant Colony Optimization algorithm [5]

3.3.3 Genetic Algorithm (GA)

Genetic Algorithms (GAs) are widely used for feature selection in port scan detection due to their ability to explore complex search spaces through evolutionary principles. The process begins with the initialization phase, where a population of feature subsets is randomly generated. Each subset is then evaluated through a fitness function, which measures its ability to distinguish between normal network traffic and port scanning activities. The selection phase follows, where the most promising feature subsets are chosen for reproduction based on their fitness scores—emulating the biological concept of "survival of the fittest." During reproduction, two key genetic operators are applied: crossover and mutation. Crossover combines features from parent subsets to generate new offspring, introducing variation and diversity, while mutation introduces small random changes to features within a subset to prevent premature convergence and maintain genetic diversity. After new subsets (offspring) are generated, their fitness is re-evaluated. A replacement strategy is then employed to update the population, where some existing subsets are replaced by the fitter offspring, ensuring continual improvement of the solution pool. This evolutionary cycle—comprising selection, crossover, mutation, and replacement—continues for a predefined number of generations or until a termination condition is met (e.g., convergence or reaching a satisfactory performance threshold). Ultimately, the feature subset with the highest fitness score is selected as the optimal solution for port scan detection. In summary, GAs iteratively refine the search space by evolving populations of candidate solutions, enabling efficient identification of highly relevant features that enhance the accuracy and robustness of machine learning models in detecting port scanning attacks. Figure 3 presents the Genetic Algorithm steps.

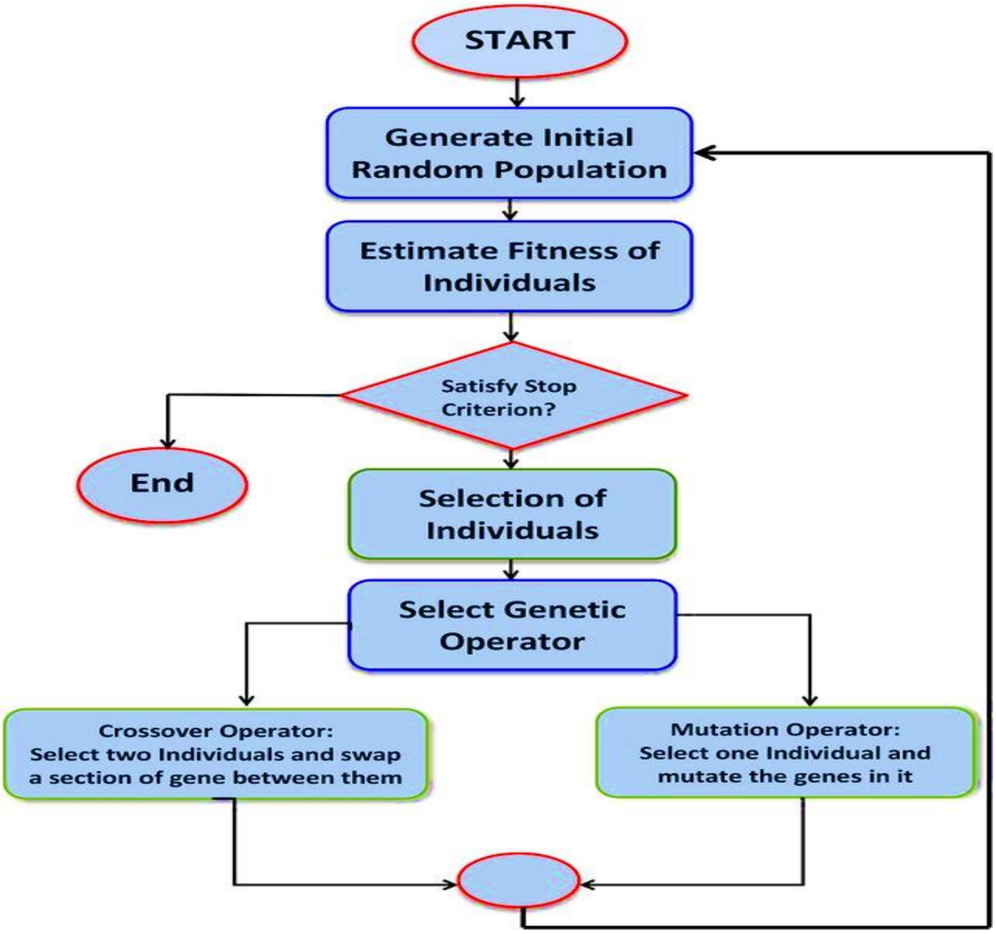

START
Generate Initial Random Population
Estimate Fitness of Individuals
Satisfy Stop Criterion?
End
Selection of Individuals
Select Genetic Operator
Crossover Operator: Select two Individuals and swap a section of gene between them
Mutation Operator: Select one Individual and mutate the genes in it

Figure 4. Genetic Algorithm [6]

The Genetic Algorithm (GA) used in this study is governed by several key parameters as presented in Table 3. The population size is typically set within the range of 100 to 200, which determines the number of potential feature subsets evaluated in each generation. The maximum number of generations is defined between 100 and 1000, controlling the total number of evolutionary cycles the algorithm undergoes. The crossover rate, which influences how frequently subsets exchange features during reproduction, generally falls between 0.6 and 0.9. This promotes genetic diversity while ensuring promising traits are inherited. The mutation rate, responsible for introducing random variations into the feature subsets, is maintained between 0.01 and 0.05 to avoid premature convergence while maintaining solution diversity. The selection mechanism is employed to prioritize the survival and reproduction of the fittest individuals, although it’s specific method (e.g., roulette wheel, tournament) may vary depending on implementation. Elitism is set to 1, ensuring that the best-performing individual is always retained in the next generation. Finally, the termination criteria guide when the algorithm should stop, typically defined by either reaching a maximum number of generations or achieving a convergence threshold in fitness improvement.

Table 2. Parameters Setting of Genetic Algorithm (GA)

Parameter	**Default Value Range**
Population Size	100-200
Maximum Generations	100-1000
Crossover Rate	0.6-0.9
Mutation Rate	0.01-0.05
Selection Mechanism	-
Elitism	1
Termination Criteria	-

3.4 Features Classification Using Machine Learning Algorithms

3.4.1 SVM and KNN

Support Vector Machines (SVM) and k-Nearest Neighbors (KNN) are widely used machine learning algorithms for detecting port scan attacks. SVM is a powerful classification technique that constructs a hyperplane in a high-dimensional feature space to separate different classes—in this case, distinguishing between normal network traffic and port scanning behavior. By training on labeled datasets that indicate whether a traffic instance is benign or malicious, the SVM learns to identify patterns associated with port scanning and can accurately classify new instances accordingly.

