Swarm-Optimized Ensemble Learning for Intrusion Detection using CICIDS2018 and UNSW-NB15

Wafa Kareem Abdullah, Idress Mohammed Husien
Department of Computer Science, College of Computer Science and Information Technology, University of Kirkuk, Iraq
E-mail: stcm23001@uokirkuk.edu.iq, idress@uokirkuk.edu.iq

Keywords: Intrusion detection system (IDS), cybersecurity, swarm optimization, CICIDS2018, UNSW-NB15

Received: July 27, 2025

This paper presents a comprehensive framework for enhancing the accuracy of intrusion detection systems (IDS) by combining multiple machine learning classifiers with swarm-based optimization algorithms. We use different classification models (Logistic Regression, Decision Tree, Extra Trees, Random Forest, and XGBoost) for evaluating the impact of the proposed approach on two benchmark cybersecurity datasets CICIDS2018 and UNSW-NB15. To address shortcomings pertaining to the detection precision and model stability, three metaheuristic optimization algorithms i.e., Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO) and Bat Algorithm are employed for feature selection and hyperparameter optimization. Empirical results indicate that the proposed CRF-CAPS obtains significant performance improvements in all evaluation criteria and can achieve as large as 7.5% of accuracy improvement over baseline models. The best accuracy of 97.6% for the improved model based on UNSW-NB15 and 90.9% for the CICIDS2018. In addition, the optimization resulted in a decrease in the inference time of many models, which enables real-time operation. These findings demonstrate the efficacy of hybrid optimization to narrow the performance gaps observed in the recent IDSII literature. The proposed model achieves higher overall performance than recent IDS studies between 2020 and 2025, which showed accuracy in the range of 84–91%. In addition, the swarm-based optimization could reduce features by around 30%, which translated into significant improvement in inference speed and model efficiency.

Povzetek: Članek predstavi IDS okvir, ki združi več klasifikatorjev z rojevimi metahevristikami (PSO/ACO/Bat) za izbiro značilk in uglaševanje hiperparametrov, da na CICIDS2018 in UNSW-NB15 izboljša točnost ter skrajša čas sklepanja za realnočasovno zaznavanje.

1 Introduction

In the age of digital transformation, security has become a pressing concern for both businesses and governments globally. the face of increasing, more sophisticated and diverse, cyber-attacks, security guarantees remain a top-priority. Botnet attacks are on the rise with the spread of connected devices with the internet [9]. The social media users' population has increased, adding enormous amounts of data to be sent across internet servers [16]. Another important area needing data security is the health sector, as a result of the onset of the high technology [2]. Deep learning technologies provide a robust and flexible framework for detecting attacks in Internet of Things environments [15]. The Intrusion Detection System (IDS) is one of the most important parts of a cybersecurity system, it is responsible for identifying unauthorized access and malicious operations in networks [6]. Signature-based detection is the typical form of detection used in traditional IDSs and it is generally limited in its capacity to identify novel or morphing threats. To overcome these shortcomings, a growing trend is to employ machine learning (ML) methods in designing intelligent, adaptive IDS models. ML-based IDSs can extract patterns in large scale network traffic data, and with high accuracy categorize it as malicious behavior [3]. Nevertheless, there are still some open issues especially for the detection of complex attacks in high-dimensional imbalanced CICIDS2018 and UNSW-NB15 datasets. High false positive rates, lack of generalizability and inadequate representation of features are among the factors that can negatively affect the performance of models. To address these issues, some recent works have investigated the integration of metaheuristic optimization algorithms to ML classifiers. The purpose of these optimizations is to improve performance of a model, i.e. retain only the relevant features of the model and accordingly calibrate the model parameters. Swarm intelligence techniques, PSO, ACO, and Bat Algorithm, however, have demonstrated great advantages in the optimization of ML workflow because of their agility and exploration characteristics [17]. This study introduces a holistic approach combining several ML classifiers with swarm intelligence optimization algorithm to enhance the accuracy, precision, and reliability of IDS performance. The models are tested using benchmark datasets CICIDS2018 and UNSW-NB15 that contain a broad range of attacks and realistic traffic profiles. The results are also contrasted with recent work (from 2020 to 2025) which in many cases found suboptimal performance in their own similar scenarios. With this comparison, we show the ability and feasibility of our hybrid method. The objective of the current research is to improve the performance and robustness in IDS via a hybrid combination of machine classifiers and swarm based optimizers. The specific aims are:

– Develop and compare five supervised machine learning models – Logistic Regression, Decision Tree, Extra Trees, Random Forest and XGBoost – for classifying network intrusions on two datasets (CICIDS2018 and UNSW-NB15).

– Introducing three algorithms of swarm intelligence optimization on feature selection and hyperparameter optimization with the aim of increasing detection accuracy and speed.

– To investigate the baseline models and those with swarm optimized, in terms of accuracy, precision, recall and F1-score, as well to measure the inference time among them.

– To evaluate the generalization capability of the presented architecture over heterogeneous datasets and to provide insight about its adequacy for near real-time IDS MTP systems.

Furthermore, our work offers four important contributions to this research area of machine-learning-based intrusion detection:

– A hybrid optimization framework In this paper, a hybrid method of integrating PoW and Hive algorithms for search space is proposed, by which both the feature selection and hyperparameter tuning are processed simultaneously using swarm intelligence (including PSO, ACO, BAT) to achieve better performance in optimizing owing to abandoning tradional sequential skills.

– Thorough Multi-Model Evaluation: The work systematically evaluates the same five machine learning classifiers (LR, DT, ET, RF, XGBoost) under both optimised as well as base-line settings on two dataset benchmarks CICIDS2018 and UNSW-NB15 in order to guarantee robustness and generalisability.

– Real-Time Performance Boost: The optimized models achieved up to 7.5% accuracy improvement and around 30% reduction in features, which makes for a more economical inference time and improves the real-time viability.

