Management Strategies and Engineering Sciences 2027; 9(4):1-15

Improving Intrusion Detection Systems Using the Human Evolutionary Optimization Algorithm

Mohammad Koochaki [1], Amir Sahafi [2*]

[1] MA, Department of Secure Telecommunications and Cryptography, ST.C., Islamic Azad University, Tehran, Iran
[2] Assistant Professor, Department of Computer Engineering, ST.C., Islamic Azad University, Tehran, Iran

*** Corresponding author email address**: sahafi@iau.ac.ir

Received: 2026-02-01 **Revised:** 2026-06-21 **Accepted:** 2026-06-28 **Initial Publish:** 2026-08-15 **Final Publish:** 2027-07-01

Abstract

An intrusion detection system is a critical tool in network security that identifies potential threats by analyzing behavioral patterns and network traffic. Despite recent advances, challenges such as imbalanced data, high feature dimensionality, and a high false-positive alarm rate remain persistent. This study aimed to improve the performance of intrusion detection systems by using the Human Evolutionary Optimization Algorithm (HEOA) for selecting effective features and by tuning the hyperparameters of the XGBoost model through a genetic algorithm. Inspired by human behaviors, HEOA has a strong capacity for rapid convergence and avoidance of local optima. The present study was conducted on two standard datasets, NSL-KDD and CIC-IDS 2017, and the preprocessing steps included normalization, encoding, and five-fold cross-validation. The evaluation results showed that the proposed HEOA-XGBoost model achieved higher accuracy and F1-score and produced a lower false-positive rate compared with conventional methods. These findings indicate the positive effect of using HEOA in improving the efficiency of machine learning models for intrusion detection. The main innovation of this study lies in applying HEOA to optimize feature selection and using a genetic algorithm for hyperparameter tuning, which enhances the accuracy and stability of intrusion detection systems.

Keywords: *Human Evolutionary Optimization Algorithm, intrusion detection system, feature selection, hyperparameter optimization, XGBoost, NSL-KDD, CIC-IDS 2017.*

How to cite this article:
Koochaki, M., & Sahafi, A. (2027). Improving Intrusion Detection Systems Using the Human Evolutionary Optimization Algorithm. Management Strategies and Engineering Sciences, 9(4), 1-15.

1. Introduction

The accelerating digitalization of organizational processes has fundamentally altered the structure, scope, and complexity of information security management. Contemporary organizations rely on interconnected networks, cloud infrastructures, industrial Internet of Things (IoT) environments, wireless communication systems, and data-driven decision platforms to maintain operational continuity, service quality, and competitive advantage. However, this dependence has simultaneously increased exposure to cyber threats, including denial-of-service attacks, probing activities, unauthorized access attempts, adversarial manipulation, and complex multi-stage intrusions. From a management perspective, network security is no longer a purely technical function but a strategic capability that supports organizational resilience, risk governance, business continuity, and stakeholder trust. Accordingly, intrusion detection systems (IDSs) have become essential components of cybersecurity management because they enable organizations to identify abnormal activities, respond to threats, and reduce the potential consequences of cyberattacks on information assets and operational processes [1, 2].

An intrusion detection system is designed to monitor network traffic or host behavior and detect suspicious patterns that may indicate malicious activity. In general, IDS approaches can be categorized into signature-based, anomaly-based, and hybrid methods. Signature-based systems identify known threats by comparing traffic patterns with predefined attack signatures, while anomaly-based systems detect deviations from expected behavior and are therefore more suitable for identifying unknown or emerging

attacks. Hybrid systems attempt to integrate the strengths of both approaches. Although traditional IDS models have contributed significantly to cyber defense, they often face limitations in highly dynamic and heterogeneous network environments. These limitations include poor adaptability to new attacks, high false-positive rates, difficulty in processing large-scale traffic, and reduced performance when attack patterns evolve rapidly. Machine learning has therefore become a dominant approach in modern intrusion detection because it enables models to learn complex nonlinear relationships from data and distinguish normal traffic from attack behavior more effectively [3, 4].

The increasing complexity of cyber threats has made machine learning-based IDS development a critical research area. Machine learning algorithms can automatically extract patterns from historical traffic data and support predictive classification of network events. This capability is especially relevant in environments where the volume, velocity, and diversity of data exceed the capacity of manual analysis. Recent studies have shown that machine learning methods can improve network security by enabling automated attack detection, adaptive decision-making, and real-time monitoring [2]. In addition, the application of machine learning is not limited to cybersecurity; it has also been widely used in managerial and financial decision-making domains, such as credit risk assessment, where predictive models support classification, risk estimation, and data-driven decision processes [5]. These developments indicate that machine learning provides a transferable analytical framework for solving complex classification problems in both management and technical domains.

