



Bio-Inspired Cyber Defense: Enhancing Intrusion Detection through the Waterwheel Plant Optimization Algorithm

Omar Bani Fayyad ^{1,*}

¹ School of ICT, Bahrain Polytechnic, PO Box 33349, Isa Town, Bahrain

Email: omar.fayyad@polytechnic.bh

Received: May 16, 2026 Revised: June 25, 2026 Accepted: August 27, 2026 * Corresponding author

ABSTRACT

In today's increasingly interconnected digital landscape, intrusion detection systems (IDS) are critical for ensuring the security and integrity of network infrastructures. However, the effectiveness of traditional IDS models can be hampered by challenges such as high-dimensional data and the complex dynamics of attack patterns. This paper addresses these challenges by leveraging the Waterwheel Plant Optimization Algorithm (WWPA), a bio-inspired metaheuristic, to optimize the hyperparameters of the Support Vector Machine (SVM), a widely used classifier for intrusion detection. Our contributions include the development of an optimized machine learning framework that significantly improves classification performance by integrating WWPA with SVM. Experimental results show that the baseline SVM achieves an accuracy of 86.36%, while the WWPA-optimized SVM yields an impressive accuracy of 97.00%. These results highlight the efficacy of WWPA in navigating the complex hyperparameter search space and enhancing the sensitivity and precision of intrusion detection models. The implications of this work are profound for cybersecurity, as it provides a robust, efficient, and adaptive solution for detecting intrusions with high accuracy, thereby improving the reliability and responsiveness of modern IDS systems.

Keywords: Intrusion Detection Systems ▪ Waterwheel Plant Algorithm (WWPA) ▪ Support Vector Machine (SVM) Optimization ▪ Metaheuristic Optimization ▪ Bio-Inspired Algorithms

1. INTRODUCTION

Today, the scope of penetration is growing, but this is happening because of the interconnected systems, clouds, and platforms that provide lots of data and are the target for the confidentiality, integrity, and availability of information.[1] Technological development elevated cybersecurity and network defense systems to become vital necessities for all modern technological frameworks.[2] Powered by the enormous and intricate network surface area and the energetic murkiness of cyber dangers, programmed and insightful frameworks that can spot, respond to, and, if possible, stop malicious or bots from getting underway naturally are essential[3].

Intrusion detection systems (IDS) are fundamental compo-

nents that cannot be disregarded in this situation.[3] These are sentinels of digital infrastructures that aim to detect involuntary access or misuse on networks and anomalies of behavior. Proactive and effective countermeasures can be applied, thus allowing risk mitigation and the system's functionality with trust. Especially, any detection in the early stage of the intrusion limits the attack's spread and damage, system downtime, and prevents data exfiltration or corruption.[4] Cyber attacks are becoming more complex and more frequent, and more sophisticated, so the traditional rule-based detection system has no idea how to come up with a solution to very novel, obfuscated attack strategies anymore[5]. There's a need for something adaptive, intelligent.

Machine learning (ML) established itself as a disruptive

technology that enables the identification of patterns and experience-based generalizations for prediction tasks across high-dimensional time-sensitive datasets. By probing non-linear patterns in network data, ML-based IDS models successfully detect patterns of suspicious activity[6]. Among these, the support vector machine (SVM) has received special attention since it can work well in a high-dimensional space, is flexible through kernel functions, and can make good decisions for a binary classification task. In intrusion detection problems, SVM performs exceptionally well because its algorithm optimizes the class separation margin between normal and attack behavior[6, 7].

Implementing machine learning approaches to network intrusion detection faces multiple long-term obstacles. Real-world security datasets present two main issues: imbalance and noise[8]. Today's NSL-KDD benchmark dataset significantly enhances research while addressing class imbalance issues and instance difficulty variations, but these problems diminish model robustness. Minority class intrusion instances that signify critical incidents often receive insufficient representation, triggering distorted learning results and elevated false negative outcomes[9].

Network traffic data contains numerous dimensions that can create redundant features and correlations among them[10]. Unmitigated semantic overlap between packet features alongside irrelevant characteristics extracted from raw data causes degradation in both classification performance and generalization capabilities of classifiers. The accuracy of SVM models requires precise hyperparameter adjustment and an understanding of the data structure because SVM responds unusually to changes in both aspects[11].

Developing optimal hyperparameters for SVM models continues to pose significant deployment difficulties. Optimizing the SVM performance requires finding the ideal values for the penalty term C and the kernel coefficient γ to strike a balance between underfitting and overfitting[10, 11]. The functionality of SVM models will suffer during real-time applications when their configurations are wrong, since they will inadequately identify attack traffic from regular user interactions. The adjustment process produces computational complexity that needs both strategic parameter exploration of the space and increased deployment complexity[12, 13].

Detection of network intrusions faces critical challenges between high sensitivity and specificity levels [14]. Network systems require classifiers demonstrating high sensitivity detection capabilities to identify most attacking events[15]. When sensitivity levels grow higher, it becomes harder for the system to maintain accurate identification of ordinary non-deliberate network traffic. Excessive false alarms from security systems reduce analysts' trust and slow their ability to take appropriate action. The successful implementation of IDS systems requires obtaining an optimal convergence of sensitivity measurements with specificity parameters[16, 17]. This study pursues two main objectives in response to the recognized difficulties. This work strictly evaluates how common classification models function on the NSL-KDD database performance when used by experts[7, 14, 18]. Five main classification models were utilized in this research including support vector machine (SVM), random forest (RF) and k -nearest neighbors (KNN), naive Bayes (NB) and de-

cision tree (DT). The picked set of models consists of established algorithms from different supervised learning domains and models that have appeared frequently in intrusion detection research [19, 20, 21].

This research focuses on optimizing SVM classification by applying different metaheuristic algorithms. Research gives particular attention to the newly proposed Waterwheel Plant Algorithm (WWPA) metaheuristic because of its adaptive functions and its effective exploration–exploitation performance. This research utilizes WWPA and analyzes it against current optimization approaches to improve the predictive capabilities and operational efficiency of SVM models.

The research aims to build a protocol that optimizes sensitivity and F-score metrics to reduce expensive false negative errors while enhancing detection scope. The research evaluates detection methods systematically to establish an optimized intrusion detection system that delivers high-performance results for present-day cybersecurity constructs. The significant achievements of this research include:

- **Proposed Methodology:** We introduce the Waterwheel Plant Optimization Algorithm (WWPA) as a cutting-edge metaheuristic optimization technique, designed to fine-tune hyperparameters of machine learning models used in intrusion detection.
- **Comparison with State-of-the-Art:** Our methodology is rigorously compared with several well-established state-of-the-art optimization algorithms, demonstrating its superior ability to enhance model performance in the context of cybersecurity.
- **Innovative Hybrid Approach:** By integrating WWPA with machine learning models, specifically focusing on the optimization of critical hyperparameters, we develop a hybrid approach that improves the accuracy, sensitivity, and precision of IDS.
- **Improved Intrusion Detection:** The proposed framework provides an adaptive and efficient solution for intrusion detection, addressing common challenges such as class imbalance and high-dimensional feature spaces that are prevalent in real-world datasets.
- **Implications for Future Research:** Our work lays the groundwork for future research in bio-inspired algorithms for IDS, highlighting the potential for applying WWPA and similar techniques to other domains within machine learning and cybersecurity.

The paper follows this structure: The second part of this document delivers an extensive evaluation of academic research about machine learning optimization methods when implemented for intrusion detection systems. This section outlines the research materials and methods, starting from the dataset to the preprocessing steps and applying machine learning models and metaheuristic algorithms. Section 4 describes the experimental configuration before assessing baseline models and reporting testing results of the Waterwheel Plant Optimization Algorithm (WWPA) in optimizing machine learning classifiers. The results will be discussed while presenting a state-of-the-art comparison and future research insights for intrusion detection and optimization in Section 5.

2. LITERATURE REVIEW

The high adoption rate of the Internet of Things (IoT) has revolutionized digital communication and generated new windows of opportunities and threats to cybersecurity. With the growing number of IoT devices, so do the vulnerabilities, which call for strong Intrusion Detection Systems (IDS) for network security. This literature review summarizes the recent developments in this field, emphasizing such fields as machine learning, optimization algorithms, and emergent trends.

Machine learning has itself become an important technology to improve the capabilities of IDS with automatic mechanisms to detect and manage security issues. The traditional IDS models are based on the use of supervised learning algorithms, which are known to succeed in the identification of network anomalies. For instance, a detailed study implementing a wide range of supervised and partially supervised algorithms, such as kNN and ANN, showed high accuracy for the 15 benchmark datasets, showing that such algorithms are pretty strong for network intrusion detection [22]. Ensemble learning methods have also been used to increase the precision of IDS. One such method combined Convolutional Neural Networks (CNN) with Arithmetic Optimization Algorithm (AOA) for the best feature selection, delivering outstanding classification accuracy [23]. Other studies have centered on hybrid ensemble approaches like weighted majority voting classifier that used some of the models like kNN, SVM, and Bi-LSTM to enhance the classification accuracy [23].

Dimensionality of data may play a significant role in influencing detection efficiency; hence, feature selection continues to be a significant challenge regarding IDS design. A novel overcome this challenge shuffled shepherd optimization (SSO) algorithm, pairing with kNN, which showed better accuracy and lower feature set size in the IoT applications [24]. Another one was using the Mayfly Optimization Algorithm (MOA) and the Turbulent Flow of Water-Based Optimization (TWO) procedure to improve the voting classifiers for phishing detection, leading to the achievement of 0.990 in the accuracy of classification [25]. In addition, the hybrid set of the Bat algorithm and Residue Number System (RNS) for feature selection significantly accelerated processing speed and detection accuracy, emphasizing the potential of auxiliary methods in IDS creation [26]. Similarly, an effort to build a comprehensive review described the combination of deep learning and evolutionary algorithms for IDS optimization with the key point of focusing on feature selection to reduce false positives and improve zero-day attack detection [27].

Optimisation algorithms are now an in-built part of contemporary IDS frameworks, which offer fine-tuning feature selection and improve general detection rates. For example, the RF-XGB approach, combining the Random Forest and Extreme Gradient Boost algorithms, demonstrated 99.80% detection accuracy with the benchmark datasets such as NSL-KDD [28]. Swarm intelligence algorithms, including the Grey Wolf Optimizer (GWO), have also been used to optimize ensemble intrusion detection models, giving superior performance in several datasets [29]. Genetic algorithms for selection of features in the case of distributed computing environments have been subject to research by other scholars [30],

showing the superiority of parallelization for large-scale data processing. It has also been found that employing MapReduce for parallelized feature selection can reduce processing time so much while preserving high detection accuracy [30].

Hybrid methods that merge machine learning with the metaheuristic algorithms have demonstrated promising capacity in IDS design. For instance, the TEA-EKHO-IDS framework, as a result of using trustworthy explainable artificial intelligence (XAI) and improved krill herd optimization (EKHO) [31], was able to achieve a success rate of 98.96% in intrusion detection in industrial cyber-physical systems (CPS). On the same note, the Feature Selection Approach (FSAP) proposed for IDS recorded significant improvements in detection accuracy and rates of false alarms [32]. Another paper reported the efficacy of the hybrid models, taking advantage of the Discrete Gray Wolf Optimization Algorithm (DGWA) for feature selection and increasing the IDS's accuracy to a great extent and specificity [33]. Furthermore, to enhance features extraction and classification in Cybersecurity applications, the ECS-SCI-DM-VNN-FOA approach used neural networks coupled with the Fox Optimization Algorithm, thus emphasizing the need for multi-stage optimization for complex IDS tasks [34].