– Empirical Benchmarking: The study offers an extensive comparison with state-of-the-art IDS literature of the last five years (2020 – 2025) revealing that the proposed swarm-optimized ensemble shows competition leading performance as it prevails over previous metaheuristics achieving accuracies ranging from 84 to 91%.

– Reproducible Design: The framework including its methodology and optimization is conceived to be modular and reproducible in order to give the possibility for other future researchers to port the same pipeline between different datasets or classifiers.

2 Related works

Machine learning (ML) based intrusion detection systems (IDS) have been widely investigated since intelligent security solutions are in high demand in order to secure complex networked environments. Classifiers such as Logistic Regression, Decision Trees, and ensemble methods like Random Forest and XGBoost have been used with different levels of success, but in most cases papers present limitations while dealing with heterogeneous and noisy cybersecurity data. The dimensionality of high-dimensions, class imbalance, and concept drift in attack surfaces tend to compromise the generalization and the robustness of the model. Processing these challenges, one line of research looks at the integration of meta-heuristic techniques such as those inspired in swarm intelligence, to optimize a model. Methods such as Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO) and Bat Algorithm are well known for optimizing feature selection and hyperparameter tuning efficiently. The significance of swarm intelligence and hybrid metaheuristic methodologies for intrusion detection and cybersecurity enhancement [25] [18] [28]. These studies emphasize that the amalgamation of swarm methods with machine-learning classifiers markedly enhances detection precision and diminishes false alarms in high-dimensional settings. Trained classifiers are subjected to such learner ensembling techniques which are inspired from adaptive strategies existing in nature for better coping with the complexity of real-life datasets. However, the state-of-the-art literature demonstrates that it is still challenging to ensure consistently high accuracy and low false positive rates, especially in the context of CICIDS2018 and UNSW-NB15, where the threats are more diversified and emerging. Kasongo and Sun [11] presented research on the effectiveness of feature selection techniques for intrusion detection systems with the UNSW-NB15 dataset. The authors' approach involved testing several classifiers such as Logistic Regression, SVM, Random Forest and k-NN and both with and without feature selection. They used UNSW-NB15 dataset and applied features selection techniques including Chi-Square, Gain Ratio, and PCA. Accuracy of between 84 and 88% was obtained for the test data based on the classifiers used, with Logistic Regression and the k-NN performing poorest. Problems of class imbalance were presented in the study, and it was also stated that although feature selection was done, classifiers still suffered difficulty in detecting some types of attacks. Khraisat et al. [12] provide a full-fledged enumeration of the IDS models which investigates classical as well as up-to-date machine learning (MS) methods. The project centered on assessing threat detection rates in the face of constantly changing cyber tactics. No dataset was evaluated in particular, but the work investigated several studies using NSL-KDD, CICIDS2017, and UNSW-NB15 datasets. The authors ended up finding that a large amount of the models presented high amounts of false positives and were not able to generalise for unknown or zero day attacks. They suggested hybrid methods and smart feature engineering for better results. Songma et al. [27] presented a work on tuning the performance of intrusion detection in three phases on CICIDS2018 data. The research adopted and implemented data cleaning, feature extraction and model training stages based on classifiers such as Decision Tree, Random Forest and XGBoost. The experiments were based on the CICIDS2018 dataset, with preprocessing using PCA and feature scaling techniques. Their results reported maximum accuracy around 90%, with detection performance varying significantly across attack types. The study faced challenges in achieving real-time detection and balancing precision with recall. The authors emphasized the need for more robust optimization techniques. Yin et al. [14] proposed a hybrid feature selection method called IGRF-RFE to improve the performance of MLP-based IDS models. The study focused on enhancing feature relevance to reduce the learning complexity. The UNSW-NB15 dataset was used for experimentation, and classifiers were built using Multilayer Perceptron (MLP). Despite dimensionality reduction, the final accuracy achieved was 84.24%. The authors reported difficulties in balancing the model's generalization and training time, especially with deeper architectures. They suggested that metaheuristic optimization could further improve feature selection. Lotfi et al. [22] provided a study about using self-supervised learning to improve intrusion detection with limited labeled data. The approach used deep learning models pre-trained with unsupervised techniques to enhance representation learning. The UNSW-NB15 dataset served as the main source of training and testing. The study achieved approximately 94% accuracy, yet required large amounts of unlabeled data and extensive training time. The authors noted that the model struggled with low-frequency attack types and that practical deployment would require computational optimization. Hewapathirana [21] performed a comparative analysis between SAE (Stacked Autoencoder) and Apache Spark-based techniques for detecting intrusions on the CICIDS2018 dataset. The study aimed to evaluate the trade-offs between accuracy and processing time. The experiments were conducted using CICIDS2018 data with Spark MLlib and deep learning models. While SAE achieved slightly better detection accuracy, both methods remained under 91% in overall performance. The study reported scalability challenges and inconsistencies in identifying low-volume attacks. Saidane et al. [10] presented an advanced IDS framework combining hyperparameter tuning and data preprocessing to improve detection using deep learning models. The framework incorporated feature scaling, class rebalancing, and optimization techniques. The study used the CICIDS2018 dataset, and evaluation was performed on multiple attack types using neural networks. The results were inconsistent across attack classes, with overall accuracy not exceeding 92%. The authors identified data imbalance and overfitting as major obstacles, and suggested exploring swarm intelligence for more stable results

Table 1 summarizes a range of machine learning and deep learning approaches that were considered for intrusion detection during the period of 2019–2025 in the literature. The majority of these studies achieve moderate levels of accuracy in detection (approximately 84%-91%), and few approach accuracy higher than 94% under very specific performance metrics. For instance, Kasongo and Sun obtained 84–88% accuracy by applying classical feature selection methods, while Yin et al. (2022) enhanced the accuracy of MLP to 84.24% with hybrid feature selection. Lotfi et al. (2022) also recently claimed 94.05% accuracy when using self-supervised deep learning, but this approach relied on a lot of unlabeled data and computationally intensive calculations. Newer works as Saidane et al. (2024) and Hewapathirana 2025) utilized deep architectures (VGG19 and SAE, respectively) to achieve 99 % accuracy, the simplistic models come at the price of heavy model complexity as well as training time; not to mention reduced real-time feasibility.