Despite these advantages, machine learning-based IDS models continue to encounter major challenges. One of the most important problems is the high dimensionality of network traffic datasets. Intrusion detection datasets often contain many features, not all of which contribute equally to classification performance. Irrelevant, redundant, or noisy features can increase computational cost, reduce model interpretability, slow training, and degrade predictive accuracy. Another challenge is class imbalance, where normal traffic or certain attack types dominate the dataset while rare but dangerous attack categories are underrepresented. This imbalance can bias the model toward majority classes and reduce its ability to identify minority attacks. Furthermore, inappropriate parameter settings may prevent learning algorithms from reaching optimal performance. Therefore, effective IDS development requires not only the selection of a powerful classifier but also the use of robust preprocessing, feature selection, and hyperparameter optimization strategies [6, 7].

Data preprocessing is a foundational step in the development of reliable supervised learning models. Raw data often contain missing values, categorical variables, inconsistent scales, noise, and redundant patterns that must be treated before model training. In intrusion detection, preprocessing usually includes normalization, label encoding, data balancing, train-test splitting, and cross-validation. These procedures help ensure that the model receives structured and comparable input features and that performance evaluation is not distorted by data quality problems. In addition, preprocessing supports better generalization by reducing the influence of irrelevant variance in the dataset. The importance of preprocessing has been emphasized in supervised learning studies, where data preparation is recognized as a decisive factor affecting prediction accuracy, model stability, and learning efficiency [6]. Therefore, in IDS research, preprocessing should be considered an integral part of the modeling pipeline rather than a preliminary technical operation.

Feature selection has been widely investigated as a solution to the dimensionality problem in intrusion detection. The objective of feature selection is to identify a subset of informative features that maximizes classification performance while minimizing computational burden. Filter-based methods rank features according to statistical measures, wrapper-based methods evaluate feature subsets using a learning algorithm, and embedded methods perform feature selection during model training. Ambusaidi et al. proposed a filter-based feature selection approach for IDS development and demonstrated that selecting relevant features can improve detection performance while reducing complexity [7]. Similarly, ensemble-based and feature-selection-based IDS frameworks have been shown to produce efficient detection models by reducing redundant information and improving classifier performance [8]. These findings indicate that feature selection is not merely a dimensionality-reduction technique but a key optimization component in intelligent IDS design.

Metaheuristic algorithms have received increasing attention for feature selection and optimization problems because they are capable of searching large and complex solution spaces without requiring strict assumptions about the objective function. Metaheuristics are particularly useful in combinatorial optimization, where exhaustive search is computationally infeasible. They seek near-optimal solutions through iterative exploration and exploitation

mechanisms inspired by natural, social, physical, or evolutionary processes. Comprehensive reviews of metaheuristics have emphasized their flexibility, adaptability, and effectiveness in solving complex optimization problems across different domains [9, 10]. More recent surveys of new-generation metaheuristic algorithms have also highlighted their relevance for high-dimensional and nonlinear optimization tasks, including machine learning optimization, engineering design, and intelligent classification systems [11]. Therefore, the integration of metaheuristics with machine learning classifiers provides a promising pathway for improving IDS performance.

In intrusion detection research, several metaheuristic methods have been used to improve feature selection and model performance. Bostani and Sheikhan developed a hybrid binary particle swarm optimization and genetic algorithm approach for feature selection in IDSs, showing that hybrid metaheuristic strategies can improve the quality of selected features and enhance classifier performance [12]. Similarly, Mahboob and Moghaddam applied the butterfly optimization algorithm to anomaly-based intrusion detection, indicating that nature-inspired optimization can support more effective classification by improving the search for optimal solutions [13]. Dinesh and Kalaivani also examined the enhancement of IDS performance on the NSL-KDD dataset using metaheuristic and machine learning algorithms, suggesting that the combination of optimization and intelligent classification can improve detection outcomes [14]. These studies collectively demonstrate that metaheuristic optimization is a valuable direction for improving IDS accuracy, stability, and computational efficiency.