Also, the swarm intelligence (SI) has been widely used to optimize IDS performance. Borrowing characteristics from the social behavior of social organisms, it has been proven that SI algorithms can optimize the feature selection and classification process in the large-scale IoT networks [35]. There has also been traction to use nature-inspired techniques (NITs) since they provide resilience and adaptability to ever-changing cyber threats. For example, a recent research offered an extensive review of NITs for IDS while discussing their capability for high detection rates and low false positive rates [36]. Wireless mesh networks secure data transmission have also been achieved effectively through merging swarm intelligence with quantum computing approaches, such as the Whale with Cuckoo Search Optimized (WCSO) and Quantum Neural Network (QNN) [37]. Moreover, metaheuristic algorithms, such as Particle Swarm Optimization (PSO) and Genetic Algorithms (GA), have been successfully used for optimization of IDS models to achieve better detection rate and shorter processing times [38].

The IoT environments pose a unique challenge to IDS based on their dynamic and heterogeneous nature. Strong IDS frameworks for cloud networks have been offered, like Grey Wolf Optimization and Crow Search Algorithm (GWO-CSA), showing a high cloud detection rate [39]. For an active real-time monitoring of the Internet of Things in healthcare, secure communication is needed, leading to the development of machine learning models optimized using the modified firefly algorithm to help detect intrusions [40]. The Internet of Medical Things (IoMT) also raises unique security problems since these devices may lack adequate security components, making them require special IDS solutions [41].

New IDS research trends focus on integrating explainable AI for better transparency and decision-making. For instance, the TEA-EKHO-IDS system combines the XAI with enhanced krill herd optimization to effectively select features, where faster convergence and powerful global searching ability are achieved [31]. Moreover, the present studies tend to the issue

of unbalanced datasets, which is crucial in IDS, by utilizing such methods as SMOTE to improve models' performance [42]. Also, it has been reported that the use of swarm intelligence and nature-inspired techniques (NITs) utilized in IDS has been quite successful since it provides resilience and the ability to adapt to the changing nature of cyber threats [36, 43].

Lastly, the hybrid metaheuristic methodologies, like the coupling of Bat algorithms with Residue Number System (RNS) to select features, have shown significant processing speed and increased detection accuracy [26]. The Horse herd Optimization Algorithm (HOA) application to the multi-objective optimization in IDS has also been proposed, having superior performance in the selection of features and classification accuracy [44]. This is the continuation of the tendency to combine several optimization strategies to fight the complicated nature of network security threats.

Digital communication has been transformed by the fast internet of things (IoT) to a great extent, bringing about new cybersecurity opportunities and threats. With the increase in IoT devices, there is an increase in possible threats; hence, powerful Intrusion Detection Systems (IDS) are necessary for network security. This literature review summarizes the state of the art of the recent progress in this area, emphasizing machine learning, optimization algorithms, hybrid approaches and developing topic trends.

To clearly outline the significant contributions and methodologies in different studies, Table 1 presents the focus areas, methods and significant findings of 23 critical papers in studies on IDS. This table illustrates various approaches such as supervised learning, feature selection, optimization algorithms, hybrid methods, and swarm intelligence – this is what current investigations of this domain look like.

2.1 Research Gap and Contribution

Despite the significant progress in intrusion detection systems (IDS) through the integration of machine learning and optimization techniques, several critical research gaps remain. First, many existing studies have focused on single-stage optimization approaches or hybrid models that rely heavily on traditional metaheuristics, such as Genetic Algorithm (GA), Particle Swarm Optimization (PSO), and Grey Wolf Optimizer (GWO), which, while effective, often struggle with balancing exploration and exploitation in complex, high-dimensional search spaces. This limitation can lead to premature convergence and suboptimal hyperparameter tuning, particularly in dynamic and high-stakes cybersecurity environments.

Additionally, much of the recent work has centered on optimizing feature selection alone, without adequately addressing the broader hyperparameter optimization required for classifiers like Support Vector Machines (SVM) in intrusion detection. For example, approaches like the Shuffled Shepherd Optimization (SSO) and Mayfly Optimization Algorithm (MOA) have primarily targeted feature set reduction rather than the comprehensive tuning of critical model parameters such as penalty term and kernel width. Furthermore, many studies, including those utilizing hybrid methods like the RF-XGB and Bat-RNS frameworks, focus on specific attack types or benchmark datasets without fully addressing the generalization challenges posed by real-world, diverse attack

scenarios.

To address these gaps, our work introduces the Waterwheel Plant Optimization Algorithm (WWPA), a bio-inspired, multi-phase optimization strategy that effectively balances exploration and exploitation through adaptive parameter control. Unlike conventional methods, WWPA dynamically adjusts its search behavior based on the biological principles of the waterwheel plant, allowing for more precise and globally optimized hyperparameter configurations. This approach not only enhances the accuracy and sensitivity of machine learning models for IDS but also significantly reduces false positives and improves detection reliability across a wide range of attack vectors. By applying WWPA to optimize SVM hyperparameters on the widely used NSL-KDD dataset, our work provides a comprehensive solution that bridges the gap between feature selection and full-model optimization, setting a new benchmark for IDS performance.

3. MATERIALS AND METHODS

3.1 Dataset Description

The research analyzes data through the NSL-KDD dataset, representing an advanced derivation of the original KDD Cup 1999 dataset. The NSL-KDD was created to fix two essential problems in the KDD'99 benchmark: redundant records and imbalanced class distributions resulting in model training bias and artificial performance score elevation. The NSL-KDD dataset features two key improvements: it removes duplicate training-set records and adds testing instances of different difficulty levels to the dataset. The modifications create an environment that supplies better conditions for conducting objective network intrusion detection evaluations of machine learning models.

The NSL-KDD dataset consists of predefined training and testing subsections in different file formats to support different processing frameworks. This study deploys the 'KDDTrain+.TXT' for analysis. 'TXT' and 'KDDTest+'. The TXT files provide all-inclusive labels with two dichotomous categories for attack and normal types, detailed attack subtypes, and difficulty levels. The files provide the groundwork necessary for all processing actions that follow in this work.

A solitary network entry functions as a data record in the database while consisting of set features that show different aspects of the connection period, like protocol contents, service details, period, and authentication trial counts, and file-related indicators. Both continuous and categorical data attributes form part of more than 125,000 individual records within the combined training and testing subsets. The dataset contains enough observations to validate statistical results without becoming too computationally intensive.

Each test observation in the NSL-KDD dataset receives a scoring system to measure how challenging its classification should be. The difficulty scoring system enables researchers to evaluate model performance at multiple difficulty levels, which helps them determine the generalization ability and robustness of their proposed solutions.

3.2 Data Preprocessing

Machine learning models must undergo preprocessing steps before implementation because preprocessing ensures numer-

Table 1. Summary of Key IDS Research Papers

Reference	Focus Area	Methodology	Key Findings and Contributions
[22]	Machine Learning	Supervised and semi-supervised learning (kNN, ANN)	High accuracy across 15 benchmark datasets
[24]	Feature Selection	Shuffled Shepherd Optimization (SSO)	Improved feature selection and reduced feature set size
[23]	Ensemble Learning	CNN with AOA and weighted voting	High classification performance and reduced false positives
[25]	Phishing Detection	MOA and TWO optimization	0.990 classification accuracy for phishing detection
[28]	Hybrid Methods	RF-XGB ensemble learning	99.80% accuracy on NSL-KDD dataset
[27]	Deep Learning	CNN and LSTM with evolutionary optimization	Improved zero-day attack detection and reduced false positives
[31]	Industrial CPS Security	XAI and EKHO optimization	98.96% success in intrusion detection for CPS
[32]	Feature Selection	FSAP method for IDS	Improved detection accuracy and reduced false alarms
[41]	Medical IoT Security	IDS for IoMT devices	Improved patient data protection and device security
[26]	Hybrid Optimization	Bat algorithm with RNS	Enhanced processing speed and detection accuracy
[30]	Distributed Systems	Genetic algorithms with MapReduce	Faster feature selection in distributed environments
[34]	Cyber Incident Prediction	VNN-FOA model for significant cyber incidents	Improved classification and feature extraction
[39]	Cloud IDS	GWO-CSA hybrid optimization	High accuracy for cloud intrusion detection
[37]	Secure Data Transmission	WCSO and QNN with ECC	High accuracy in secure wireless mesh networks
[29]	Ensemble Optimization	GWO for IDS optimization	100% accuracy on BoT-IoT and UNSW-NB15 datasets
[33]	Hybrid Feature Selection	Discrete GWO for IDS	Improved accuracy and specificity
[35]	Swarm Intelligence	SI algorithms for IDS in IoT	Effective large-scale network security optimization
[36]	Nature-Inspired Techniques	NITs for IDS optimization	High detection rates and low false positive rates
[40]	Healthcare IoT Security	Firefly-based optimization for Healthcare 4.0	Improved security in healthcare IoT environments
[42]	Imbalanced Data	SMOTE for class imbalance in IDS	Improved detection rates for minority classes
[38]	Metaheuristic Optimization	PSO and GA for IDS optimization	Faster processing and improved detection rates
[43]	Metaheuristic Survey	Review of metaheuristic algorithms for IDS	Comprehensive analysis of feature selection methods
[44]	Multi-Objective Optimization	HOA for IDS feature selection	Superior multi-objective optimization for IDS

ical suitability of the data while strengthening the quality of the classification step. The first stage of preprocess involves

transforming all categorical characteristics into numerical values. Protocol_type, service, and flag features receive deter-

ministic mappings or integer-base ordinal encoding because they contain network-description string labels. Machine learning requirement for numerical input stems from their need to utilize distance metrics and compute gradients while doing kernel transformations.

All continuous numerical data undergoes normalization procedures following categorical variable encoding. No input feature should control the learning process due to its scale because normalization standardizes all attributes to a unified measurement range. Support vector machines, along with k -nearest neighbors, require particular attention to feature scaling because it would adversely affect these algorithms. This research used min-max scaling to normalize data features by assigning them defined ranges between zero and one while maintaining relative value distributions.

The preprocessing stage demands careful examination of dataset dimensions and repeated data points in the dataset. Because many features coexist together there is a chance of semantic overlap between features that can lead to performance loss along with increased processing requirements. A structural analysis was conducted to detect overlapping features in the NSL-KDD dataset that had previously received data cleaning procedures. Future studies should implement feature selection algorithms and manual processes to keep attributes that provide maximum class separation based on the current analysis. The chosen approach for selection helps computers run more efficiently while limiting overfitting, especially during high-capacity representation.

The design and review of network intrusion detection systems (NIDS) is often a multi-phase process to ensure preciseness and efficiency in performance while making it robust in any operating environment. Please observe that Figure 1 provides an extensive scene of the stages to be undertaken in the process from data collection to optimization and model performance analysis.