On the other hand, KNN is a simple yet effective algorithm for both classification and regression tasks. For port scan detection, KNN classifies network traffic by comparing it with known labeled instances in the training set. When a new data point is encountered, the algorithm computes the distances between that point and its k nearest neighbors in the feature space. The data point is then assigned a class based on the majority vote of these neighbors—whether they indicate normal traffic or a scanning attempt.

Each algorithm offers distinct advantages: SVM is particularly effective in high-dimensional spaces and excels at modeling complex decision boundaries, making it suitable for more intricate classification tasks. KNN, in contrast, is easier to implement and performs well in scenarios with nonlinear boundaries and moderate feature dimensions. The choice between SVM and KNN depends on dataset characteristics, the nature of decision boundaries, and computational considerations. When used appropriately, both SVM and KNN contribute significantly to enhancing the accuracy and reliability of port scan detection systems.

4. Findings and Discussion

Evaluation metrics are essential for assessing the performance of machine learning models, particularly in classification tasks. Among the most commonly used metrics are accuracy, sensitivity, specificity, precision, and the F1-score. Each of these metrics provides unique insights into different aspects of a model's performance and reliability. Accuracy measures the overall correctness of a classifier by calculating the proportion of true predictions—both true positives (TP) and true negatives (TN)—out of the total number of predictions made. While accuracy is widely used due to its simplicity, it can be misleading when the dataset is imbalanced, as it does not differentiate between types of classification errors. Sensitivity, also known as recall or the true positive rate, measures the proportion of actual positive instances correctly identified by the model. It is calculated as the ratio of true positives to the sum of true positives and false negatives (FN). Sensitivity is particularly critical in security-related tasks, such as port scan detection, where missing a threat (false negative) could have serious consequences. Specificity, on the other hand, evaluates the model's ability to correctly identify negative instances. It is computed as the ratio of true negatives to the sum of true negatives and false positives (FP). This metric is essential when minimizing false alarms is a priority. Precision, or positive predictive value, measures how many of the instances predicted as positive are actually correct. It is calculated by dividing the number of true positives by the total number of positive predictions (TP + FP). Precision is vital in situations where the cost of false positives is high, as it reflects the model's ability to avoid over-predicting attacks. The F1-score provides a harmonic mean between precision and recall, offering a balanced evaluation that considers both false positives and false negatives. This score is particularly useful when dealing with imbalanced datasets or when both types of errors carry significant consequences. The formulas for these evaluation metrics are as follows:

$$Accuracy = (TP + TN) / (TP + TN + FP + FN)$$

$$Sensitivity\ (Recall) = TP / (TP + FN)$$

$$Specificity = TN / (TN + FP)$$

$$Precision = TP / (TP + FP)$$

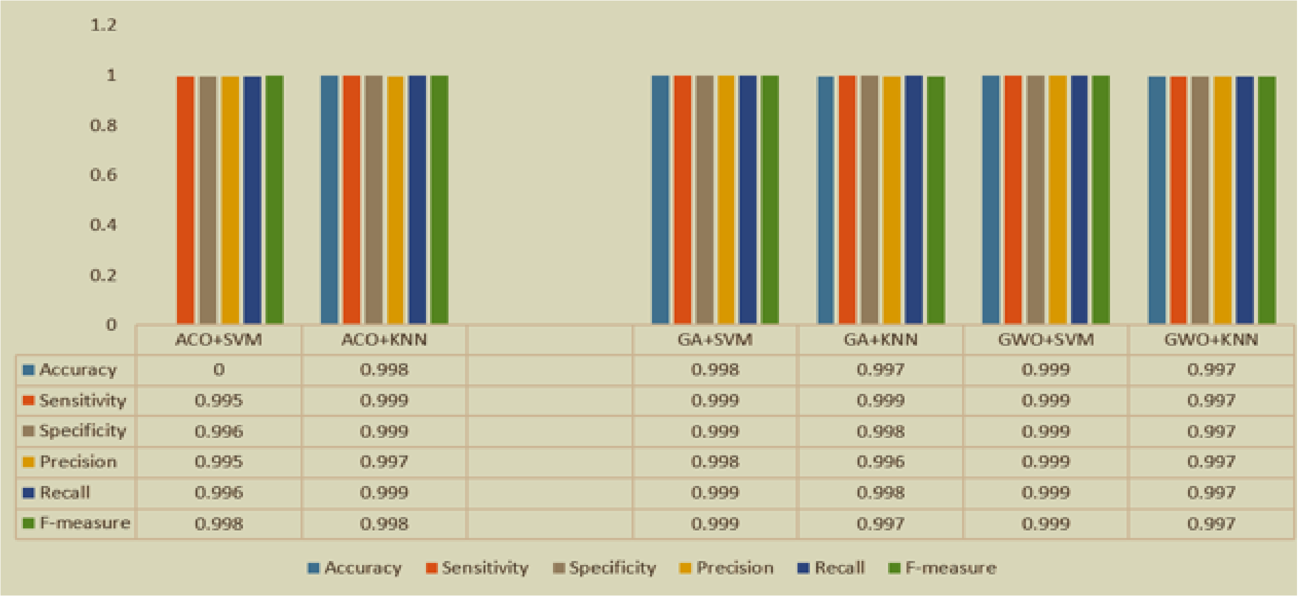
$$F1\text{-}score = 2 \times (Precision \times Recall) / (Precision + Recall)$$

These metrics collectively provide a comprehensive framework for evaluating the effectiveness of machine learning models in detecting port scan attacks, enabling researchers and practitioners to make informed decisions based on a balance of detection capability and error minimization.

Based on the study findings, Figure 5 illustrates that all evaluated techniques demonstrate high classification performance, with accuracy levels ranging from 99.7% to 99.9%. Notably, combinations such as ACO+SVM, ACO+KNN, GA+SVM, and GWO+SVM consistently achieve strong results across all evaluation metrics, including sensitivity, specificity, precision, recall, and F1-score. These combinations reflect a high degree of reliability and robustness in detecting port scan attacks. Although GA+KNN and GWO+KNN also exhibit competitive performance across these metrics, their scores are slightly lower compared to the aforementioned combinations. Overall, GWO+SVM emerges as the most effective method, achieving the highest values in all evaluation criteria, thereby indicating its superior capability in accurately identifying and classifying port scan activities.