Among machine learning classifiers, XGBoost has gained considerable attention due to its strong predictive performance, scalability, and ability to handle complex structured data. XGBoost is an ensemble learning method based on gradient boosting that sequentially combines weak learners to produce a strong predictive model. Its advantages include regularization, handling of nonlinear relationships, robustness to heterogeneous features, and suitability for high-dimensional classification tasks. However, XGBoost performance depends strongly on hyperparameters such as the number of estimators, maximum tree depth, learning rate, subsampling ratio, and feature subsampling ratio. Poorly selected hyperparameters can lead to overfitting, underfitting, or inefficient learning. Therefore, hyperparameter tuning is essential for obtaining optimal model performance. In IDS applications, combining feature selection with hyperparameter optimization can improve both the efficiency and reliability of detection models [8, 14].

The need for robust IDS models has become more urgent as cyber threats evolve across IoT, industrial systems, wireless networks, and critical infrastructure. IoT environments introduce additional security challenges because they involve resource-constrained devices, heterogeneous communication protocols, and massive volumes of traffic. Research on IoT security has emphasized that learning-based intrusion detection is necessary for identifying attacks in distributed and dynamic environments [4]. In industrial IoT settings, online IDS models have been proposed to support continuous monitoring and timely detection of abnormal patterns [15]. Similarly, the security of smart and distributed energy networks increasingly depends on effective standards, protocols, and cyber defense mechanisms, as distributed energy resource management systems require secure communication and reliable monitoring infrastructures [16]. These developments show that IDS design must respond to broader organizational and infrastructural security requirements.

Another emerging concern is the vulnerability of machine learning-based IDS models to adversarial attacks. Adversarial machine learning refers to techniques in which attackers manipulate input data or exploit model weaknesses to evade detection or degrade model performance. Alhajjar et al. reviewed adversarial machine learning in network IDSs and emphasized that learning-based detection models may be vulnerable when attackers intentionally craft malicious inputs to mislead classifiers [17]. More recent work has also identified adversarial attacks against machine learning in network security as a serious threat to the reliability of intelligent cybersecurity systems [18]. In next-generation wireless networks, the emergence of native generative artificial intelligence attacks may create new forms of uncertainty and require evidence-based defensive approaches [19]. These trends suggest that IDS models should be accurate, stable, and resilient against complex and adaptive threats.

From a strategic information security management perspective, the development of optimized IDS models contributes to organizational readiness and cyber risk mitigation. National and organizational information infrastructures increasingly depend on secure data exchange, reliable network monitoring, and proactive threat detection. Studies on national information network scenarios have

emphasized that security consequences must be considered when designing future digital infrastructures [20]. In this context, IDS models are not isolated technical tools; they are part of a broader security architecture that supports governance, compliance, continuity, and resilience. Cyberattack management also requires systematic strategies for prevention, detection, response, and recovery, and intelligent IDS technologies can strengthen these strategies by enabling faster recognition of abnormal network behavior [1].

Benchmark datasets play an important role in evaluating IDS models and comparing new methods. The NSL-KDD dataset, developed as an improved version of the KDD Cup 99 dataset, remains widely used for evaluating intrusion detection algorithms. Tavallaee et al. provided a detailed analysis of the KDD Cup 99 dataset and highlighted several issues related to redundancy and evaluation reliability, which motivated the use of improved datasets such as NSL-KDD [21]. The CIC-IDS 2017 dataset has also become an important benchmark because it includes more contemporary traffic patterns and diverse attack scenarios. Using both datasets allows researchers to evaluate IDS performance under different data distributions, attack structures, and classification settings. This is particularly important because a model that performs well on one dataset may not generalize equally well to another dataset due to differences in feature structure, class imbalance, and attack diversity.

Accordingly, previous studies demonstrate that effective IDS development requires simultaneous attention to data preprocessing, feature selection, classifier optimization, and model evaluation. Although many studies have examined machine learning classifiers, feature selection methods, and metaheuristic algorithms separately, there remains a need for integrated frameworks that combine robust feature optimization with high-performance classification and systematic hyperparameter tuning. In particular, using a human-inspired evolutionary optimization strategy for selecting effective features and combining it with genetically optimized XGBoost can provide a structured approach for improving intrusion detection accuracy while reducing unnecessary feature complexity. Such an approach is especially relevant for management-oriented cybersecurity research because it links algorithmic optimization with practical needs for efficient, reliable, and scalable security decision-making.

The aim of this study is to improve the performance of intrusion detection systems by applying the Human Evolutionary Optimization Algorithm for effective feature selection and using a genetic algorithm to optimize XGBoost hyperparameters on the NSL-KDD and CIC-IDS 2017 datasets.