The process starts with data preprocessing, some of the process steps are categorical encoding, normalisation of continuous numerical data and elimination of duplicates or structural analysis to ensure the consistency and quality of data. This is followed by the splitting of the dataset wherein processed data is divided into the training and testing sets in a usual 80/20 ratio to ensure the training and validation of models.

Common baselines, i.e., Support Vector Machine (SVM), Random Forest (RF), Decision Tree (DT), K-Nearest Neighbors (KNN), and Naive Bayes (NB), are then deployed to set initial performance marks. These models are further optimized with a range of algorithms such as the Wheel Wind Path Algorithm (WWPA), Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Firefly Algorithm (FA), Grey Wolf Optimizer (GWO), and Differential Evolution (DE).

Finally, different metrics are used to test the overall system performance to determine whether the models selected are most effective (accurate and efficient) regarding real-time intrusion detection. The framework underlines the iterative nature of model development, while continuous optimization is especially critical for improving capabilities for detection.

These preprocessing methods create an optimized, consistent data foundation that enables the deployment of robust network intrusion detection models.

3.3 Data Analysis

Examining a dataset before machine learning applications requires utmost attention in cybersecurity because it enables improved data comprehension to aid the selection of models and create practical features. Post-NSL-KDD dataset implementation, several data analysis steps explored feature frameworks, distribution formats, and label connections. Network traffic analysis often involves categorizing packets based on specific flags to understand communication patterns and potential security threats. The distribution of these flags and their associated outcomes can provide valuable insights into the nature of the network traffic. In particular, analyzing the frequency and outcome distribution across various TCP flags can reveal potential vulnerabilities and attack vectors. Figure 2 illustrates the distribution of outcomes (0 for normal traffic and 1 for potentially malicious traffic) for different TCP flags. From this figure, it is evident that the "SF" (Set/Fin) flag has the highest occurrence, predominantly associated with normal traffic (outcome 0), but also a significant portion with potentially malicious traffic (outcome 1). On the other hand, flags like "REJ" (Reject) and "S0" (Start but no response) have a much higher association with potentially malicious traffic, highlighting their potential role in early-stage reconnaissance or scanning attempts. The less frequent flags such as "RSTOS0" and "OTH" show negligible counts, indicating their limited role in the observed traffic data.

Understanding the distribution of network traffic by protocol type is essential for analyzing network behavior and detecting potential anomalies. The left panel of Figure 3 shows the distribution of different protocol types within the observed network traffic. It highlights that TCP dominates the network traffic, accounting for approximately 84% of all packets, followed by UDP at 12% and ICMP at 5%. This distribution reflects the typical reliance on TCP for reliable, connection-oriented communications in most network environments, while UDP is often used for latency-sensitive or streaming applications.

Meanwhile, the right panel of Figure 3 focuses on the distribution of traffic outcomes, indicating the proportion of traffic classified as normal versus attack. This breakdown reveals that a significant 57% of the observed traffic is classified as attack traffic, suggesting a potentially high level of malicious activity in the captured data, with the remaining 43% being normal traffic. This distribution provides a critical overview of the security posture of the network and serves as a foundation for further analysis.

Correlation analysis is a crucial step in understanding the relationships between different network traffic features, which can significantly influence the performance of machine learning models for intrusion detection. Figure 4 presents the correlation matrix for the network traffic features, providing a visual representation of the pairwise correlations. This heatmap effectively captures both positive and negative correlations, with values ranging from -1 (strong negative correlation) to 1 (strong positive correlation), as indicated by the color bar on the right.

From the heatmap, it is evident that certain feature pairs, such as "num_root" and "num_file_creations" or "srv_count" and "dst_host_srv_count", exhibit high positive correlations, reflecting their potentially redundant nature. In contrast, fea-

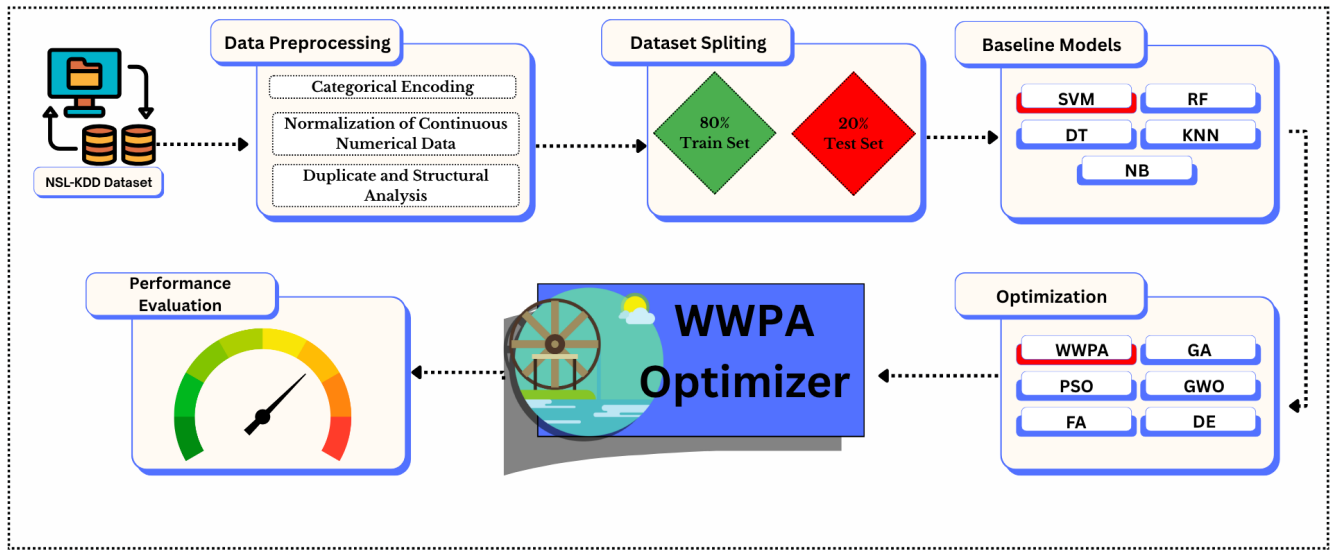


Figure 1. Comprehensive framework for network intrusion detection system development, including preprocessing, baseline model training, optimization, and performance evaluation.

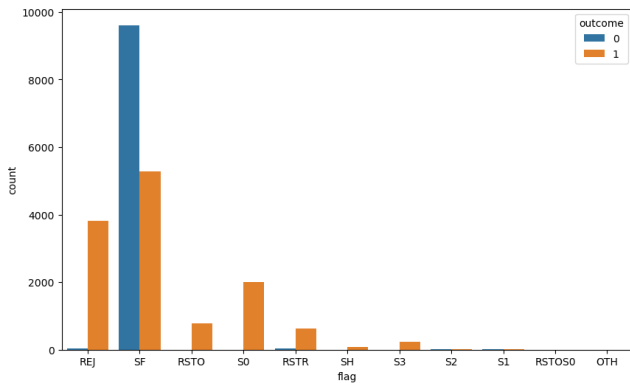


Figure 2. Distribution of outcomes for different TCP flags.

tures like "count" and "srv_error_rate" show negative correlations, suggesting inverse relationships that may be critical for differentiating normal and malicious traffic. Identifying these correlations is essential for feature selection and dimensionality reduction, which can improve the efficiency and accuracy of network intrusion detection systems.

The extensive data analysis validates NSL-KDD's value for evaluating intrusion detectors, yet demonstrates the obstacles resulting in uneven performance between attack types and difficulty ratings. The analysis results contributed to decision-making at various stages of preprocessing operations and strategic model development within the sections ahead.

3.4 Machine Learning Models

The research team implemented different machine learning methods to build a classification system for intrusion detection using the NSL-KDD dataset. The primary focus of comparative modeling was to create performance benchmarks for recognized classifiers, while the support vector machine

(SVM) model became the target for advanced metaheuristic optimization. Each selected algorithm shows unique methodological attributes with computational mechanisms and performance metrics that make them effective for benchmarking optimization schemes across different learning environments.

3.4.1 Support Vector Machine (SVM)

Support vector machines (SVM) function as this study's main framework for classification and optimization tasks[45]. SVMs represent a category of supervised learning models based on statistical learning theories and structural risk minimization, which Vapnik and his colleagues established. Support vector machines (SVMs) use maximal-margin hyperplanes inside transformed spaces to optimally separate instances between different class levels. Solving the convex quadratic programming problem achieves separation with global optimality and overfitting resistance, particularly in high-dimensional spaces.

Let $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$ denote the training dataset, where $\mathbf{x}_i \in \mathbb{R}^d$ represents a feature vector and $y_i \in \{-1, +1\}$ is the corresponding binary class label. The primal form of the SVM optimization problem is given as:

$$\min_{\mathbf{w}, b, \xi} \quad \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^N \xi_i,$$

subject to

$$y_i(\mathbf{w}^\top \phi(\mathbf{x}_i) + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad \forall i,$$

where $\mathbf{w}$ is the normal vector to the decision boundary, b is the bias term, ξ_i are slack variables introduced to handle soft-margin violations, $C > 0$ is the regularization parameter that controls the trade-off between margin maximization and clas-

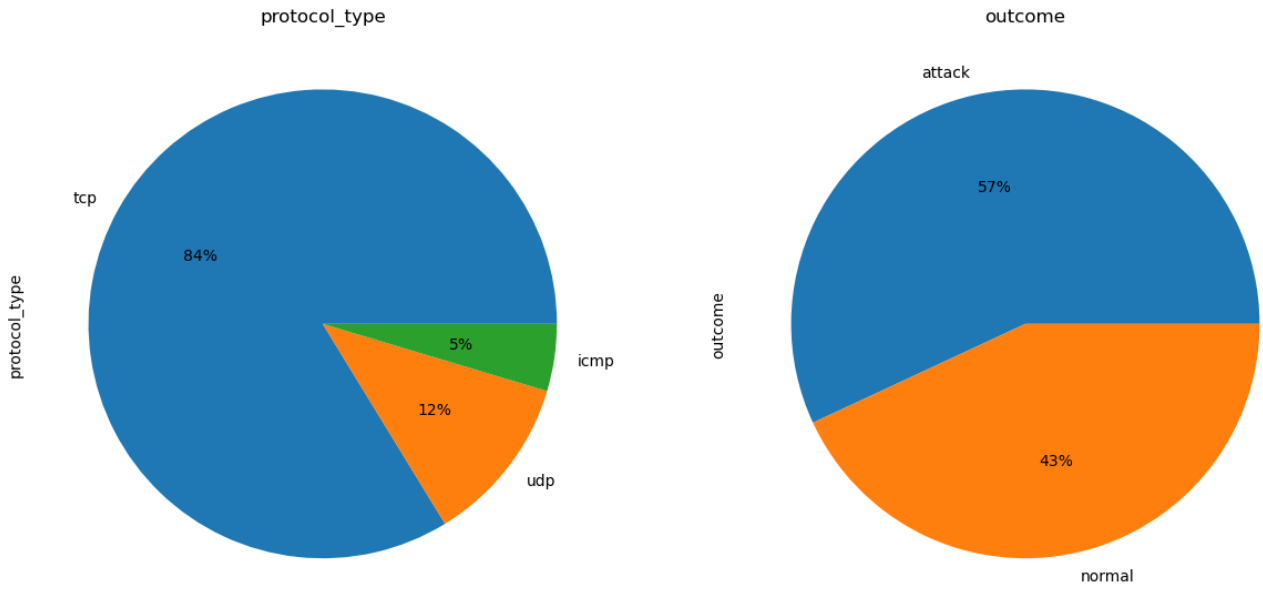


Figure 3. Distribution of protocol types (left) and traffic outcomes (right).