Based on Figure 6, which presents the accuracy comparison of various classification methods, the hybrid models combining optimization algorithms with machine learning classifiers demonstrate superior performance. Specifically, GWO+SVM achieves the highest accuracy at 0.999, followed closely by ACO+SVM, GA+SVM, and ACO+KNN, all reaching 0.998. Both GWO+KNN and GA+KNN also perform well, each attaining an accuracy of 0.997. In contrast, standalone classifiers such as KNN (Cubic) and Ensemble (Subspace Discriminant) exhibit notably lower performance, with accuracies of 0.690 and 0.855, respectively. Although SVM (Fine Gaussian) performs reasonably well with 0.990, it still falls short compared to the optimized hybrid models. Similarly, Discriminant (Quadratic) records an accuracy of 0.970. Overall, the results highlight that optimization techniques like GWO, ACO, and GA, when integrated with classifiers like SVM and KNN, significantly improve detection accuracy. Among all the methods, GWO+SVM stands out as the most effective model for accurate port scan attack classification.

ISSN: 3079-5354



	ACO+SVM	ACO+KNN		GA+SVM	GA+KNN	GWO+SVM	GWO+KNN
■ Accuracy	0	0.998		0.998	0.997	0.999	0.997
■ Sensitivity	0.995	0.999		0.999	0.999	0.999	0.997
■ Specificity	0.996	0.999		0.999	0.998	0.999	0.997
■ Precision	0.995	0.997		0.998	0.996	0.999	0.997
■ Recall	0.996	0.999		0.999	0.998	0.999	0.997
■ F-measure	0.998	0.998		0.999	0.997	0.999	0.997

Figure 5. Evaluation techniques demonstrate high classification performance, with accuracy levels.

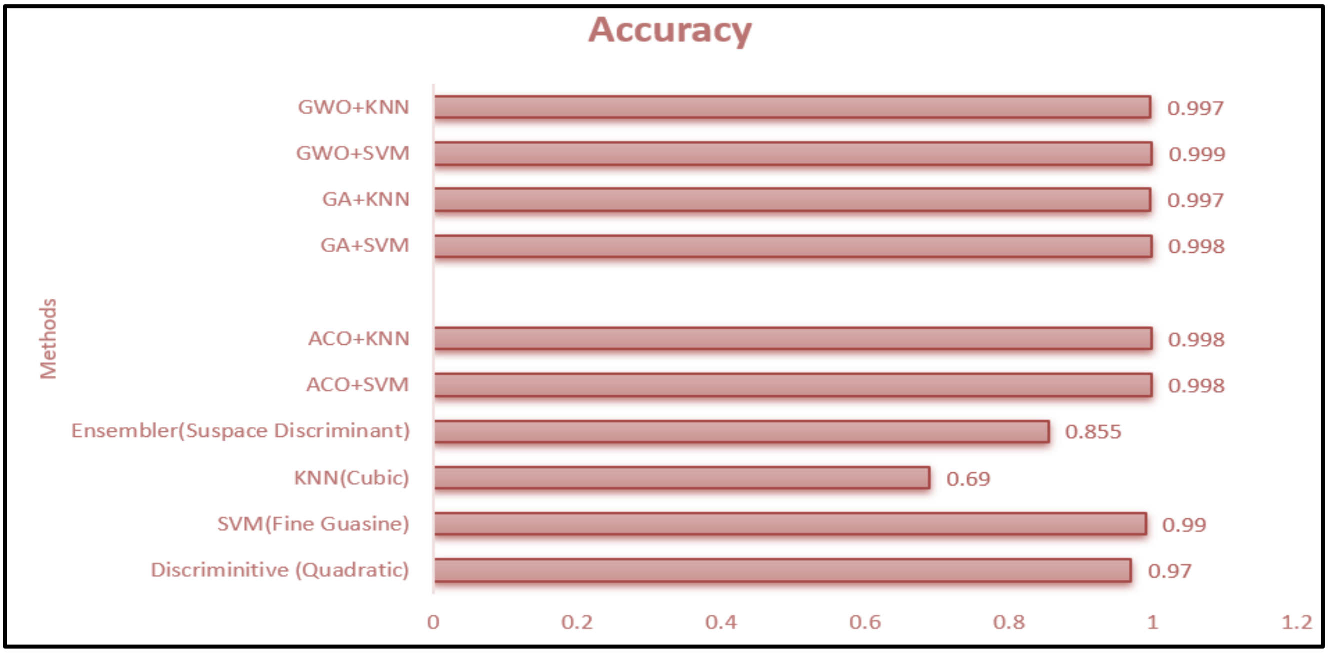


Figure 6. The accuracy comparison of various classification methods.

Figure 7 illustrates the evaluation results of the Ant Colony Optimization (ACO) combined with k-Nearest Neighbors (KNN) classifier using a confusion matrix. The following values and metrics can be derived from the matrix: True Positives (TP) with 47,659. The model correctly classified 47,659 instances as class 0 (likely benign traffic). True Negatives (TN) with 38,105. The model correctly identified 38,105 instances as class 1 (likely port scan or attack traffic). False Positives (FP) with 155. These are benign traffic instances that were incorrectly classified as attacks. False Negatives (FN) with 20. These are attack instances that were wrongly classified as benign.