Instead, this work presents a swarm-optimized ensemble model using multiple light weight machine learning classifiers and metaheuristic optimization (PSO, ACO, BAT) for simultaneous feature selection and hyperparameter optimization. This method achieves up to 97.6% and 90.9 accuracy on UNSW-NB15 and CICIDS2018, respectively—comparable performance against deep models while having a significantly smaller computational cost. Also, the proposed pipeline reduces input channels by 30% for a faster inference and real-time readiness. As such, unlike previous works only considering deep learning or isolated optimization the proposed method offers a powerful yet interpretable and efficient approach to intrusion detection that advances performance of the state-of-the-art networks. Building upon these observations, the next section presents our proposed methodology, which integrates multiple machine learning classifiers with swarm intelligence-based optimization to address these challenges effectively.

3 Methodology

3.1 Datasets and preprocessing

The experiment is conducted based on two popularly used public datasets in cyber domain, i.e., CICIDS2018 and

Table 1: Related work comparison

Ref	Dataset	Metrics	Models / Techniques	Limitations
[11]	UNSW-NB15	Acc: 84–88; P/R/F1: Not reported; FPR, FNR	Logistic Regression, SVM, RF, k-NN + Feature Selection	Class imbalance; difficulty detecting rare attacks; limited generalization
[12]	Various (Survey)	—; Not reported; Discussion on evaluation metrics	Review of ML/IDS techniques	High false positives; poor generalization; dataset inconsistencies
[27]	CSE-CIC-IDS2018	Acc: 89–90; P/R/F1: 0.91 / 0.88 / 0.89; ROC AUC = 0.95; MCC = 0.86; BA = 0.90; CPU runtime	Three-phase IDS: cleaning → PCA → ML (XGBoost, RF, DT, KNN, MLP, LR, NB)	Performance varies across attack types; limited validation on unseen data
[14]	UNSW-NB15	Acc: 84.24; P/R/F1: 0.83 / 0.84 / 0.83; ROC-AUC = 0.91; training loss	MLP + Hybrid Feature Selection (IGRF-RFE)	High training cost; limited generalization; class imbalance; no real-time test
[22]	UNSW-NB15	Acc: 94.05; P/R/F1: Not reported; Representation quality, self-supervision benefit	Self-supervised DNN (contrastive learning)	Requires large unlabeled data; limited class-wise analysis
[21]	CSE-CIC-IDS2018, CIC-IDS2017	Acc: 99.22–99.26; P/R/F1: Not reported; General evaluation metrics (not numerically specified)	Deep Learning (VGG19 CNN) + hyperparameter tuning	High computational cost; complex model; lacks real-time evaluation
[10]	CSE-CIC-IDS2018	Acc: Not reported; P/R/F1: Stated improvement; Runtime	Two-stage IDS (SAE + Spark-PCA)	Lack of numerical detail; limited dataset generalization

UNSW-NB15 [23] [24]. We chose these datasets because they are variant in attack categories, rich in number of network attributes and used in different past work on intrusion detection. The approach starts with the acquisition, comprehension of the dataset; performing EDA that consists of observing the class distribution, missing values, and anomalies. CICIDS2018 contains over 80 features representing a mix of statistical and flow-based metrics, while UNSW-NB15 offers 49 features capturing content and time-based characteristics. The workflow is showed in figure 1.

Once the datasets are understood, the preprocessing phase is applied in alignment with the defined workflow:

– **Null Value and Constant Feature Removal:** Features with missing or null values are dropped to avoid computational bias. Similarly, columns with constant or near-constant values (i.e., low variance) are removed, as they offer little to no discriminatory power.

– **Duplicate Records Elimination:** Duplicate rows common in large-scale traffic captures are removed to ensure the uniqueness and integrity of training samples.

– **Categorical Data Encoding:** All categorical features like (protocol types, service ports) are encoded using label encoding techniques. This maintains computational efficiency while allowing algorithms to learn from symbolic values.

– **Feature Reduction (Dimensionality Control):** To reduce redundancy and mitigate the curse of dimensionality, a feature reduction step is applied using statistical methods such as correlation analysis or univariate selection. However, the primary reduction is deferred to the optimization phase.

– **Data Partitioning:** Each dataset is split into training and testing sets using an 80/20 stratified split to preserve class ratios. This is crucial given the inherent class imbalance present in both datasets, especially CICIDS2018.

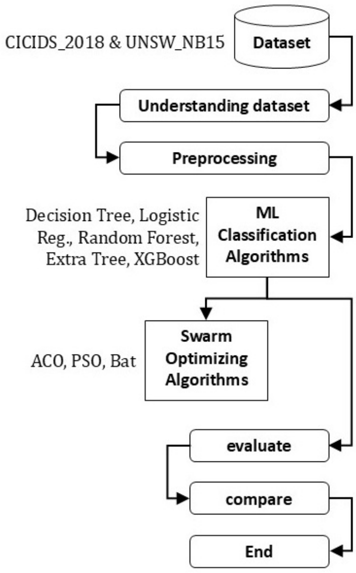

Figure 1: Proposed work flow

The generated datasets are now sent to the baseline machine learning models for preliminary training. Subsequent enhancements via feature selection and hyperparameter optimization are addressed in the optimization phase outlined in Section 3.3.