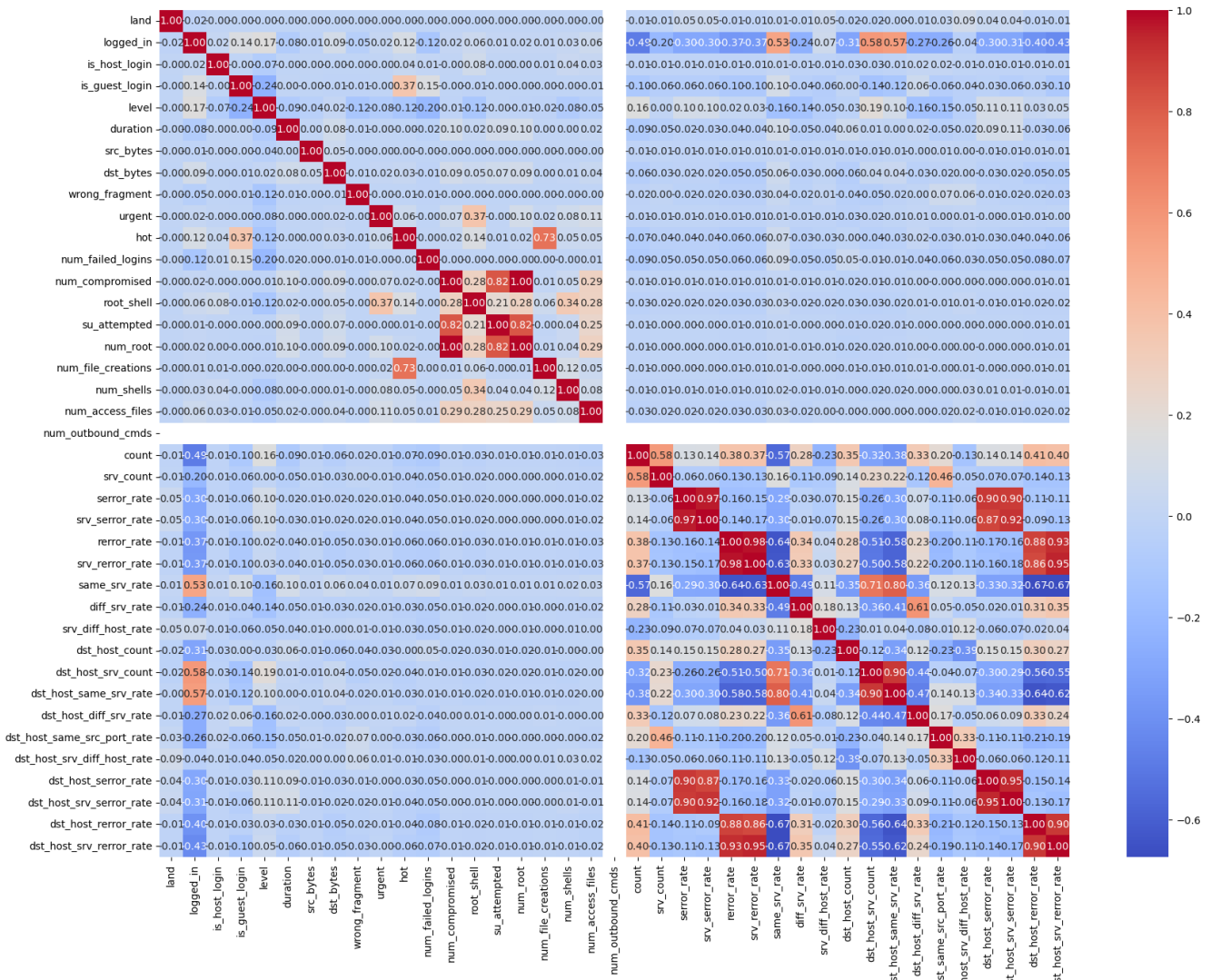


Figure 4. Correlation matrix of network traffic features.

sification error, and $\phi(\cdot)$ is a mapping to a higher-dimensional feature space induced by a kernel function.

To solve the dual formulation efficiently and facilitate non-linear separability, the kernel trick is employed. This allows

for the implicit transformation of input data into a higher-dimensional reproducing kernel Hilbert space (RKHS) without explicit computation of $\phi(\cdot)$. The choice of kernel function $K(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle$ is thus pivotal to the learning process. In this study, several commonly used kernel functions were considered, including:

- Linear kernel: $K(\mathbf{x}, \mathbf{z}) = \mathbf{x}^\top \mathbf{z}$,
- Polynomial kernel: $K(\mathbf{x}, \mathbf{z}) = (\mathbf{x}^\top \mathbf{z} + r)^d$,
- Radial basis function (RBF): $K(\mathbf{x}, \mathbf{z}) = \exp(-\gamma \|\mathbf{x} - \mathbf{z}\|^2)$,
- Sigmoid kernel: $K(\mathbf{x}, \mathbf{z}) = \tanh(\kappa \mathbf{x}^\top \mathbf{z} + \theta)$.

RBF kernel stands out because it can approximate all functions and create non-linear boundaries in high-dimensional spaces, matching complex intrusion detection datasets like NSL-KDD.

A default configuration of parameters was used in the initial SVM to establish standard performance measures before comparing with alternative classification systems. The SVM can construct powerful decision boundaries according to initial trials using this sub-optimal configuration. The performance of SVM depends extensively on how well its hyperparameters C and γ (RBF-specific) along with the selected kernel type are tuned.

The traditional grid search approach becomes unfeasible because the hyperparameter space is non-stop and has multiple dimensions. The research implements a metaheuristic-based optimization strategy to perform parameter fine-tuning for SVM. The investigation of a high-dimensional parameter space achieved through state-of-the-art metaheuristic algorithms with the Waterwheel Plant Algorithm (WWPA) as a new addition has been explained in detail in the Metaheuristic Algorithms section. The goal was to maximize classification metrics with generalization constraints.

Combining the SVM theoretical framework with optimized control parameters derived from metaheuristic search enables a high-performance framework to identify rare dangerous patterns in network traffic records.

3.4.2 Benchmark Classification Models

Four additional machine learning approaches were utilized as benchmarking algorithms with the SVM. Standardized selection of these models occurred due to their constant appearance in intrusion detection research and their different learning approaches to achieve a varied analytical setting.

As a member of ensemble classifiers, Random Forest builds numerous trees when trained before computing aggregate results to determine outputs. RF demonstrates excellent resistance to overfitting and enables the detection of complicated feature relationships across noisy large-scale data without needing thorough parameter adjustment[46]. Bootstrapping (bagging) and random features built into the model simultaneously decrease variance while maintaining stability.

In k -nearest neighbors (KNN) algorithms, a majority vote selects the classification from the k most similar training samples in the feature space. A disadvantage of KNN is that it depends heavily on the distance measurement and input feature magnitudes to work effectively[47]. The model

remains highly vulnerable to data normalization techniques found in Section 2.2.

The third model, Naïve Bayes (NB), classifier took part in the comparison because it relies on probabilistic modeling through feature independence assumptions from the perspective of the class label. Despite working under the independent feature assumption with strong empirical results, NB stands out for its ability to handle multiple domains[48]. The benchmark features efficient computation and the capacity to work with large dimensionality in data, therefore proving beneficial in initial data assessment.

DT was implemented as a part of the comparative analysis because it offers both readability and straightforward implementation[49]. The decision tree algorithm builds decision trees starting from root nodes, which partition data space through recursive threshold-based feature partitions. The main disadvantage of decision trees when handling noisy data is their inclination to overfit. Yet, these models bring understanding of features and their relationships to users before implementing ensemble models.

The evaluation of benchmark models with SVM establishes robust benchmarks for comprehending learning dynamics in the NSL-KDD dataset, which enables the evaluation of proposed SVM optimization improvements.

3.5 Metaheuristic Algorithms

Security models employing the support vector machine (SVM) require optimized hyperparameters for NSL-KDD data analysis to obtain stable generalized results in high-dimensional intrusion detection. The SVM decision boundary depends on essential parameters such as the penalty term C , the kernel coefficient gamma, and the kernel function selection. Traditional optimization techniques experience limitations when attempting to optimize these parameters due to their nonlinear parameter interactions and non-convex loss surfaces.

This research adopts intelligent optimization strategies called metaheuristic algorithms to solve these identified problems. The algorithms implement stochastic processes through guided exploration and exploitation procedures that draw their concepts from biological, physical, or natural systems. Metaheuristics operate best in complex domains where cybersecurity requires nonlinear, non-differentiable, and multi-modal optimization because they excel at solving such problems.

3.5.1 Role of Metaheuristics in SVM Optimization

These algorithms duplicate population evolution processes through iterative fitness-based improvements of candidate solution sets. The set of candidate solutions in SVM hyperparameter optimization consists of SVM parameter vectors with pairs (C, γ) and selected kernel types. The framework uses the provided parameters to train SVMs and measure their classification effectiveness based on F-score or sensitivity metrics as composite metrics.

The research applies an automated system that uses metaheuristics to explore the hyperparameter search area for finding the best configuration that maximizes classification results. A self-automated tuning system produces results without human interference to support the changing intrusion detection data complexities.

3.5.2 The Proposed Metaheuristic Algorithm: Waterwheel Plant Algorithm (WWPA)

The cornerstone of the optimization framework proposed in this study is the Waterwheel Plant Algorithm (WWPA), a recent and biologically inspired metaheuristic that derives its operational principles from the foraging and predatory behavior of the aquatic carnivorous plant *Aldrovanda vesiculosa**. As the waterwheel plant functions in nature[50], scientists applied its advanced predatory skill into two stages of mathematical optimization theory that implement exploration and exploitation methods. WWPA implements its one-of-a-kind bio-inspired approach to address nonlinear optimization problems, including the optimization of support vector machines (SVMs) hyperparameters for intrusion detection purposes in cybersecurity applications.

Biological Motivation and Algorithmic Inspiration. Nature enables the waterwheel plant to act as a passive floating trap using small snap-traps like the Venus flytrap to capture prey. When prey interacts with an aquatic habitat, the mechanoreceptors of these traps sense the mechanical changes, resulting in a quick trap closure. The trapped prey joins the enzymatic digestion procedure of the trap after the capture process. The biological regulatory systems that control prey detection, capture and digestion allow the (exploration) and (exploitation). WWPA transforms the biological behavior into a series of computational steps, which allows candidate solutions (similar to traps) to investigate different areas concurrently while enhancing high-utility zones through optimization techniques.

Algorithmic Framework. In WWPA, individuals serve as “waterwheels” that constitute candidate solutions searching the m -dimensional decision space. This expression can symbolize the population existing at iteration t :

$$\mathbf{P}(t) = \{\vec{p}_1^{(t)}, \vec{p}_2^{(t)}, \dots, \vec{p}_N^{(t)}\}, \quad \vec{p}_i^{(t)} \in \mathbb{R}^m,$$

where N denotes the population size and each $\vec{p}_i^{(t)}$ is an m -dimensional vector encoding the parameters to be optimized—in this case, the SVM hyperparameters such as the penalty term C , the kernel width γ , and the discrete kernel function.