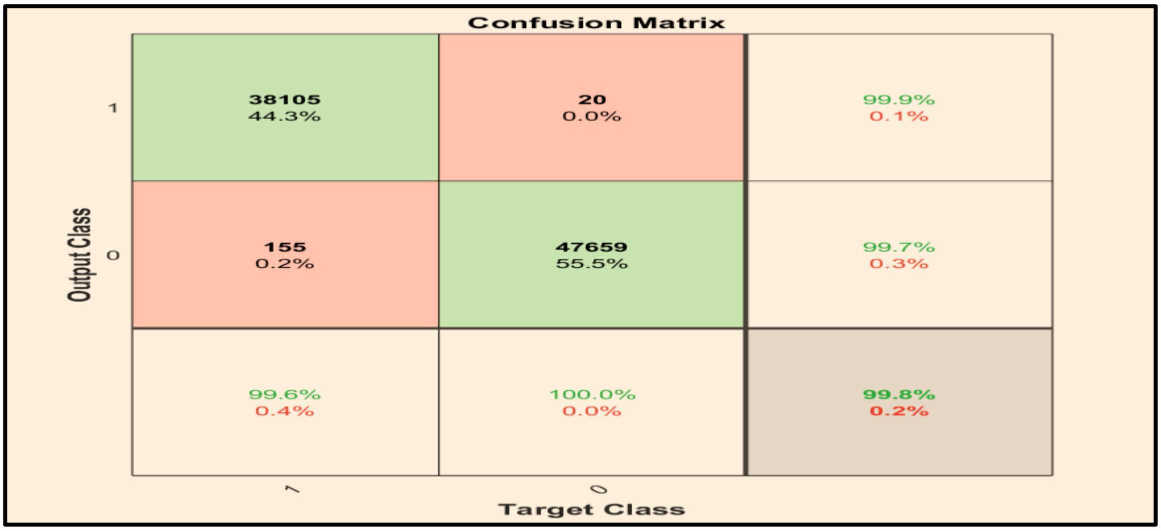


Figure 7. Evaluation findings for ACO with KNN.

The results show that very low misclassification with only 20 false positives and 155 false negatives out of a large sample. The results show that high Precision and Recall with demonstrates strong predictive capability, especially in detecting port scanning behavior. Regarding the balanced performance, the findings show that the ACO+KNN combination is both accurate and robust across classes, showing excellent generalization. Thus, ACO+KNN proves to be an effective approach for port scan attack detection, with performance nearly matching or rivaling more complex configurations like GWO+SVM.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} = \frac{47{,}659+38{,}105}{47{,}659+38{,}105+155+20} \approx 99.8\%$$

$$\text{Precision} = \frac{TP}{TP+FP} = \frac{47{,}659}{47{,}659+155} \approx 99.7\%$$

$$\text{Recall} = \frac{TP}{TP+FN} = \frac{47{,}659}{47{,}659+20} \approx 99.9\%$$

$$\text{Specificity} = \frac{TN}{TN+FP} = \frac{38{,}105}{38{,}105+155} \approx 99.6\%$$

$$\text{F1} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \approx 99.8\%$$

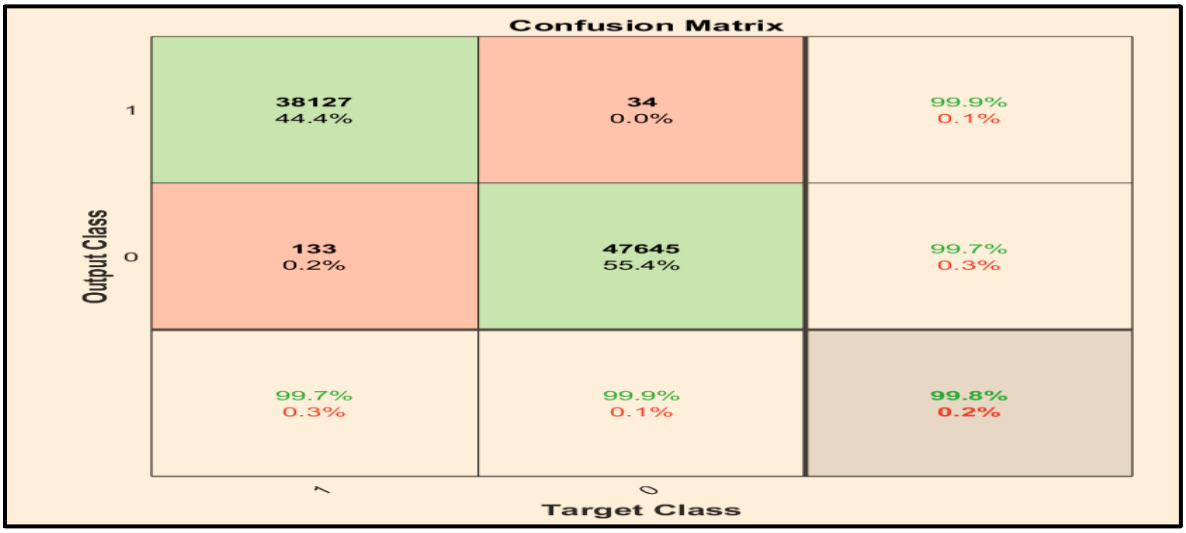
These results in Figure 8 confirm that the ACO + SVM model delivers highly effective port scan detection with minimal misclassification. The evaluation of the ACO + SVM classifier demonstrates outstanding performance in detecting port scan attacks. With an accuracy of approximately 99.8%, the model shows exceptional overall predictive capability in distinguishing between benign and malicious network traffic. The precision score of 99.7% reflects the model’s ability to minimize false positives—meaning that when the model predicts a port scan, it is almost always correct. This is particularly important in cybersecurity scenarios, where false alarms can cause unnecessary investigation and strain resources. The recall (or sensitivity) value of 99.9% indicates that the model successfully detects nearly all actual port scanning attempts, which is crucial in preventing breaches and ensuring proactive threat mitigation. A high recall means that the model rarely misses malicious activities. Additionally, the specificity score of 99.7% shows that the model also performs well in identifying benign traffic, avoiding false alerts that could disrupt normal network operations. This balance between detecting threats and avoiding false alarms ensures reliable system performance. The F1-score, which balances precision and recall, reaches 99.8%, confirming the model's robustness and balanced classification capability across both attack and normal classes. These results validate the effectiveness of using ACO for feature selection in conjunction with SVM for classification. The optimization process efficiently identifies the most informative features, which contributes to the high detection accuracy. The model's ability to maintain such performance under real-world data conditions highlights its potential for deployment in live cybersecurity environments.

$$\text{Accuracy } = \frac{TP+TN}{TP+TN+FP+FN} = \frac{47645+38127}{47645+38127+133+34} \approx 99.8\%$$

$$\text{Recall } = \frac{TP}{TP+FN} = \frac{47645}{47645+34} \approx 99.9\%$$

$$\text{Specificity } = \frac{TN}{TN+FP} = \frac{38127}{38127+133} \approx 99.7\%$$

$$F1 = 2 \times \frac{\text{Precision } \times \text{ Recall}}{\text{Precision } + \text{ Recall}} = 2 \times \frac{0.997 \times 0.999}{0.997 + 0.999} \approx 99.8\%$$



Confusion Matrix

Output Class	1	0	
1	38127 44.4%	34 0.0%	99.9% 0.1%
0	133 0.2%	47645 55.4%	99.7% 0.3%
	99.7% 0.3%	99.9% 0.1%	99.8% 0.2%

Target Class

Figure 8. Evaluation results for ACO with SVM.