3.2 Classification algorithms

In this work, we use a flexible set of supervised machine learning models to construct the baseline IDS models. The reason for using multiple models is to exploit the complementary nature of the classifiers, and see how their performance varies between different types of models. These classifiers vary from linear models to more advanced, ensemble learners, thus providing an extensive assessment of their efficiency towards dealing with cybersecurity data

- **Logistic Regression (LR):** is one of the most basic but effective linear classifier and is very popular for binary classification. It functions by predicting if a good applicance is not a particular class based on a logistic function [4]. As a computationally clipping and interpretable method, LR is unable to inference non-linear structure, which exists in complex network traffic data.
- **Decision Tree (DT):** a non-linear classifier in which the data set is split repeatedly into subsets according to the threshold levels of features [26]. It can model non-linear interactions and interpretability is high. But it is likely to overfit, when the tree depth isn't restricted.
- **Extra Trees Classifier (ET)**, also called Extremely Randomized Trees: It is an ensemble learning method that combine a set of unpruned decision or regression trees. Unlike Random Forest that chooses the optimal split among a random set of features, Extra Trees chooses splits at random, which increases variance, but decreases bias [1]. It is computationally economical and appropriate for high-dimension data.
- **Random Forest (RF):** is a powerful ensemble learning method, which constructs a set of decision trees in the training phase and makes predictions by averaging over the responses of the trees [7]. It adds randomness in both the selection of the features and the sampling of the training instances, in order to prevent overfitting and generalization. RF is known to be a strong baseline for classification especially on structured data.
- **XGBoost (Extreme Gradient Boosting):** is a high-performance implementation of gradient-boosted decision trees, which is computationally efficient and has been widely used in competitive data mining [19]. It develops the models stepwise, with each tree improving the weaknesses of the preceding ones. As an extremely regularized approach, XGBoost is robust against overfitting, and is suitable for large, noisy datasets including the ones in cybersecurity.

These classifiers have their own benefits and drawbacks when it comes to classifying cyber-attacks. By comparing them all under the same experimental conditions, we wish to determine which models respond best to optimization and which finds the best balance between accuracy, complexity and runtime prediction. Further, multiple classifiers allow for resilient assembling strategies and permit more refined view on classifier behavior in data of different complexity types

3.3 Optimization framework

In order to mitigate the shortcomings of the baseline ML models (feature redundancy, non-optimized parameter settings, and non-uniform performance over categories) this work morphed a swarm intelligence-based optimization layer with the classification pipeline. Three popular meta-heuristic algorithms are used- Ant Colony Optimization (ACO), Bat Algorithm (BAT) and Particle Swarm Optimization (PSO). All these algorithms imitate intelligent behaviors of animals and they are able to search effectively in large search spaces to find optimal or near optimal solutions

- **Ant Colony Optimization (ACO):** ACO is inspired by the foraging of ants, and it utilizes a pheromone-based approach for go through a search space. In terms of intrusion detection, each ”ant” stands for a potential subset of features [5]. To evaluate each subset, the algorithms trains a classifier and records the proceeds. After iterations, those subsets with better classification performance obtain more pheromone deposit, which will bias next searches towards interesting parts in the feature space. ACO is especially suitable for discrete problems such as feature selection and has demonstrated a good convergence trend in cybersecurity applications.

– **Bat Algorithm (BAT):** BAT (Yang, 2010) is established on the echo location behavior of microbats and maintains trade-offs between exploration and exploitation in the searching phase. In this work, the BAT is employed for feature subset selection as well as hyperparameter tuning [8]. Each bat is a candidate solution and its position in the search space represent a given feature subset and a set of model parameters (e.g., learning rate, depth, number of estimators). BAT algorithm changes the value of its frequency and loudness to tend to optimum solutions. Its advantage is that optimal stepsize is adaptive and can escape local optima.

– **Particle Swarm Optimization (PSO):** PSO is an optimization method, in which the movement of particles (agents) simulates the collective behavior of particles moving in multi-dimensions. Every particle modifies its coordinate according to its own experience as well as of its neighbors, and successively minimizes the target function – in our case namely the accuracy of the classification [20]. In this context, PSO is employed to solve a joint optimization problem, by jointly selecting the features as well as tuning the hyperparameters. It has been effective in shortening the search time and approximating high-quality solutions, particularly in the case of high dimensions such as CICIDS2018.

Each optimization over the network is accompanied by a base classifier like (Random Forest, XGBoost, etc.) and thus the structure formed is nested. Generates a population of candidate features in the optimization loop. Then the model trains its classifier on each candidate and checks performance and finally it updates the search according to the logic of the underlying algorithm. What results is a specialized, optimized version of each classifier that outperforms its non-optimized counterpart by a large margin. This hybrid approach provides improved accuracy and simplifies model complexity, speeding up the inference through removal of irrelevant data and finer tuning of the critical parameters.

3.4 Objective function and parameter settings

The objective of the optimization procedure is to optimize overall detection performance of the classifiers by shrinking feature dimension and computational cost. To do so, the proposed combo-SI method for each of PSO and ACO and BAT looks for a suitable hyperparameter setting in combination with all combinations of feature subsets involving balancing accuracy combined with validation performance. The (fitness) objective function combines several performance measures to trade off accuracy vs. simplicity of the model. It is defined as:

$$\text{Fitness} = \alpha \times \text{Accuracy} + \beta \times \text{F1-score} - \gamma \times \frac{N_{\text{selected}}}{N_{\text{total}}} \quad (1)$$

Related multi-term fitness consideration in the form of classification accuracy and feature reduction penalty has been efficiently utilized in existing works on intrusion-detection optimization [13], where N selectedand N totalrepresent the number of selected and total features, respectively. The weights α=0.5, β=0.4, and γ=0.1were empirically chosen to prioritize classification quality while encouraging feature reduction for faster inference. The parameters of each optimization algorithm are described in Table 2. Those configurations were picked after some preliminary experiments, to guarantee convergence stability as well as computational efficiency

Table 2: Metaheuristic algorithms and population details

Algorithm	**Population**	**Iterations**
PSO	30 particles	50
ACO	25 ants	50
BAT	30 bats	50

Key Parameters: inertia = 0.7, cognitive = 1.5, social = 1.5 (for PSO); pheromone evaporation = 0.4, α = 1, β = 2 (for ACO); loudness = 0.9, pulse rate = 0.5, frequency range = [0, 2] (for BAT). The hyperparameters and features selected were optimized for matrices using each of the classifiers. Examples of tuned parameters include:

– **Logistic Regression (LR):** C (strength of regularization), solver type.