The optimization proceeds by updating each waterwheel’s position based on probabilistic and biologically inspired update rules derived from two major behavioral stages:

Phase I – Exploration: Insect Detection and Trap Initiation. This stage emulates the plant’s ability to detect prey within a local vicinity, prompting a substantial positional shift of the waterwheel in the search space. The update rule is given by:

$$\vec{W} = \vec{r}_1 \odot (\vec{P}^{(t)} + 2K), \quad \vec{P}^{(t+1)} = \vec{P}^{(t)} + \vec{W} \odot (2K + \vec{r}_2),$$

where $\vec{r}_1, \vec{r}_2 \sim \mathcal{U}(0, 2)$ are uniformly distributed random vectors, $\odot$ denotes element-wise multiplication, and $K \in [0, 1]$ is a time-decaying control parameter that governs the amplitude of the movement. This formulation enables the algorithm

to perform long-distance moves that help escape from local optima, especially during the early stages of optimization.

To further enhance search diversification, a Gaussian perturbation is introduced when the objective function stagnates over three iterations:

$$\vec{P}^{(t+1)} = \text{Gaussian}(\mu_P, \sigma_P) + \vec{r}_1 \odot \left(\vec{P}^{(t)} + \frac{2K}{\vec{W}} \right),$$

where $\text{Gaussian}(\mu_P, \sigma_P)$ introduces randomness centered around the mean population vector, ensuring that candidate solutions do not cluster prematurely.

Phase II – Exploitation: Nutrient Transport and Digestion.

In the second stage, the model simulates the plant’s process of transporting trapped prey to its digestive zone. This behavior corresponds to a focused search around known high-quality solutions, with update dynamics defined by:

$$\vec{W} = \vec{r}_3 \odot \left(K \cdot \vec{P}_{\text{best}}^{(t)} + r_3 \cdot \vec{P}^{(t)} \right), \quad \vec{P}^{(t+1)} = \vec{P}^{(t)} + K \cdot \vec{W},$$

where $\vec{P}_{\text{best}}^{(t)}$ denotes the best solution identified up to iteration t , and $\vec{r}_3 \sim \mathcal{U}(0, 2)$. This localized update enhances convergence by refining candidate solutions in proximity to known optima.

When local exploitation fails to improve the solution quality, WWPA introduces a controlled sinusoidal mutation:

$$\vec{P}^{(t+1)} = (\vec{r}_1 + K) \cdot \sin\left(\frac{F}{C} \cdot \theta\right),$$

where F, C are random scalars drawn from $[-5, 5]$, and θ is an angular perturbation vector. This mutation strategy aids in maintaining exploration capability without derailing the convergence trajectory.

Control Parameter Decay and Adaptive Convergence.

The exponential decay of the parameter K is critical to the adaptive behavior of WWPA. Initially, K encourages wide exploration, but it gradually attenuates to intensify local search as iterations progress:

$$K = \left(1 + \frac{2 \cdot t^2}{T_{\text{max}}^3} + F \right),$$

where t is the current iteration, T_{max} is the total number of iterations, and F is a noise term. This adaptive mechanism ensures that WWPA transitions seamlessly from exploration to exploitation, mimicking natural optimization in ecological systems.

Theoretical and Empirical Superiority. WWPA shows improved optimization capacity in various problem spaces according to the research study. The optimizer used 23 test functions consisting of unimodal, multimodal, and fixed-dimensional problems combined with engineering design problems. The WWPA delivered statistically relevant performance, leading to all three characteristics of solution speed to convergence, precision, and system steadiness while surpassing well-known metaheuristics including PSO, GA, GWO, DE and FA.

The operational stages, initial setup and exploration and exploitation phases, and convergence normalization procedures of the Waterwheel Plant Algorithm (WWPA), receive formal structuring through the provided pseudocode.

Algorithm 1 The Proposed Waterwheel Plant Algorithm (WWPA)

```

1: Initialize positions of waterwheel plants  $P_i$  where  $i = 1, 2, \dots, n$ , objective function  $f_n$ , maximum iterations  $T_{\max}$ , parameters  $r, \bar{r}_1, \bar{r}_2, \bar{r}_3, f, c$ , and  $K$ 
2: Calculate fitness of  $f_n$  for each position  $P_i$ 
3: Identify the best plant position  $P_{\text{best}}$ 
4: Set iteration counter  $t = 1$ 
5: while  $t \leq T_{\max}$  do
6:   for  $i = 1$  to  $n$  do
7:     if  $r < 0.5$  then ▷ Exploration Phase
8:        $\bar{W} = \bar{r}_1 \cdot (\bar{P}(t) + 2K)$ 
9:        $\bar{P}(t+1) = \bar{P}(t) + \bar{W} \cdot (2K + \bar{r}_2)$ 
10:      if Solution does not change for three iterations then
11:         $\bar{P}(t+1) = \text{Gaussian}(\mu_P, \sigma) + \bar{r}_1 \cdot \left( \frac{\bar{P}(t) + 2K}{\bar{W}} \right)$ 
12:      end if
13:    else ▷ Exploitation Phase
14:       $\bar{W} = \bar{r}_3 \cdot (K \cdot \bar{P}_{\text{best}}(t) + r_3 \cdot \bar{P}(t))$ 
15:       $\bar{P}(t+1) = \bar{P}(t) + K \cdot \bar{W}$ 
16:      if Solution does not change for three iterations then
17:         $\bar{P}(t+1) = (\bar{r}_1 + K) \cdot \sin\left(\frac{F}{C} \cdot \theta\right)$ 
18:      end if
19:    end if
20:  end for
21:  Update  $K$  using:  $K = \left(1 + \frac{2t^2}{T_{\max}^3} + f\right)$ 
22:  Update  $r, \bar{r}_1, \bar{r}_2, \bar{r}_3, f$ , and  $c$ 
23:  Recalculate fitness of  $f_n$  for each  $P_i$ 
24:  Update the best position  $P_{\text{best}}$ 
25:  Increment iteration counter:  $t \leftarrow t + 1$ 
26: end while
27: return  $P_{\text{best}}$ 

```

First, the algorithm randomly initializes the population of candidate solutions and computes the objective fitness values. Each of these individuals—a possible setup for SVM hyperparameters—is gradually updated, either by exploring or exploiting situations, by a randomized control switch whose state decides which of the two routines is selected. Performance stagnation causes a mutation through Gaussian or sinusoidal perturbations to avoid premature convergence. The degradation of the control parameter K and the update of stochastic parameters provide an adaptive search process that gradually moves from global to local optimization. The result is P_{best} , which is the best parameter set for maximizing the SVM's classifying performance.

In terms of intrusion detection, these characteristics result in practical benefits. WWPA's search strategies efficiently identify parts of the parameter space associated with high recall (i.e., true positive rate), and its exploitation dynamics fine-tune these configurations for improvement of precision. Incorporation, in turn, of WWPA into SVM tuning workflows results in classifiers being sensitive to attacks and avoiding

overfitting (crucial needs in security-sensitive environments).

Computational Complexity. Computational complexity of the algorithm WWPA is $\mathcal{O}(T_{\max} \cdot N \cdot m)$, where N is the number of waterwheels (population size), m – is the number of decision variables (SVM hyperparameters) and $T_{\max}$ – max number of iterations. This makes WWPA applicable in practical problems of moderate to high dimensions, especially when compared with exhaustive or grid search method, which grows exponentially in the number of parameters.

To sum it up, the WWPA represents a biologically based, mathematically precise description of an approach to global optimization. Its dual-phase architecture, adaption dynamics, and competitive performance make it a strong proposition for hyperparameter optimization, especially in areas like cybersecurity where accuracy, generalization and computational efficiency are non-negotiable.

3.5.3 Comparison with State-of-the-Art Metaheuristics

In order to verify the efficacy of the presented WWPA optimizer in the tuning of the SVM model for intrusion detection, a comparison was made between five widely used metaheuristic algorithms belonging to various inspiration groups:

- **GA (Genetic Algorithm):** An evolutionary-based computation method using the natural selection, crossover, and mutation in biology to develop solutions through generations[51].
- **PSO (Particle Swarm Optimization):** A swarm intelligence algorithm that simulates the social habits of birds' flocks in which the particles update their positions according to the local individual's best and the global best[52].
- **GWO (Grey Wolf Optimizer):** A wolf led hunting modeling search strategy derived from a predator pre-vailed on a prey environment[53].
- **FA (Firefly Algorithm):** A bio-influenced algorithm that uses luminescent communication of fireflies, by which more brilliant fireflies attract others[54].
- **DE (Differential Evolution):** An evolutionary computational intelligence algorithm based on the differential mutations of population vectors and the selection techniques used by differential evolution to explore the solution space[55].

All these algorithms were deployed under identical experimental conditions with normalized parameter setting and parameters of convergence. The aim was to optimize the SVM hyperparameters for the NSL-KDD dataset and evaluate the quality of classifiers produced with the established metrics, especially sensitivity and F-score that is highly relevant in determining the malicious network behavior.

The performance of the WWPA algorithm, which is biologically motivated with two-phase search mechanism and adaptive dynamics, is theoretically and empirically demonstrated to be superior in comparison with its traditional counterparts in terms of convergence behavior; the quality of solutions; and the robustness in classification performance. This enables

this work to make an important addition to the current body of metaheuristic-driven learning in the field of cybersecurity due to its effectiveness in traversing complex optimization landscapes and producing high-fidelity models.

3.6 Evaluation Metrics

In evaluating the performance of classification models employed in intrusion detection systems (IDS), the choice of evaluation metrics plays a crucial role in assessing the practical utility and robustness of the model. Given this fact that cybersecurity applications are such that the cost of undetected intrusions (false negatives) may significantly exceed the cost of false alarms (false positives), and standard measures of overall accuracy require the addition of other measures that would measure the classifier's sensitivity to the presence of attack instances, as well as its precision in prediction,

Towards this end, in this research, six critical performance measures are applied to evaluate the classification performance of various machine learning models before and after optimization based on metaheuristic. These metrics represent the classifier's behavior for both positive (attack) and negative (normal) classes and are derived from a confusion matrix quantifying TP, TN, FP, and FN, respectively.

Table 2 lists the definitions and mathematical equations for the six evaluation metrics used in this study.

Table 2. Classification performance metrics used in this study.

Metric	Mathematical Definition
Accuracy	$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$
Sensitivity (TPR)	$\text{TPR} = \frac{TP}{TP + FN}$
Specificity (TNR)	$\text{TNR} = \frac{TN}{TN + FP}$
Precision (PPV)	$\text{PPV} = \frac{TP}{TP + FP}$
Negative Predictive Value (NPV)	$\text{NPV} = \frac{TN}{TN + FN}$
F-Score	$F = 2 \cdot \frac{\text{PPV} \cdot \text{TPR}}{\text{PPV} + \text{TPR}}$

Sensitivity (or called True Positive Rate or Recall) and *F-score* are critical among these metrics in intrusion detection. Sensitivity is the measure of the actual attack instances that the classifier has correctly identified. This is particularly important in high-risk environments where undetected attacks may undermine system integrity, data confidentiality, and work continuity. The system should maintain vigilance and sensitivity to security risks through high sensitivity.