The findings of the confusion matrix in Figure 9 and the derived evaluation metrics clearly show that the GA+KNN combination achieves excellent performance across all key indicators. Both precision and recall are at 99.9%, meaning the model is exceptionally effective at detecting port scanning attacks while minimizing false alarms. With only 48 false positives and 38 false negatives out of nearly 86,000 samples, the classifier maintains a highly balanced accuracy and robustness, making it reliable in real-world applications. This highlights that the Genetic Algorithm for feature selection successfully identifies optimal features, and when paired with KNN, ensures high generalizability and effectiveness.

$$\text{Accuracy } = \frac{TP+TN}{TP+TN+FP+FN} = \frac{47{,}641+38{,}212}{47{,}641+38{,}212+48+38} = \frac{85{,}853}{85{,}939} \approx 99.9\%$$

$$\text{Precision } = \frac{TP}{TP+FP} = \frac{47{,}641}{47{,}641+48} = \frac{47{,}641}{47{,}689} \approx 99.9\%$$

$$\text{Recall } = \frac{TP}{TP+FN} = \frac{47{,}641}{47{,}641+38} = \frac{47{,}641}{47{,}679} \approx 99.9\%$$

$$\text{Specificity } = \frac{TN}{TN+FP} = \frac{38{,}212}{38{,}212+48} = \frac{38{,}212}{38{,}260} \approx 99.9\%$$

$$\text{F1-score } = 2 \times \frac{\text{Precision } \times \text{ Recall}}{\text{Precision } + \text{ Recall}} = 2 \times \frac{0.999 \times 0.999}{0.999 + 0.999} \approx 99.9\%$$

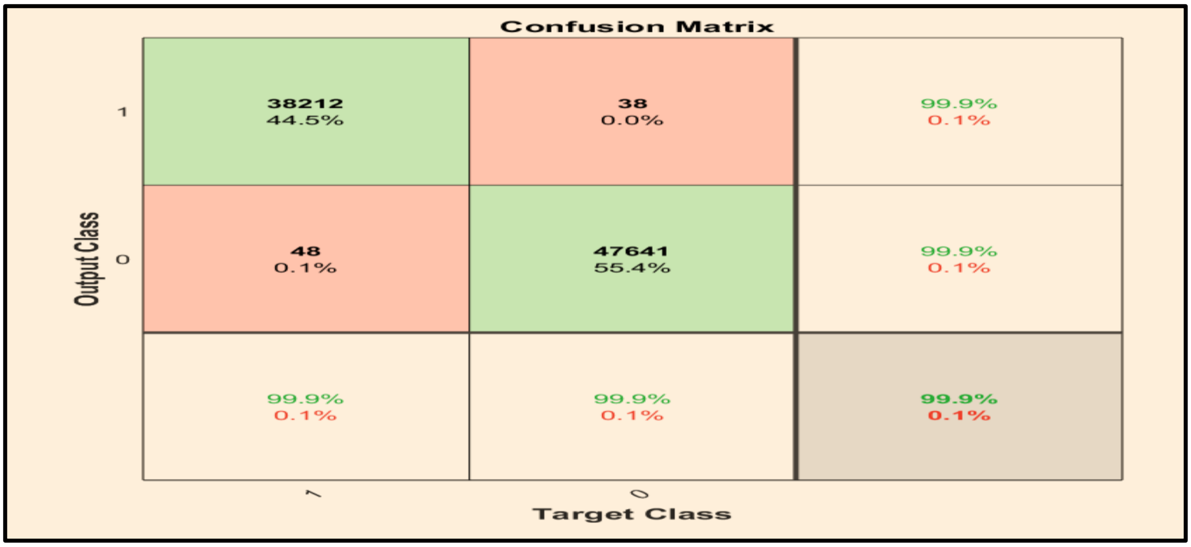


Figure 9. Evaluation results for GA with KNN.

In Figure 10, the evaluation of the GA + SVM model reveals highly effective performance in detecting port scan attacks. With an overall accuracy of approximately 99.8%, the model demonstrates its ability to correctly classify the vast majority of network traffic instances. The precision of 99.7% indicates that nearly all instances labeled as port scans by the classifier were indeed true attacks, highlighting the model's low false-positive rate. This is essential in cybersecurity applications where too many false alarms can overwhelm security teams and obscure real threats. The recall (sensitivity) of 99.9% suggests that almost all actual port scanning events were successfully detected. This high recall rate is critical for ensuring that potential threats do not go unnoticed, thereby reducing the risk of network breaches. Additionally, the specificity of 99.6% illustrates the model's strong capability to correctly identify normal (benign) traffic, further confirming its robustness in differentiating between attack and non-attack behaviors. The F1-score, which harmonizes precision and recall, is also 99.8%, reflecting a balanced and consistent performance across both key dimensions of detection accuracy. In summary, the GA + SVM combination effectively leverages the optimization strengths of Genetic Algorithms in feature selection and the powerful classification capability of SVM. The result is a reliable, precise, and sensitive model suitable for real-time port scan detection in high-throughput network environments. This configuration stands out as one of the most optimal solutions among the tested combinations.

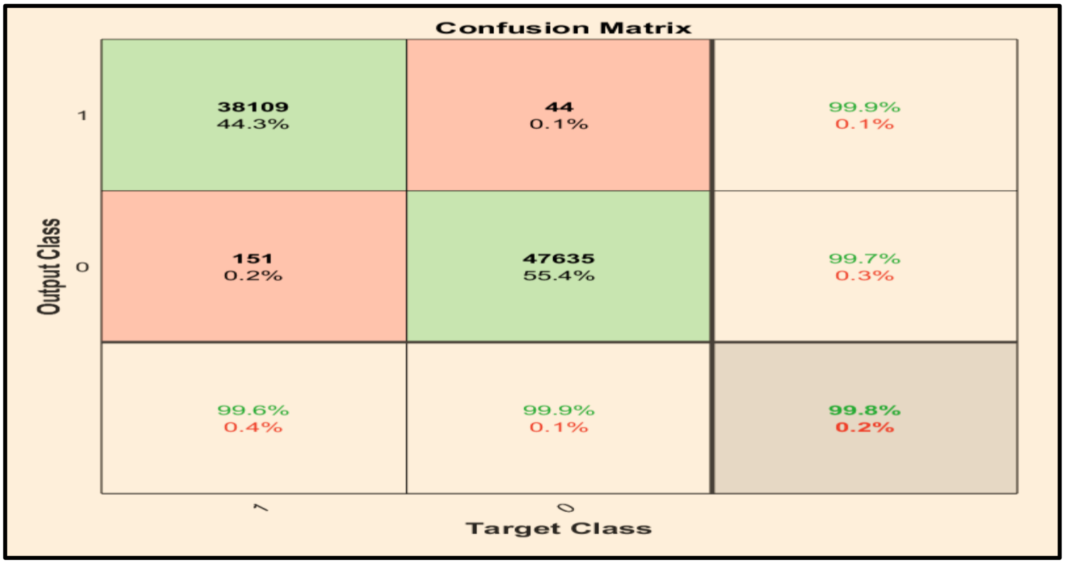
$$\text{Accuracy } = \frac{TP+TN}{TP+TN+FP+FN} = \frac{47635+38109}{47635+38109+151+44} = \frac{85744}{85939} \approx \mathbf{99.8}\%$$