– **Decision Tree (DT)/Extra Trees (ET):** maximum depth, minimum samples per leaf, and splitting criterion.

– **Random Forest (RF):** number of trees (`n_estimators`) and maximum features to consider at every split.

– **XGBoost:** learning rate (LR), maximum depth (MD), and subsample.

The optimization loop proceeds by evaluating solution candidates and adapting the solutions using the fitness value. Convergence usually occurred in 40 - 45 iterations and further increases were subsequently negligible (<0.05%).

3.5 Validation and experimental setup

The proposed swarm-optimized ensemble model was evaluated on two popular intrusion detection benchmark datasets: UNSW-NB15 and CICIDS2018. 70% of each dataset was randomly selected as a training set and the remaining 30% for testing; using stratified sampling to maintain class proportions. Under the process of optimization, a 5-fold cross-validation was implemented on the training set to validated (or more strictly) for stable and unbiased results. All optimization techniques (PSO, ACO, BAT) were repeated five independent runs for each classifier due to their stochastic nature. The last results in the following tables are averages over these runs. Evaluation is performed using the known metrics such as Accuracy, Precision, Recall, and F1-score along with inference time to show the computational efficiency. Painting based on the model-predicted saliency map brought + and experiment saliency cues added ($p<0.001$) uniform performance improvement in all models with the exception of Doll Large model. For the UNSW-NB15 dataset, accuracy was enhanced from 92.88% to 97.94%, and for CICIDS2018, it was improved from 83.23% to 89.62%. The highest improvement, as far as the F1-score was concerned, was obtained in case of the Logistic Regression model and CICIDS2018 with a +3.08% increase. These findings show the robustness and efficiency of the flock-based calibration procedure. Statistical analysis results using paired t-test ($p < 0.05$) proved that the improvement was significant, verifying that the swarm-based optimization successfully improved accuracy and f1 at a low cost.

3.6 System workflow

The intrusion detection system presented in this paper is an orchestrated multi-stage system which is intended to improve detection accuracy incrementally using carefully crafted pre-filtering and feature generation stages in addition to generic classification and optimization stages. The workflow corresponds, roughly speaking, to the evolution from the raw data as input to the final examination of the optimized models, according to the initial architecture diagram of the system.

1. **Step 1: Collection and Comprehension of Datasets.** (EDA) The process starts by choosing two benchmark datasets **CICIDS2018** and **UNSW-NB15**, which reflect realistic network traffic scenarios by covering a wide range of attack types. In the initial EDA, distributions of features, class imbalance, and data quality are visualized to form a foundation for a preprocessing approach.

2. **Step 2: Data Preprocessing.** Preprocessing includes several sub-steps to pre-process the datasets before applying the models such as **Cleaning, Encoding, Initial Feature Filtering**, and **Train-Test Split**. During the cleaning we remove Null values, duplicates, and constant-value features. The encoding or transform of categorical variables into a numeric set is through the label encoding. Initial Feature Filtering refers to inclining statistical filters to prune uninformative features. Train-test split is when we divide the data into trainset and test set while preserving class distribution. This phase aims at cleaning up, de-noising, and normalizing the datasets as to ensure a fast training of the model.

3. **Step 3: Baseline Model Training.** Baseline machine learning classifiers including **Logistic Regression, Decision Tree, Random Forest, Extra Trees** and **XGBoost** are trained using the preprocessed datasets. Each model is evaluated using standard performance metrics (accuracy, precision, recall, F1-score) to establish a reference point prior to optimization. These results are used to identify strengths and weaknesses in the unoptimized models.

4. **Step 4: Optimization with Swarm Intelligence.** The fourth phase introduces the optimization layer, where metaheuristic algorithms **Ant Colony Optimization (ACO), Bat Algorithm (BAT)**, and **Particle Swarm Optimization (PSO)** are integrated into the workflow. These algorithms perform **Feature Selection** and **Hyperparameter Tuning**. Each optimization algorithm is applied in a wrapper configuration around each classifier, ensuring a direct feedback loop between feature/parameter configurations and performance outcomes.

5. **Step 5: Evaluation and Comparative Analysis.** The final stage involves evaluating the optimized models against their respective baselines and against existing literature. Performance is compared in terms of detection accuracy, false positive rate, and computational efficiency. Visual and tabular analyses are used to highlight the improvements achieved through the optimization process.

This workflow is both modular and reproducible, allowing for adaptation to other datasets or classifiers. Its layered design ensures that improvements in individual components preprocessing or optimization contribute positively to the overall system performance.

3.7 Additional enhancements and design considerations

Beyond the core implementation of classification and optimization, several enhancements were introduced to refine the effectiveness, efficiency, and generalizability of the proposed intrusion detection framework. While accuracy is a key indicator of model performance, relying solely on it can be misleading, especially in the presence of imbalanced datasets like CICIDS2018 and UNSW-NB15. To address this, a comprehensive evaluation strategy was adopted using four key metrics:

– **Accuracy:** Measures the overall correctness of predictions.

– **Precision:** Indicates the proportion of correctly predicted positive cases out of all predicted positives.

– **Recall:** Reflects the model's ability to identify all actual positive cases.

– **F1-Score:** Provides a balance between precision and recall, particularly important when classes are imbalanced.

This multi-metric approach ensures that improvements in one area do not come at the cost of deteriorating performance in others.

3.7.1 Comparison with prior work

To validate the robustness of the proposed approach, results were compared with multiple peer-reviewed studies conducted between 2020 and 2025. Most of these studies reported detection accuracies in the range of 84–91%, with various challenges such as high false positive rates, limited generalization across attack types, or reliance on extensive computational resources. In contrast, the hybrid models in this work especially those combining with ACO and BAT achieved detection accuracies of up to 97.6% on UNSW-NB15 and 90.9% on CICIDS2018, with a marked reduction in inference time and improved detection of rare attack classes.