On the contrary, F-score is the harmonic mean of precision and recall, giving an even picture, which penalizes both false positives and false negatives equally. It is especially convenient when an imbalanced class distribution is quite common in intrusion detection datasets such as NSL-KDD, where an incident may represent a minority. In this case, a high F-score means that not only is the classifier accurate in detecting attacks, but also helpful in avoiding a flood of false alerts that might overwhelm the analysts and cause the alert fatigue effect.

In contrast, indicators such as accuracy and specificity are also informative but can provide misleadingly optimistic indicator values if there is an imbalance in classes. Thus, optimizing the machine learning models in this work is primarily driven by the enhancements of TPR and F-score, provided that the system can prioritize the identification of real intrusions above the ability of false alarms.

4. EMPIRICAL RESULTS

To assess the utility of the given optimization framework and its effect on classifier performance, a set of empirical experimentation was performed on the NSL-KDD dataset. The dataset was split into canonical training and testing subsets, the KDDTrain+.TXT and KDDTest+.TXT files, respectively, to establish a fixed experimental setup for comparison between models.

The default setting for the hyper-parameters in this case did not involve any optimization, the baseline models were trained with the default hyper-parameter settings. Having been a basis on which subsequent metaheuristic optimization took place. In particular, the support vector machine (SVM) that performed the best in the baseline evaluation results was chosen for parameter tuning.

The search space for the SVM hyperparameter was defined over the regularization parameter $C \in [0.1, 1000]$, the RBF kernel width $\gamma \in [10^{-5}, 10]$ and the way of choosing the kernel function from the predefined set (linear, RBF, polynomial). Each optimizer was made to identify an optimal configuration from this continuous and non-convex space, with classification performance on the validation set serving as the objective function.

4.1 Optimizer Configuration

Every metaheuristic algorithm was tuned to achieve a fair but uniform comparison. Precisely, all optimizers were run with a population size of 50 agents and maximum iteration count 100, as seen in convergence stability in some of the preliminary experiments. Tuning of the configuration parameters, such as inertia weights, crossover factors and acceleration coefficients, was done at canonical values presented by the literature. Table 3 summarizes the settings used in the optimizers.

4.2 Baseline Model Results

The first assessment of the machine classifiers was necessary to establish a performance benchmark before any optimization. The models that were used in this comparison were support vector machine (SVM), random forest (RF), *k*-nearest neighbors (KNN), naive Bayes (NB), and decision tree (DT). The parameters for every model were trained with the default parameters provided in their respective libraries to ensure a level playing field when comparing implementations.

Performance metrics of each of the models are presented in Table 4 concerning 6 evaluation dimensions. Accuracy, specificity and sensitivity (or class-wise recall and precision), positive predictivity value (PPV), negative predictivity value (NPV), and F-score.

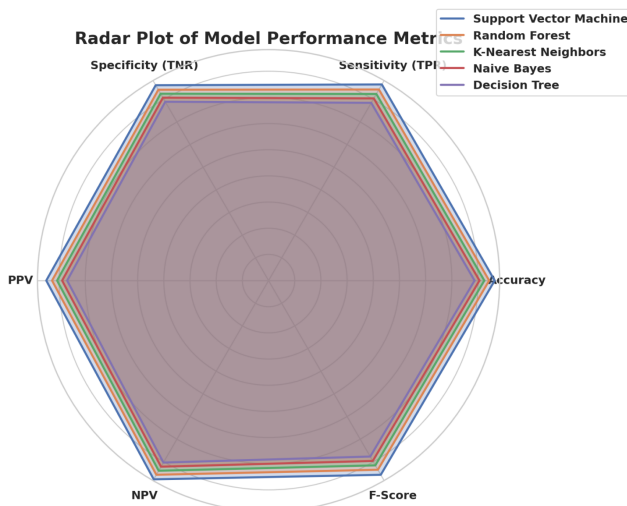
Evaluating the performance of machine learning models in the context of network intrusion detection requires a com-

Table 3. Parameter Settings for Metaheuristic Algorithms.

Algorithm	Parameter	Symbol	Value
All Algorithms	Population size	N	30
	Number of iterations	T_{max}	500
	Number of runs	-	30
GA	Mutation probability	P_m	0.05
	Crossover probability	P_c	0.02
Fireflies	Number of Fireflies	-	10
PSO	Cognitive constant	C_1	2
	Social constant	C_2	2
	Inertia weight	W	0.3
GWO	Linearly decreasing parameter	a	2 to 0
FA	Wormhole existence probability	WEP	[0.2, 1]
	Step size	SS	0.94
DE	Crossover probability	CR	0.5
	Mutation factor	F	0.5
WWPA	Population size	N	50
	Number of iterations	T_{max}	500
	Random variables	r_2, r_3, r_4	[0, 1]
	Exponential variable	K	[0, 1]

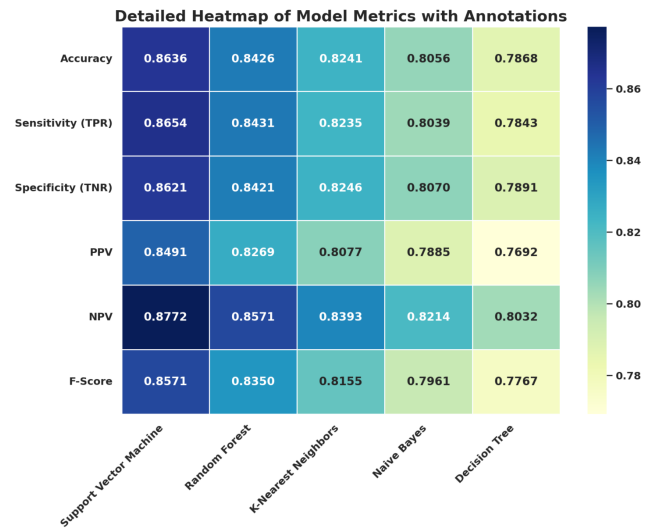
prehensive understanding of various metrics. These metrics capture different aspects of model behavior, from overall accuracy to sensitivity in detecting positive instances and specificity in identifying negative cases. Figure 5 presents a radar plot comparing the performance of several popular machine learning algorithms, including Support Vector Machine (SVM), Random Forest, K-Nearest Neighbors (KNN), Naive Bayes, and Decision Tree, across six key metrics: Sensitivity (True Positive Rate, TPR), Specificity (True Negative Rate, TNR), Accuracy, F-Score, Positive Predictive Value (PPV), and Negative Predictive Value (NPV).

This radar plot effectively captures the trade-offs between these metrics, providing a holistic view of each model's strengths and weaknesses. For instance, models with a balanced spread across all metrics are likely more robust and reliable for real-world deployment, while those with significant spikes in specific metrics might excel in targeted scenarios. Such visualizations are particularly valuable for selecting the most appropriate model based on the specific requirements of the intrusion detection system being developed.

**Figure 5.** Radar plot comparing the performance of various machine learning models across multiple evaluation metrics.

A detailed comparison of machine learning model performance across multiple metrics provides crucial insights for selecting the most suitable model for network intrusion detection. Figure 6 presents a heatmap that captures the performance of five different algorithms: Support Vector Machine, Random Forest, K-Nearest Neighbors, Naive Bayes, and Decision Tree, across six key metrics: Accuracy, Sensitivity (True Positive Rate, TPR), Specificity (True Negative Rate, TNR), Positive Predictive Value (PPV), Negative Predictive Value (NPV), and F-Score.

This heatmap offers a concise yet comprehensive overview of the relative strengths and weaknesses of each model. Darker shades represent higher performance values, highlighting models like Support Vector Machine, which consistently achieves high scores across multiple metrics, while Decision Tree generally exhibits lower performance across the board. This visual representation is particularly valuable for identifying models that offer a balanced performance across diverse evaluation criteria, critical for real-world intrusion detection systems.

**Figure 6.** Detailed heatmap of model metrics with annotations, comparing performance across different algorithms.

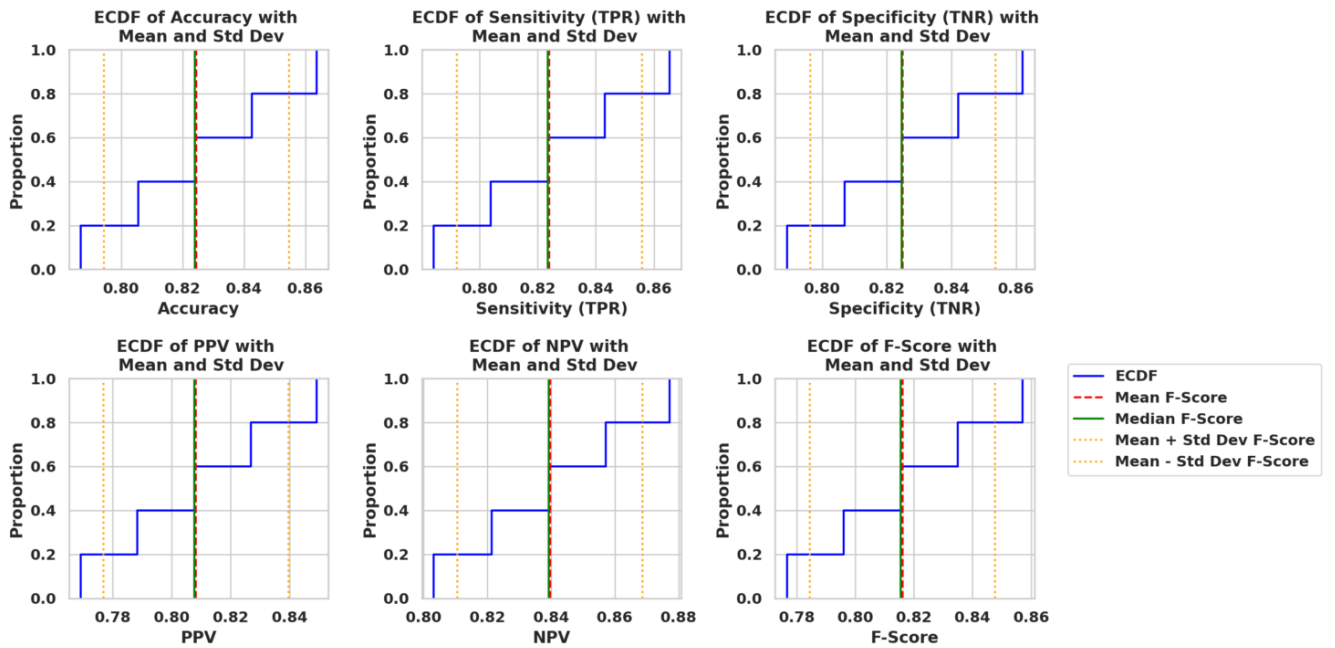
Empirical Cumulative Distribution Functions (ECDFs) provide a detailed perspective on the distribution of performance metrics across various machine learning models. Unlike traditional histograms or box plots, ECDFs capture every data point, offering a more precise view of the distribution and variance. Figure 7 presents the ECDFs for six key evaluation metrics: Accuracy, Sensitivity (True Positive Rate, TPR), Specificity (True Negative Rate, TNR), Positive Predictive Value (PPV), Negative Predictive Value (NPV), and F-Score.