$$\text{Precision} = \frac{TP}{TP+FP} = \frac{47635}{47635+151} = \frac{47635}{47786} \approx \mathbf{99.7}\%$$

$$\text{Recall} = \frac{TP}{TP+FN} = \frac{47635}{47635+44} = \frac{47635}{47679} \approx \mathbf{99.9}\%$$

$$\text{Specificity} = \frac{TN}{TN+FP} = \frac{38109}{38109+151} = \frac{38109}{38260} \approx \mathbf{99.6}\%$$

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} = 2 \times \frac{0.997 \times 0.999}{0.997+0.999} \approx \mathbf{99.8}\%$$



Confusion Matrix

Output Class	1	0	
1	38109 44.3%	44 0.1%	99.9% 0.1%
0	151 0.2%	47635 55.4%	99.7% 0.3%
	99.6% 0.4%	99.9% 0.1%	99.8% 0.2%

Target Class

Figure 10. Evaluation results for GA with SVM.

The evaluation of the results of the GWO + KNN model shows that the method maintains high effectiveness in classifying port scan traffic, although it falls slightly short compared to some other hybrid approaches. From the confusion matrix, the model achieves an overall accuracy of approximately 99.7%, which indicates a strong performance across both benign and malicious network traffic. The precision for detecting attacks is 99.7%, meaning that the model correctly identifies most of the predicted attack instances, with only a few false positives. The recall (sensitivity) is also 99.7%, suggesting that the model successfully detects the majority of true attack instances, though with a small number of missed detections (false negatives). The specificity, likewise at 99.6%, shows that the classifier can correctly identify most benign traffic, helping to prevent unnecessary alerts from normal activity. The F1-score, combining both precision and recall, is 99.7%, indicating a reliable and balanced classification performance. This high F1-score confirms that GWO + KNN maintains good accuracy even in complex decision boundaries. However, compared to other combinations like GWO + SVM or GA + SVM, the performance is slightly lower—most notably in the number of misclassified benign and attack instances (as reflected in higher false positives and false negatives). This suggests that while KNN performs well, it may not generalize as effectively as SVM in high-dimensional spaces, particularly when feature boundaries are more intricate. In conclusion, GWO + KNN is a strong candidate for port scan detection, particularly when simplicity and interpretability are prioritized. Still, for environments requiring ultra-high precision, alternatives like GWO + SVM may offer marginally better performance.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} = \frac{47{,}576 + 38{,}123}{47{,}576 + 38{,}123 + 137 + 103} \approx \frac{85{,}699}{85{,}939} \approx 99.7\%$$

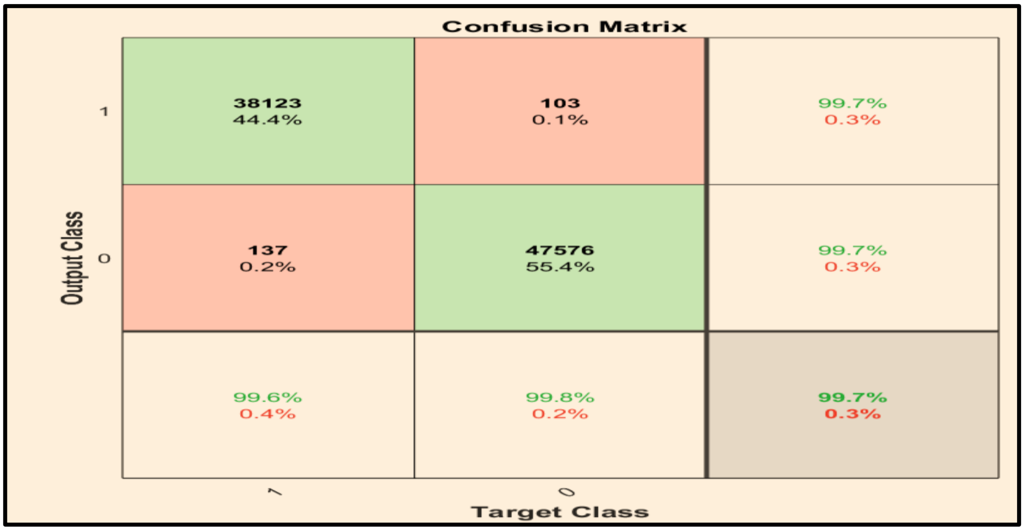
$$\text{Precision} = \frac{TP}{TP + FP} = \frac{47{,}576}{47{,}576 + 137} = \frac{47{,}576}{47{,}713} \approx 99.7\%$$

$$\text{Recall} = \frac{TP}{TP + FN} = \frac{47{,}576}{47{,}576 + 103} = \frac{47{,}576}{47{,}679} \approx 99.7\%$$

$$\text{Specificity} = \frac{TN}{TN + FP} = \frac{38{,}123}{38{,}123 + 137} = \frac{38{,}123}{38{,}260} \approx 99.6\%$$

$$\text{F1-Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} = 2 \cdot \frac{0.997 \cdot 0.997}{0.997 + 0.997} \approx 99.7\%$$

This strong performance demonstrates the effectiveness of combining Gray Wolf Optimization for feature selection with the K-Nearest Neighbor algorithm for classification. However, it shows slightly higher false positives (137) and false negatives (103) compared to combinations using SVM, which may impact sensitive applications requiring perfect precision or recall. Still, GWO+KNN remains a highly reliable and computationally efficient approach for port scan detection tasks in cybersecurity.