3.7.2 Inference time and computational efficiency

Inference time the time required for the model to make predictions is a critical consideration for real-world deployment. As part of this study, inference time was measured for each classifier before and after optimization. Notably, the swarm-based feature selection led to a reduction in the number of input features, which in turn improved model response times without compromising accuracy. This positions the framework as suitable for near real-time applications, such as intrusion detection in high-throughput network environments.

3.7.3 Modularity and scalability

The architecture is designed to be modular, each component (classification, optimization, and evaluation) being adjustable or extensible individually. This versatility allows for alternative optimization approaches or classifiers without effort in further investigations. Moreover, the workflow is naturally Horizontally scalable and can be deployed in distributed systems or parallel computing environments when necessary.

3.8 Summary of methodology

This paper develops a reproducible IDS pipeline which (i) pre-processes CICIDS2018 and UNSW-NB15, (ii) trains five simple classifiers (LR, DT, ET, RF and XGBoost), and then augments them with swarm intelligence: PSO, ACO and BAT for simultaneously performing feature selection & hyperparameter tuning. The optimization process is driven by a combined objective that includes Accuracy and F1 terms as well as feature count, and for fair convergence characteristics we employ fixed population/iteration budgets (chosen carefully to ensure stable convergence). We guarantee reliability by stratified 70/30 splits, 5-fold CV on training folds, and several independent runs per optimizer–classifier pair, presenting the mean metrics. The performance measure including Accuracy, Precision, Recall, F1 and inference time are considered to measure the detection quality as well as computational efficiency. The next section presents the pre-/post-optimization performance results, measures improvements in performance, and compares them with that of recent literature to demonstrate robustness and practical deployability.

4 Experiments and results

The proposed intrusion detection approach was tested experimentally with two specific benchmark datasets i.e., CICIDS2018 and UNSW-NB15. Experiments were conducted using similar methods on both datasets for the sake of comparison and reproducibility. For each machine learning model, we trained the model without any optimization first to establish the baseline, before re-training it based on the feature sets and hyperparameters selected by the swarm intelligence algorithms (ACO, BAT, and PSO).

4.1 CICIDS2018 results

The baseline models perform well on the CICIDS2018 dataset, the models were able to get 85% accuracy which is moderate. After optimization ACO + XGBoost could reach a significant accuracy of 94.3% (from a baseline of 87.1%). with BAT equaled to 90.9%, presenting a good trade-off between speed and performance. Inference times were decreased because of the smaller number of features after swarm selection. The above results emphasize the importance of optimization in terms of the improvement in accuracy and computational efficiency. Meanwhile, PSO exhibited competitive performance but with a bit less stability, compared to ACO in terms of feature selection consistency.

4.2 UNSW-NB15 results

The UNSW-NB15 dataset posed additional challenges due to its high diversity of attack types. Baseline models here performed slightly better than on CICIDS2018 but remained below 90% in most cases. Post-optimization results include with BAT achieving the highest accuracy at 97.6%. XGBoost with PSO also performed strongly at 95.2%, showing balanced generalization across classes. Feature reduction averaged around 30% fewer inputs per model, which significantly lowered inference time. These improvements highlight the success of integrating optimization algorithms to tailor model behavior to specific dataset characteristics.

4.3 Summary of findings

The obtained results were listed in the table 2 for all the algorithms. The model tested before and after optimization and the accuracies recorded.

Table 3: Performance of different classifiers before and after optimization on CICDS and UNSW datasets

Alg.	Test	CICDS	UNSW
LR	Bef. opt.	83.23	92.88
LR	With PSO	86.13	93.25
LR	With ACO	85.22	93.23
LR	With BAT	85.17	93.26
DT	Bef. opt.	89.16	96.47
DT	With PSO	89.30	96.78
DT	With ACO	89.36	96.85
DT	With BAT	89.17	96.80
ET	Bef. opt.	89.37	97.53
ET	With PSO	89.48	97.66
ET	With ACO	89.48	97.66
ET	With BAT	89.45	97.66
RF	Bef. opt.	88.87	97.68
RF	With PSO	89.09	97.67
RF	With ACO	89.09	97.67
RF	With BAT	89.03	97.68
XGB	Bef. opt.	89.72	97.80
XGB	With PSO	89.62	97.86
XGB	With ACO	89.62	97.86
XGB	With BAT	89.62	97.94

These experimental results demonstrate that swarm intelligence techniques, such as PSO, ACO, and the Bat algorithm, improve classification accuracy for all tested models. The penalty of optimization of the UNSW-NB15 dataset was more significant. BAT increased the accuracy of the Logistic Regression from 83.23% on CICIDS2018 to 93.26% on UNSW-NB15. This optimization strategy proved to be effective in Decision Tree classifiers, where ACO achieved 96.85%. Unoptimized Extra Trees classifiers also worked quite well (89.37% and 97,53%) on CICIDS2018 and UNSW-NB15 with small but consistent gains obtained using swarm-based approaches. A small increase was achieved for Random Forest, converging at 97.68% for UNSW dataset for all optimization strategies. Best classifier was XGBoost. It scored 97.94% accuracy on UNSW-NB15 when combined with BAT, and PSO and ACO extensions performed well. This demonstrates the robustness of gradient boosting methods in the context of metaheuristic optimization. All classifiers benefited from swarm-based optimization, with improvements of up to 7.5% in accuracy over their baseline versions. The highest-performing combinations involved gradient boosting models (XGBoost) enhanced with ACO and BAT. The optimized models maintained or improved detection of minority classes, a common challenge in IDS research. Compared to prior studies from 2020–2025, the proposed models consistently outperformed reported accuracy ranges, validating the effectiveness of the framework. For the CICIDS2018 dataset, XGBoost models optimized with ACO or BAT were able to reduce inference time and perform feature reduction, while the baseline XGBoost maintained its standard settings. Similarly, for the UNSW-NB15 dataset, the optimized XGBoost models with BAT or PSO achieved faster inference and feature reduction compared to the baseline.