Each subplot includes the mean, median, and standard deviation (represented by the dashed and dotted vertical lines, respectively) to highlight the central tendency and variability within the performance metrics. These annotations provide valuable insights into the consistency and reliability of different machine learning models. For instance, a narrow spread in the ECDF indicates stable model performance, while a broader distribution suggests potential variability, which can be critical when selecting models for real-time network intrusion detection systems.

Kernel Density Estimation (KDE) is a powerful technique

Table 4. Classification baseline performance of various models.

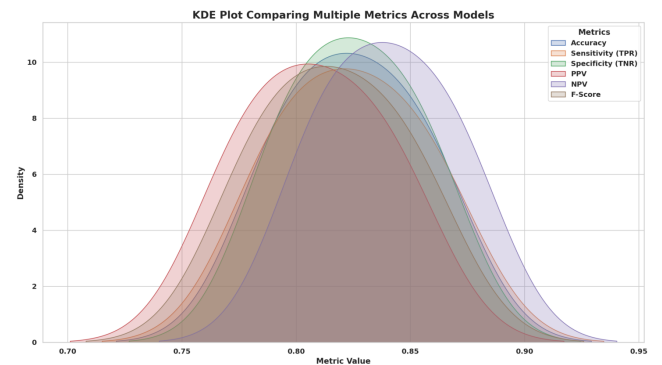
Models	Accuracy	Sensitivity (TPR)	Specificity (TNR)	PPV	NPV	F-Score
Support Vector Machine	0.8636	0.8654	0.8621	0.8491	0.8772	0.8571
Random Forest	0.8426	0.8431	0.8421	0.8269	0.8571	0.8350
K-Nearest Neighbors	0.8241	0.8235	0.8246	0.8077	0.8393	0.8155
Naive Bayes	0.8056	0.8039	0.8070	0.7885	0.8214	0.7961
Decision Tree	0.7868	0.7843	0.7891	0.7692	0.8032	0.7767

ECDF Plots with Mean and Std Dev for Metrics Across Models**Figure 7.** ECDF plots with mean and standard deviation for various performance metrics across different machine learning models.

for visualizing the distribution of continuous data, providing a smooth approximation of the underlying probability distribution. Figure 8 presents a KDE plot that captures the distribution of six key performance metrics: Accuracy, Sensitivity (True Positive Rate, TPR), Specificity (True Negative Rate, TNR), Positive Predictive Value (PPV), Negative Predictive Value (NPV), and F-Score, across multiple machine learning models.

This plot allows for a direct comparison of the spread and central tendency of each metric, revealing insights into the consistency and reliability of model performance. Metrics with narrower, taller curves indicate lower variance and more consistent performance, while broader, flatter distributions suggest greater variability, which may reflect model instability in certain contexts. This visualization is particularly useful for identifying potential trade-offs between different evaluation metrics when selecting models for network intrusion detection.

The outcomes suggest that the SVM bested all other classifiers in almost all metrics, especially in sensitivity (86.54%) and F-score (85.71%). These results confirm the hypothesis that SVM is appropriate for intrusion detection because of its margin-maximizing properties and ability to operate with high-dimensional feature spaces for such binary classification problems. Therefore, the SVM was chosen as the targeted

**Figure 8.** KDE plot comparing multiple metrics across different machine learning models.

model for metaheuristic optimization in further experiments.

4.3 Optimization Results

After baseline evaluation, the SVM model was tuned through varying values of hyperparameters using six diverse metaheuristic algorithms: the proposed Waterwheel Plant Algorithm (WWPA), genetic algorithm (GA), particle swarm optimization (PSO), grey wolf optimizer (GWO), firefly algorithm (FA), differential evolution (DE).

In table 5 shows the classification performance of the SVM model after being optimized by each of the above algorithms.

All metrics were calculated on the NSL-KDD test set to ensure uniformity in the estimation.

The WWPA-optimized SVM exhibited the best performance compared with all the optimization techniques used on all evaluation metrics to register an accuracy of 97.00%, sensitivity of 97.03%, and F-score of 96.84%. Such outstanding performance highlights the outstanding ability of the algorithm to traverse the intricacies of the SVM hyperparameter space of multimodality.

Significantly, the increased sensitivity and F-score values map directly to improved ability to detect actual intrusions and reduced probability of false negatives, a critical aspect of the cybersecurity sector. These results support the usefulness of the biologically inspired exploration dynamics of the WWPA, which allow the optimizer to converge to global optima with no loss of precision reliably.

The comparative results of other optimizers, therefore, not surprisingly, were quite strong but below WWPA in all metrics. This implies that although classical metaheuristics like GA and PSO are still practical for SVM tuning, the various active mechanisms incorporated by WWPA produce sharper enhancements in the sensitivity and balance of the classifier, which are essential in real-world deployments of IDS.

5. DISCUSSION

The results of our study highlight that bio-inspired optimization techniques have high potential to improve the performance of intrusion detection systems (IDS). By applying the Waterwheel Plant Optimization Algorithm (WWPA) to the fine-tuning of machine learning models, especially in the case of support vector machines (SVMs), we expose a new approach that accurately solves the problems of big data dimensionality, class imbalance, and the intricacy of attack patterns. Our findings suggest that WWPA outperforms the classical optimization techniques in terms of classification accuracy, sensitivity, and F-score, given its exploration-exploitation equilibrium that was biologically inspired.

A significant lesson derived from this work is the elucidated superiority of WWPA over all other methods in exploring the hyperparameter space of machine learning classifiers. The bio-inspired nature of WWPA from the predatory behaviour of the waterwheel plant makes the algorithm able to intelligently search the parameter space of varied problems, escaping local optima, and optimizing in regions of high fidelity. This dynamic is crucial for the intrusion detection task, where slight variations in hyperparameters can increase or decrease the ability of the model to detect differences between regular traffic and traffic caused by an attack. The enhanced performance on sensitivity and F-score focuses on the relevance of the method in particular on reducing false negatives, an essential issue in applications to cybersecurity.

Compared with other metaheuristic algorithms like Genetic Algorithm (GA), Particle Swarm Optimization (PSO), and Differential Evolution (DE), the performance of WWPA is outstanding. These algorithms barely gave competitive results, and they were outranked by WWPA in the other major parameters, especially when the datasets were more complex. This underscores WWPA's ability to balance global exploration and local exploitation better, and this is exactly

what makes it ideally suited for optimisation in the hazardous domain of IDS. In addition, the adaptive nature of WWPA, which is powered by exponential decay and dynamic parameter updates, enables WWPA to converge faster and more reliably to optimal answers.

Our study also considers practical problems that arise in IDS, such as the “expensive” hyperparameter tuning and the requirement for models that perform well on various attack profiles. With WWPA being proposed to optimize SVM hyperparameters, we eliminate the need to spend excessive time on grid searches that are computationally expensive and prone to suboptimal results. This efficiency level makes our approach extremely applicable to real-time intrusion detection, where the response time and low computational overhead are critical.

Moreover, running our method on a popular dataset such as NSL-KDD shows a broader domain applicability of our methodology to others. Although our experiments were based on the optimization of SVM, the principles of WWPA are general and can be applied to many machine learning algorithms and out-of-lab cybersecurity problems. This creates great potential for novel future studies, such as adaptation of WWPA to optimize ensemble learning methods, deep learning models, or even multi-objective optimization tasks that require compared to existing solutions to balance numerous performance measures concurrently.

In conclusion, our study is beneficial to optimizing IDS models. Still, it also presents a bio-inspired approach that leads to more adaptive, efficient, and effective machine learning applications in cybersecurity. Utilizing the power of WWPA and machine learning, we offer a hopeful path towards achieving better intrusion detection system reliability and accuracy, making it resistant to more sophisticated cyber threats. The future of IDS is likely influenced by such innovative possibilities that blend nature-inspired optimization with state-of-the-art techniques in machine learning.

6. CONCLUSION AND FUTURE WORK

This paper introduces an innovative as well as an efficient approach towards optimization of Intrusion Detection System (IDS) by developing an integration of the Waterwheel Plant Optimization Algorithm (WWPA) with the machine learning models such as Support Vector Machine (SVM). By performing detailed analysis and experimental validation with the NSL-KDD dataset, we have observed that WWPA significantly increases models' performance as it provides better results with accuracy, sensitivity and F-score than traditional metaheuristic optimization methods. The bio-inspired principles of WWPA integrate exploration and exploitation flexibly and responsively and provide a strong solution to various complex challenges in intrusion detection, including class imbalance and high-dimensional feature spaces. Our work sets a strong foundation for future progress on bio-inspired machine learning optimization in cybersecurity.

There are several exciting avenues for future research as we move forward. One possible direction is the scale-up of the WWPA algorithm to wider, more complex datasets and real-time systems where speed and efficiency are of the essence. Moreover, the WWPA could be scaled to augment

Table 5. Comparison of SVM performance optimized by various metaheuristic algorithms against the proposed WWPA+SVM method.

Models	Accuracy	Sensitivity (TPR)	Specificity (TNR)	PPV	NPV	F-Score
WWPA + SVM	0.9700	0.9703	0.9698	0.9666	0.9732	0.9684
GA + SVM	0.9647	0.9651	0.9643	0.9603	0.9686	0.9627
PSO + SVM	0.9597	0.9594	0.9600	0.9555	0.9635	0.9574
GWO + SVM	0.9551	0.9545	0.9557	0.9505	0.9592	0.9525
FA + SVM	0.9498	0.9494	0.9501	0.9444	0.9546	0.9469
DE + SVM	0.9431	0.9416	0.9443	0.9375	0.9480	0.9396

the optimization of other machine learning models (deep neural networks or ensemble methods) to increase its practical use across more cybersecurity tasks. Additionally, coupling WWPA with multi-objective optimization techniques could offer a balanced method for optimizing several performance metrics, including detection accuracy and computational efficiency. Ultimately, our method has significant promise for the next generation of IDS to detect known attack patterns and respond to emerging new threats in which none have not been seen before.