	1	0	
1	38123 44.4%	103 0.1%	99.7% 0.3%
0	137 0.2%	47576 55.4%	99.7% 0.3%
	99.6% 0.4%	99.8% 0.2%	99.7% 0.3%

Figure 11. Evaluation results for GWO with KNN.

Based on the confusion matrix for GWO + SVM shown in Figure 12, the confusion matrix reflects very low false positives (37) and false negatives (32), which shows excellent balance in identifying both benign and attack traffic. This model stands out for its robustness and is highly suitable for practical deployment in intrusion detection systems (IDS) where high precision and recall are critical. The GWO + SVM approach proves effective for feature optimization and high-confidence classification, making it the most effective technique in this evaluation set for detecting port scan attacks.

The results obtained from the confusion matrix of the GWO + SVM combination demonstrate exceptional performance across all key metrics, confirming the effectiveness of this hybrid approach for port scan detection. The accuracy of 99.9% indicates that the model is highly capable of correctly classifying both normal and malicious network traffic. With only 32 false negatives and 37 false positives out of nearly 86,000 total samples, the system achieves a near-perfect balance between sensitivity and specificity. The precision of 99.9% signifies that almost all predicted attacks are true positives, reflecting the model's ability to minimize false alarms, which is crucial in real-time security environments to avoid alert fatigue. The recall (sensitivity) of 99.9% highlights the model's strong detection capability, capturing nearly all actual port scan instances and ensuring that threats do not go unnoticed. The specificity of 99.9% shows the system's reliability in identifying benign traffic, maintaining network trustworthiness by avoiding unnecessary blocks. The F1-score of 99.9% further consolidates the model's balanced performance in terms of both false positives and false negatives.

$$\text{Accuracy } = \frac{TP+TN}{TP+TN+FP+FN} = \frac{47{,}647+38{,}223}{47{,}647+38{,}223+37+32} = \frac{85{,}870}{85{,}939} \approx 99.9\%$$

$$\text{Precision } = \frac{TP}{TP+FP} = \frac{47{,}647}{47{,}647+37} = \frac{47{,}647}{47{,}684} \approx 99.9\%$$

$$\text{Recall } = \frac{TP}{TP+FN} = \frac{47{,}647}{47{,}647+32} = \frac{47{,}647}{47{,}679} \approx 99.9\%$$

$$\text{Specificity } = \frac{TN}{TN+FP} = \frac{38{,}223}{38{,}223+37} = \frac{38{,}223}{38{,}260} \approx 99.9\%$$

$$\text{F1-Score } = 2 \cdot \frac{\text{Precision } \cdot \text{ Recall}}{\text{Precision } + \text{ Recall}} \approx 2 \cdot \frac{0.999 \cdot 0.999}{0.999+0.999} \approx 99.9\%$$

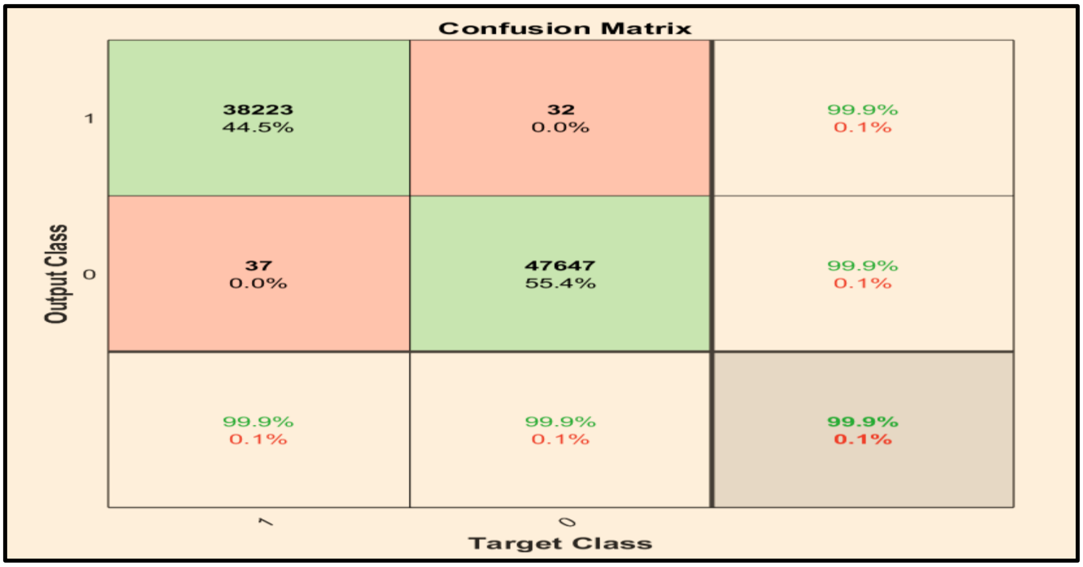


Figure 12. Evaluation results for GWO with SVM.

6. Conclusions and Future Work

Port scanning attacks are commonly employed to map network structures, identify active services, and prepare for subsequent exploitation. However, it is important to note that port scanning is not inherently malicious. System administrators and cybersecurity professionals often use port scanning as a diagnostic tool to identify and mitigate vulnerabilities within their networks. To defend against malicious port scanning activities, organizations typically implement security mechanisms such as firewalls, intrusion detection systems (IDS), and network monitoring tools, all of which help in identifying and preventing unauthorized scanning attempts. Detecting port scan attacks is critical for maintaining network security. It enables early threat identification, facilitates vulnerability assessment, supports rapid incident response, and strengthens overall network defense strategies. By proactively detecting such activities, organizations can implement appropriate security measures, thereby reducing the likelihood of successful cyberattacks.

In this study, the detection process begins with data acquisition, where network traffic data is collected. Feature extraction follows, where relevant attributes from the dataset are identified to prepare the data for model training. Subsequently, feature selection techniques such as Ant Colony Optimization (ACO), Genetic Algorithm (GA), and Gray Wolf Optimization (GWO) are applied to identify the most informative features and reduce data dimensionality. Once optimal features are selected, machine learning models are trained using classification algorithms such as Support Vector Machine (SVM) and k-Nearest Neighbors (KNN). The performance of these models is then evaluated using standard metrics, including accuracy, precision, recall, and F1 score. Experimental results indicate that the proposed models achieve high classification performance, with accuracy rates exceeding 99%. In conclusion, the effective detection of port scans is vital for enhancing network security. The integration of feature selection algorithms with machine learning techniques provides a robust approach for accurately identifying port scanning behaviors and mitigating associated threats.