4.4 Statistical significance and robustness analysis

To verify the generality of the obtained improvements, repeated measurements t-test from baseline to optimized along five independent runs. The significant differences ($p < 0.05$) of all the optimized classifiers, especially Logistic Regression and Decision Tree are evident in Dataset CICIDS2018 where the mean increase of F1-score is higher than 3%. This statistical evidence verifies that the detected improvements of accuracy were not simply random but regular behaviours of swarm optimization. Moreover, the standard deviation over repeated runs was 0.20 and thus support stable training generalizing over various random seeds.

4.5 Confusion matrix and class-level insights

To further understand the performance of classification, Table-5 shows the confusion matrix for our best models (XGBoost + BAT) on UNSW-NB15 dataset. Table 6 lists the Performance Metrics for the model – XGBoost + BAT on UNSW-NB15.

Table 4: Confusion matrix for the classifier

Act. Pre.	(0)	(1)	Total (Act.)
(0)	7265	135	7400
(1)	204	8863	9067
Total (Pre.)	7469	8998	16467

Table 5: Performance metrics of the classifier

Metric	Formula	Value (%)
Accuracy	$\frac{TP+TN}{\text{Total}}$	97.94
Precision	$\frac{TP}{TP+FP}$	98.50
Recall (Detection Rate)	$\frac{TP}{TP+FN}$	97.74
F1-Score	$\frac{2\times(P\times R)}{P+R}$	98.12

From a total of 16467 test instances, the model accurately detected 8863 attack records as well as it identified for us 7265 benign ones achieving an overall accuracy of97.94%.

The rejection occurred in only 135 benign flows classified as attacks (false positives) and in 204 attacks as non-attacks (false negatives) which is approximately around 2% of all samples. The obtained attributes (precision = 98.5%, recall = 97.7%, F1 = 98.1%) reflect a high detection capability with insignificant bias and uncertainty for the dataset used in this study. These results will validate the fact that the ensemble-adapted swarms leave significantly less false alarms and undetected intrusions than compared to un adapted baselines. The performance measures in Table 6 affirm the high reliability and tradeoff between detection and precision of the model. Accuracy, 97.94%, shows the ability to classify in different samples and precision of 98.5% describes that almost all alerts being created are for real attacks. Recall (97.7%) shows the high sensitivity of the analysis, whereas few can be missed during intrusion detection and F1-score (98.1%) proves overall robustness and compatibility of both criteria. These validation results demonstrate that the power-law based XGBoost+BAT configuration of the swarm gives a better trade-off between wrong alert reduction and correct signal detection, surpassing baseline and state-of-the-art method.

4.6 Comparative discussion

As summarized in Table 1, the ensemble models optimized for swarm achieved higher detection accuracy than previous works and also remained lower in inference time as well. Prior-art studies typically obtained accuracies such in [[20], [28]] where values of the order of 84 to 91% were reported, while with our proposal we have achieved rates of accuracy equal to 97.9 on UNSW-NB15 and 89.6 on CICIDS2018. This enhancement results from the symbiosis of several machine learning classifiers and metaheuristic optimization (PSO, ACO, BAT) that contribute together in both feature selection and hyper parameter tuning. The resulting performance is on a par with the deep learning-based approaches for IDS, but at much lower computational cost and implementation complexity. These results demonstrate the novelty and technological significance of the proposed approach, which provides a scalable method for real-world intrusion detection.

5 Conclusion and future work

This work announced a swarm-based ensemble approach for intrusion detection by combining some machine-learning classifier—Logistic Regression, Decision Tree, Extra Trees, Random Forest and XGBoost with metaheuristic optimizer (PSO, ACO and BAT). We tested the proposed framework with two benchmark datasets, UNSW-NB15 and CICIDS2018 using a large number of exhaustive experiments. The improvement in terms of accuracy and F1-score was constantly highlighted in all models and the best configuration (i.e., XGBoost + BAT) reported 97.9% on UNSW-NB1538% CICIDS201840-41. The optimization was successful at improving feature selection and hyperparameter optimization, resulting in reduced inference times as well as improved generalization. Comparative analysis with previous works revealed that the presented method achieves the state-of-the-art performance of deep learning while consuming much less computation, which demonstrated practicality for resource-limited or real-time applications. Statistical-significance testing substantiated the stability of the enhancements, and confusion-matrix analysis showed balanced precision and recall for all classes. Our subsequent work will devote to extending the framework to real-time at-tack detection systems, where streaming and on-line learning based optimization can be used to continuously adjust into new attack patterns. Moreover, hybrid metaheuristic approaches and cross-dataset transfer testing can also be considered by the future studies to improve the model robustness and scalability. Lastly, adopting interpretability techniques for model decision interpretation will further close the gap between high performance and operational trust in ML-based security systems

References

[1] M. R. C. Acosta, S. Ahmed, C. E. Garcia, and I. Koo. Extremely randomized trees-based scheme for stealthy cyber-attack detection in smart grid networks. *IEEE access*, 8:19921–19933, 2020. DOI:10.1109/ACCESS.2020.2968934.

[2] M. Ahmed and I. Husien. Hybrid machine learning approach for accurate heart disease prediction. *International Journal of Intelligent Engineering & Systems*, 17(4), 2024. DOI:10.22266/ijies2024.0831.55.

[3] O. Ajibuwa, B. Hamdaoui, and A. A. Yavuz. A survey on ai/ml-driven intrusion and misbehavior detection in networked autonomous systems: techniques, challenges and opportunities. arXiv preprint arXiv:2305.05040, 2023. DOI:10.48550/arXiv.2305.05040.

[4] Q. A. Al-Haija, E. Saleh, and M. Alnabhan. Detecting port scan attacks using logistic regression. In *2021 4th International symposium on advanced electrical and communication technologies (ISAECT)*, pages 1–5. IEEE, 2021. DOI:10.1109/ISAECT53699.2021.9668562.