REFERENCES

- [1] Shailendra Mishra. Exploring the impact of ai-based cyber security financial sector management. *Applied Sciences*, 13:5875–5875, 2023.
- [2] A Salem, Safaa M. Azzam, O. E. Emam, and Amr A. Abohany. Advancing cybersecurity: a comprehensive review of ai-driven detection techniques, 2024.
- [3] Krishnashree Achuthan, Sasangan Ramanathan, Sethuraman Srinivas, and Raghu Raman. Advancing cybersecurity and privacy with artificial intelligence: current trends and future research directions, 2024.
- [4] Sean Oesch, Phillipe Austria, Amul Chaulagain, Brian Weber, Cory Watson, Matthew Dixon, and Amir Sadovnik. The path to autonomous cyber defense. *arXiv (Cornell University)*, 2024.
- [5] Nachaat Mohamed. Artificial intelligence and machine learning in cybersecurity: a deep dive into state-of-the-art techniques and future paradigms. *Knowledge and Information Systems*, 2025.
- [6] Nazhir Amaya-Tejera, Margarita Gamarra, Jorge I. Vélez, and Eduardo Zurek. A distance-based kernel for classification via support vector machines. *Frontiers in Artificial Intelligence*, 7, 2024.
- [7] Pratik Waghmode, Manideep Kanumuri, Hosam El-Ocla, and Tanner Boyle. Intrusion detection system based on machine learning using least square support vector machine. *Scientific Reports*, 15, 2025.
- [8] Gayeong Eom and Haewon Byeon. Searching for optimal oversampling to process imbalanced data: Generative adversarial networks and synthetic minority over-sampling technique. *Mathematics*, 11:3605–3605, 2023.
- [9] Cataldo Guaragnella. Improving classification performance by addressing dataset imbalance: A case study for pest management, 2025.
- [10] Alberto Mozó, Amit Karamchandani, Luis de la Cal, Sandra Gómez-Canaval, Antonio Pastor, and Lluís Gifre. A machine-learning-based cyberattack detector for a cloud-based sdn controller. *Applied Sciences*, 13:4914–4914, 2023.
- [11] Mahdi Soltani, Khashayar Khajavi, Mahdi Jafari Siavoshani, and Amir Hossein Jahangir. A multi-agent adaptive deep learning framework for online intrusion detection. *Cybersecurity*, 7, 2024.
- [12] Niloofar Heidarikohol, Shuvalaxmi Dass, and Akbar Siami Namin. Evolutionary defense: Advancing moving target strategies with bio-inspired reinforcement learning to secure misconfigured software applications. 2025.
- [13] Tian Tian, Chen Zhang, Bo Jiang, Huamin Feng, and Zhigang Lu. Insider threat detection for specific threat scenarios. *Cybersecurity*, 8, 2025.
- [14] Pierpaolo Dini, Abdussalam Elhanashi, Andrea Begni, Sergio Saponara, Qinghe Zheng, and Kaouther Gasmi. Overview on intrusion detection systems design exploiting machine learning for networking cybersecurity. *Applied Sciences*, 13:7507–7507, 2023.
- [15] Tanwir Ahmad and Dragoş Truşcan. Efficient early anomaly detection of network security attacks using deep learning. pages 154–159, 2023.
- [16] Wa’ad H. Aljuaid and Sultan S. Alshamrani. A deep learning approach for intrusion detection systems in cloud computing environments. *Applied Sciences*, 14:5381–5381, 2024.
- [17] Lorenzo Diana, Pierpaolo Dini, and Davide Paolini. Overview on intrusion detection systems for computers networking security. *Computers*, 14:87–87, 2025.
- [18] K.A. Dhanya, Sulakshan Vajipayajula, Kartik Srinivasan, Anjali Tibrewal, Senthil Kumar Thangavel, and T. Gireesh Kumar. Detection of network attacks using machine learning and deep learning models. *Procedia Computer Science*, 218:57–66, 2023.
- [19] M Eraqi Hesham, Mohammed Essam, Mohamed Bahaa, Ahmed Mohamed, Mohamed Gomaa, Mena Hany, and Wael Elersy. Evaluating predictive models in cybersecurity: A comparative analysis of machine and

- deep learning techniques for threat detection. 45:33–38, 2024.
- [20] Md Liakat Ali, Kutub Thakur, Suzanna Schmeelk, Joan E. DeBello, and Denise Dragos. Deep learning vs. machine learning for intrusion detection in computer networks: A comparative study. *Applied Sciences*, 15:1903–1903, 2025.
- [21] Aymen Akremi. Design and analysis of an effective architecture for machine learning based intrusion detection systems, 2025.
- [22] S. Saif, Aqeef Alim Ansari, Suparna Biswas, and D. Giri. A comprehensive analysis of machine learning-based intrusion detection systems: evaluating datasets and algorithms for internet of things. *Journal of Cyber Security Technology*, 0(0):1–27, 2024.
- [23] G. Rohini, C. Gnana Kousalya, and J. Bino. Intrusion detection system with an ensemble learning and feature selection framework for iot networks. *IETE Journal of Research*, 69(12):8859–8875, 2023.
- [24] M. Alweshah, S. Alkhalailah, M. Beseiso, M. Almiyani, and S. Abdullah. Intrusion detection for iot based on a hybrid shuffled shepherd optimization algorithm. *The Journal of Supercomputing*, 78(10):12278–12309, 2022.
- [25] C. Zhou. Detection and classification of phishing websites using machine learning approach taking advantage of metaheuristic algorithms efficiency. *Journal of Cyber Security Technology*, 0(0):1–18, 2025.
- [26] Y. K. Saheed, Temitope Olubanjo Kehinde, Mustafa Ayobami Raji, and U. A. Baba. Feature selection in intrusion detection systems: a new hybrid fusion of bat algorithm and residue number system. *Journal of Information and Telecommunication*, 8(2):189–207, 2024.
- [27] R. Akula and G. S. N. Kumar. Optimizing intrusion detection systems using deep learning and genetic algorithms for network traffic analysis: A survey. In *Artificial Intelligence in Internet of Things (IoT): Key Digital Trends*, page 357–369. Springer Nature, 2025.
- [28] G. G. Gebremariam, J. Panda, and S. Indu. Design of advanced intrusion detection systems based on hybrid machine learning techniques in hierarchically wireless sensor networks. *Connection Science*, 35(1):2246703, 2023.
- [29] Y. K. Saheed and S. Misra. A voting gray wolf optimizer-based ensemble learning models for intrusion detection in the internet of things. *International Journal of Information Security*, 23(3):1557–1581, 2024.
- [30] D. Mehanović, D. Kečo, J. Kevrić, S. Jukić, A. Miljković, and Z. Mašetić. Feature selection using cloud-based parallel genetic algorithm for intrusion detection data classification. *Neural Computing and Applications*, 33(18):11861–11873, 2021.
- [31] S. Sivamohan, S. S. Sridhar, and S. Krishnaveni. Teakho-ids: An intrusion detection system for industrial cps with trustworthy explainable ai and enhanced krill herd optimization. *Peer-to-Peer Networking and Applications*, 16(4):1993–2021, 2023.
- [32] M. Ozkan-Okay, R. Samet, O. Aslan, S. Kosunalp, T. Iliev, and I. Stoyanov. A novel feature selection approach to classify intrusion attacks in network communications. *Applied Sciences*, 13(19):Article 19, 2023.
- [33] Y. Pourardebil Khah, M. Hosseini Shirvani, and H. Motameni. A hybrid machine learning approach for feature selection in designing intrusion detection systems (ids) model for distributed computing networks. *The Journal of Supercomputing*, 81(1):254, 2024.
- [34] B. Mallick. Enhancing cyber security: a comprehensive approach to the classification and prediction of significant cyber incidents (sci) through data mining and variational neural network with fox optimization algorithm. *International Journal of Computers and Applications*, 46(8):621–627, 2024.
- [35] D. K. K. Reddy, J. Nayak, H. S. Behera, V. Shanmuganathan, W. Viriyasitavat, and G. Dhiman. 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, 2024.
- [36] K. Thakur and G. Kumar. Nature inspired techniques and applications in intrusion detection systems: Recent progress and updated perspective. *Archives of Computational Methods in Engineering*, 28(4):2897–2919, 2021.
- [37] H. Kadry, A. Farouk, E. A. Zanyat, and O. Reyad. Intrusion detection model using optimized quantum neural network and elliptical curve cryptography for data security. *Alexandria Engineering Journal*, 71:491–500, 2023.
- [38] A. Shanbhag, S. Vincent, S. B. B. Gowda, O. P. Kumar, and S. A. J. Francis. Leveraging metaheuristics for feature selection with machine learning classification for malicious packet detection in computer networks. *IEEE Access*, 12:21745–21764, 2024.
- [39] P. K. Keserwani, Mahesh Chandra Govil, and E. S. Pilli. An optimal intrusion detection system using gwo-csadsae model. *Cyber-Physical Systems*, 7(4):197–220, 2021.
- [40] N. Savanović, A. Toskovic, A. Petrovic, M. Zivkovic, R. Damaševičius, L. Jovanovic, N. Bacanin, and B. Nikolic. Intrusion detection in healthcare 4.0 internet of things systems via metaheuristics optimized machine learning. *Sustainability*, 15(16):Article 16, 2023.
- [41] A. Naghib, F. S. Gharehchopogh, and A. Zamanifar. A comprehensive and systematic literature review on intrusion detection systems in the internet of medical things: current status, challenges, and opportunities. *Artificial Intelligence Review*, 58(4):114, 2025.

-
- [42] A. H. Ali, B. Ammar, M. Charfeddine, and B. B. Hamed. Enhanced intrusion detection based hybrid meta-heuristic feature selection. In *Advances in Computational Collective Intelligence*, page 3–15. Springer Nature Switzerland, 2024.
 - [43] I. H. Hassan, A. Mohammed, and M. A. Masama. Metaheuristic algorithms in network intrusion detection. In *Comprehensive Metaheuristics*, page 95–129. Academic Press, 2023.
 - [44] R. Ghanbarzadeh, A. Hosseinalipour, and A. Ghafari. A novel network intrusion detection method based on metaheuristic optimisation algorithms. *Journal of Ambient Intelligence and Humanized Computing*, 14(6):7575–7592, 2023.
 - [45] Vladimir Vapnik, Seymour Golowich, and Alex Smola. Support vector method for function approximation, regression estimation and signal processing. In *Advances in Neural Information Processing Systems*, volume 9, pages 281–287, 1996.
 - [46] Adele Cutler, David R. Cutler, and Joshua R. Stevens. Random forests. In *Machine Learning - ML*, volume 45, pages 157–176. Springer, 2011.
 - [47] Thomas Cover and Peter Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27, 1967.
 - [48] David D. Lewis. Naive (bayes) at forty: The independence assumption in information retrieval. In Claire Nédellec and Céline Rouveirol, editors, *Machine Learning: ECML-98*, pages 4–15. Springer, 1998.
 - [49] Lior Rokach and Oded Maimon. Decision trees. In Oded Maimon and Lior Rokach, editors, *Data Mining and Knowledge Discovery Handbook*, pages 165–192. Springer US, 2005.
 - [50] Ahmed A. Abdelhamid, Shaimaa K. Towfek, Negin Khodadadi, Abdullah A. Alhussan, Doaa S. Khafaga, Mohamed M. Eid, and Abeer Ibrahim. Waterwheel plant algorithm: A novel metaheuristic optimization method. *Processes*, 11(5):Article 5, 2023.
 - [51] Colin Reeves. Genetic algorithms. In *Handbook of Metaheuristics*, volume 146, pages 109–139. Springer, 2010.
 - [52] James Kennedy and Russell Eberhart. Particle swarm optimization. In *Proceedings of ICNN'95 - International Conference on Neural Networks*, volume 4, pages 1942–1948, 1995.
 - [53] Seyedali Mirjalili, Seyed Mohammad Mirjalili, and Andrew Lewis. Grey wolf optimizer. *Advances in Engineering Software*, 69:46–61, 2014.
 - [54] Noraini Johari, Aishah Zain, Noorhaniza Mustaffa, and Aini Udin. Firefly algorithm for optimization problem. *Applied Mechanics and Materials*, 421:512–517, 2013.
 - [55] Rainer Storn and Kenneth Price. Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11:341–359, 1997.
-