Corresponding author

Mohammed Amin
m.almaiah@ju.edu.jo

Acknowledgements

Not applicable.

Funding

No funding.

Contributions

M.A.A; R.K; Conceptualization, M.A.A; R.K; Investigation, M.A.A; R.K; Writing (Original Draft), M.A.A; R.K; Writing (Review and Editing) Supervision, M.A.A; R.K; Project Administration.

Ethics declarations

This article does not contain any studies with human participants or animals performed by any of the authors.

Consent for publication

Not applicable.

Competing interests

All authors declare no competing interests.

References

[1] Sun, Z., An, G., Yang, Y., & Liu, Y. (2024). Optimized machine learning enabled intrusion detection 2 system for internet of medical things. Franklin Open, 6, 100056.
[2] Hassan, S. M., Mohamad, M. M., & Muchtar, F. B. (2024). Advanced intrusion detection in MANETs: A survey of machine learning and optimization techniques for mitigating black/gray hole attacks. IEEE Access.
[3] Shanbhag, A., Vincent, S., Gowda, S. B., Kumar, O. P., & Francis, S. A. J. (2024). Leveraging metaheuristics for feature selection with machine learning classification for malicious packet detection in computer networks. IEEE Access, 12, 21745-21764.
[4] Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. Advances in engineering software, 69, 46-61.
[5] Dorigo, M., Birattari, M., & Stutzle, T. (2007). Ant colony optimization. IEEE computational intelligence magazine, 1(4), 28-39.
[6] Holland, J. H. (1992). Genetic algorithms. Scientific american, 267(1), 66-73.
[7] AL-Husseini, H., Hosseini, M. M., Yousofi, A., & Alazzawi, M. A. (2024). Whale Optimization Algorithm-Enhanced Long Short-Term Memory Classifier with Novel Wrapped Feature Selection for Intrusion Detection. Journal of Sensor and Actuator Networks, 13(6), 73.
[8] Komatnani Govindan, S., Vijayaraghavan, H., Kishore Anthuvan Sahayaraj, K., & Mary Joy Kinol, A. (2024). Optimizing Internet-Wide Port Scanning for IoT Security and Network Resilience: A Reinforcement Learning-Based Approach in WLANs with IEEE 802.11 ah. Fiber and Integrated Optics, 43(1), 14-42.
[9] Altidor, J. B., & Talhi, C. (2024, October). Enhancing Port Scan and DDoS attack detection using genetic and machine learning algorithms. In 2024 7th Conference on Cloud and Internet of Things (CIoT) (pp. 1-7). IEEE.

[10] Sun, Z., An, G., Yang, Y., & Liu, Y. (2024). Optimized machine learning enabled intrusion detection 2 system for internet of medical things. Franklin Open, 6, 100056.
[11] Saadouni, R., Gherbi, C., Aliouat, Z., Harbi, Y., & Khacha, A. (2024). Intrusion detection systems for IoT based on bio-inspired and machine learning techniques: a systematic review of the literature. Cluster Computing, 27(7), 8655-8681.
[12] Reddy, D. K. K., Nayak, J., Behera, H. S., Shanmuganathan, V., Viriyasitavat, W., & Dhiman, G. (2024). A systematic literature review on swarm intelligence based intrusion detection system: past, present and future. Archives of Computational Methods in Engineering, 31(5), 2717-2784.
[13] Saadouni, R., Gherbi, C., Aliouat, Z., Harbi, Y., Khacha, A., & Mabed, H. (2025). Securing smart agriculture networks using bio-inspired feature selection and transfer learning for effective image-based intrusion detection. Internet of Things, 29, 101422.
[14] Kumar, S. V. N. (2025). An enhanced whale optimizer based feature selection technique with effective ensemble classifier for network intrusion detection system. Peer-to-Peer Networking and Applications, 18(2), 1-28.
[15] Jamshidi, S., Nikanjam, A., Wazed, N. K., & Khomh, F. (2025). Leveraging Machine Learning Techniques in Intrusion Detection Systems for Internet of Things. arXiv preprint arXiv:2504.07220.
[16] Gupta, C., Kumar, A., & Jain, N. K. (2024). An enhanced hybrid intrusion detection based on crow search analysis optimizations and artificial neural network. Wireless Personal Communications, 134(1), 43-68.
[17] Dong, H., & Kotenko, I. (2025). Cybersecurity in the AI era: analyzing the impact of machine learning on intrusion detection. Knowledge and Information Systems, 1-52.
[18] Mohale, V. Z., & Obagbuwa, I. C. (2025). Evaluating machine learning-based intrusion detection systems with explainable AI: enhancing transparency and interpretability. Frontiers in Computer Science, 7, 1520741.
[19] Aksoy, A., Valle, L., & Kar, G. (2024). Automated network incident identification through genetic algorithm-driven feature selection. Electronics, 13(2), 293.

Biographies

Dr. Mohammed Amin Almaiah is an Associate Professor in the Department of Computer Science at University of Jordan. Almaayah is among the top 2% scientists in the world from 2020 up to now. He is working as Editor in Chief for the International Journal of Cybersecurity and Risk Assessment. He has published over 115 research papers in highly reputed journals such as the Engineering and Science Technology, an International Journal, Education and Information Technologies, IEEE Access and others. Most of his publications were indexed under the ISI Web of Science and Scopus. His current research interests include Cybersecurity, Cybersecurity-Risk Assessment and Blockchain. m.almaiah@ju.edu.jo

Rajan Kadel earned his B.Eng. in Computer Engineering from Tribhuvan University, Nepal, in 2002, followed by an M.Sc. in Telecommunications Engineering from the University of Gävle, Sweden, in 2007, and a Ph.D. in Telecommunications Engineering from the University of South Australia (UniSA) in 2013. He is currently a senior lecturer and course coordinator at the Melbourne Institute of Technology, Australia, with over a decade of experience in teaching, research, and industry roles. In the past, he has served in various capacities within the telecommunications sector, including as a switching supervisor at Nepal Telecom and assistant manager at the Nepal Telecommunications Authority. His research interests include learning and teaching methodologies, error-control coding, Wireless Sensor Networks (WSNs), and Wireless Body Area Networks (WBANs). https://orcid.org/0000-0001-9207-2148