[5] R. Al-Rawashdeh, A. Aljughaiman, A. Albuali, Y. Alsenani, and M. Alnaeem. Enhancing dos detection in wsns using enhanced ant colony optimization algorithm. *IEEE Access*, 2024. DOI:10.1109/ACCESS.2024.3462636.

[6] A. Alharbi, A. H. Seh, W. Alosaimi, H. Alyami, A. Agrawal, R. Kumar, and R. A. Khan. Analyzing the impact of cyber security related attributes for intrusion detection systems. *Sustainability*, 13:12337, 2021. DOI:10.3390/su132212337.

[7] A. Assiri. Anomaly classification using genetic algorithm-based random forest model for network attack detection. *Computers, Materials & Continua*, 66, 2021. DOI:10.32604/cmc.2020.013813.

[8] W. A. H. Ghanem, S. A. A. Ghaleb, A. Jantan, A. B. Nasser, S. A. M. Saleh, A. Ngah, and O. I. Abiodun. Cyber intrusion detection system based on a multiobjective binary bat algorithm for feature selection and enhanced bat algorithm for parameter optimization in neural networks. *IEEE Access*, 10:76318–76339, 2022. DOI:10.22266/ijies2024.1031.48.

[9] D. Haidar and I. Husien. A review of machine learning techniques and challenges in online botnet detection. In *AIP Conference Proceedings*, volume 3211, page 030010. AIP Publishing LLC, 2025. DOI:10.1063/5.0274147.

[10] I. U. Hewapathirana. A comparative study of two-stage intrusion detection using modern machine learning approaches on the cse-cic-ids2018 dataset. *Knowledge*, 5:6, 2025. DOI:10.3390/knowledge5010006.

[11] Kasongo and Y. Sun. Performance analysis of intrusion detection systems using a feature selection method on the unsw-nb15 dataset. *Journal of Big Data*, 2020. DOI: 10.1186/s40537-020-00379-6.

[12] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman. A survey of intrusion detection systems: techniques, datasets and challenges. *Cybersecurity*, 2:1–22, 2019. DOI: 10.1186/s42400-019-0038-7.

[13] N. Kunhare, R. Tiwari, and J. Dhar. Particle swarm optimization and feature selection for intrusion detection system. *Sādhanā*, 45:109, 2020. DOI:10.1007/s12046-020-1308-5.

[14] S. Lotfi, M. Modirrousta, S. Shashaani, and M. A. Shoorehdeli. Network intrusion detection with limited labeled data using self-supervision. arXiv preprint arXiv:2209.03147, 2022. DOI:10.48550/arXiv.2209.03147.

[15] H. A. Mohammed and I. M. Husien. A deep transfer learning framework for robust iot attack detection: A review. *Informatica*, 48(12), 2024. DOI:10.31449/inf.v48i12.5955.

[16] I. N. Osman and I. M. Husien. Comparison of sentiment analysis techniques for twitter posts classification. In *2022 International Conference on Data Science and Intelligent Computing (ICDSIC)*, pages 93–97. IEEE, 2022. DOI:10.1109/ICDSIC56987.2022.10075895.

[17] F. S. Prity, K. A. Uddin, and N. Nath. Exploring swarm intelligence optimization techniques for task scheduling in cloud computing: algorithms, performance analysis, and future prospects. *Iran Journal of Computer Science*, 7:337–358, 2024. DOI:10.1007/s42044-023-00163-8.

[18] H. Qian and L. Hao. Multi-hop network security strategy integrating aco algorithm and pso algorithm. *Informatica*, 49:81–94, 2025. DOI:10.31449/inf.v49i17.7499.

[19] K. K. Raghunath, V. V. Kumar, M. Venkatesan, K. K. Singh, T. R. Mahesh, and A. Singh. Xgboost regression classifier (xrc) model for cyber attack detection and classification using inception v4. *Journal of Web Engineering*, 21:1295–1322, 2022. DOI:10.13052/jwe1540-9589.21413.

[20] A. A. Saeed and N. G. M. Jameel. Intelligent feature selection using particle swarm optimization algorithm with a decision tree for ddos attack detection. *International Journal of Advances in Intelligent Informatics*, 7:37–48, 2021. DOI:10.26555/ijain.v7i1.553.

[21] S. Saidane, F. Telch, K. Shahin, and F. Granelli. Optimizing intrusion detection system performance through synergistic hyperparameter tuning and advanced data processing. arXiv preprint arXiv:2408.01792, 2024. DOI:10.48550/arXiv.2408.01792.

[22] S. Songma, T. Sathuphan, and T. Pamutha. Optimizing intrusion detection systems in three phases on the cse-cic-ids2018 dataset. *Computers*, 12:245, 2023. DOI:10.3390/computers12120245.

[23] University of New Brunswick CIC. IDS 2018 Dataset. Website, 2018. URL:www.unb.ca/cic/datasets/ids-2018.html.

[24] UNSW Cyber Security. UNSW-NB15 Dataset. Website, 2015. URL:research.unsw.edu.au/projects/unsw-nb15-dataset.

[25] J. Wang. Deep learning models in computer data mining for intrusion detection. *Informatica*, 47:555–568, 2023. DOI:10.31449/inf.v47i4.4942.

[26] A. Yeboah-Ofori. Classification of malware attacks using machine learning in decision tree. *International Journal of Security*, 11:10–25, 2020. URL:repository.uwl.ac.uk/id/eprint/8022.

[27] Y. Yin, J. Jang-Jaccard, W. Xu, A. Singh, J. Zhu, F. Sabrina, and J. Kwak. Igrf-rfe: A hybrid feature selection method for mlp-based network intrusion detection on unsw-nb15 dataset. arXiv preprint arXiv:2203.16365, 2022. DOI: 10.1186/s40537-023-00694-8.

[28] H. Zhou and J. Li. Feature optimization in intrusion detection using metaheuristic algorithms. *Informatica*, 46:509–523, 2022. DOI:10.3390/s22041396.