2. Methodology

2.1. Data Classification

One of the important preprocessing steps for this dataset is the classification or grouping of attack labels. Under normal conditions, the NSL-KDD and CIC-IDS 2017 datasets contain dozens of different attack names, each of which has its own specific nature and pattern.

Data classification in this dataset is particularly important because of the following reasons (Tolaei et al., 2009):

Simplification of modeling: Without classification, the machine learning model must predict a large number of separate labels, which increases complexity and requires more data.

Improvement of detection accuracy: Many algorithms experience a reduction in accuracy when dealing with a very large number of classes. Grouping attacks enables the model to learn more general and stable patterns.

Behavioral analysis of attacks: Classification makes it possible to study and analyze the common characteristics among similar attacks more deeply and contributes to a better understanding of the nature of threats.

Applicability in the real world: In operational environments, identifying the general type of attack is usually sufficient for rapid response, and there is no immediate need to identify the exact name of the attack.

Facilitation of IDS performance evaluation: By classifying attacks into four main groups, the performance of the system against each general type of threat can be easily evaluated and compared.

2.2. Data Preprocessing

In the development process of machine learning systems, the quality and structure of data are considered the most important factors in the success or failure of predictive and analytical models. Raw data usually contain noise, missing values, heterogeneous scales, and unstructured information, which can seriously reduce model performance if not properly prepared. Therefore, data preprocessing, as a key step prior to model training, plays an essential role in improving the accuracy, efficiency, and generalizability of algorithms.

Data preprocessing in machine learning helps the model correctly identify patterns and reduces the negative effects of peripheral factors and low-quality data. This process includes a set of systematic procedures, each pursuing a specific objective, including data classification for initial structuring, data normalization for placing features on a common scale, label encoding to convert categorical data into processable values, and data splitting into training, validation, and test sets for accurate evaluation of model performance.

By properly implementing these steps, raw data are transformed into a coherent, clean, and analysis-ready dataset, which can significantly increase prediction accuracy, algorithm convergence speed, and result stability. Consequently, data preprocessing is not merely a preliminary stage but an integral part of the life cycle of machine learning projects (Kotsiantis et al., 2007).

2.3. Human Exploration Stage

After initializing the population, the next step is to calculate the fitness of each solution. In the conducted experiments, the exploration stage was defined as the first $\frac{1}{4}$of the maximum number of iterations. In human developmental stages, when individuals encounter unknown territories and limited knowledge, they tend to adopt a uniform search strategy. This can be mathematically represented as follows:

$$X_i^{t+1} = \beta \cdot \left(1-\frac{t}{Max\_iter}\right) \cdot (X_i^t - X_{best}) \cdot Levy(dim) + X_{best} \cdot \left(1-\frac{t}{Max\_iter}\right) + (X_{mean}^t - X_{best}) \cdot floor\left(\frac{rand}{f_{jump}}\right) \cdot f_{jump}$$

In this equation, the adaptive function $\beta$is defined, and $t$represents the current number of iterations. dimindicates the dimensionality of the problem or the number of variables involved. X_i^tdenotes the current position, whereas X_i^{t+1}denotes the next updated position. X_{best}refers to the best position explored so far. X_{mean}^trepresents the mean position of the current population. $floor$is the downward rounding operator. $Levy$denotes the Lévy distribution. $f_{jump}$is the jump factor. $rand(0,1)$generates a random number in the interval $[0, 1]$.

The corresponding parameters and functions are explained as follows:

1. Mean position X_{mean}^t: The mean position of the current population, represented by X_{mean}^t, is calculated as follows:

$$X_{mean}^t = \frac{1}{N}\sum_{k=1}^{N} X_k^t$$

2. Adaptive function β: This function is responsible for adjusting the parameters based on the number of iterations and the current position. It is defined by considering the increasing difficulty of human exploration in acquiring knowledge and its cumulative characteristics:

$$\beta = 0.2\left(1-\frac{t}{Max\_iter}\right) \cdot (X_i^t - X_{mean}^t)$$

3. Lévy distribution: To simulate the complex nature of knowledge acquisition in the human exploration stage and the characteristics of spiral development, the Lévy distribution is used. This distribution helps the algorithm simulate nonlinear and unexpected movements in the search space. In this study, the value $\gamma = 1.5$was selected.

$$\begin{cases} Levy(D) = \dfrac{\mu \cdot \sigma}{|\ v\ |^{1/\gamma}} \\ \mu \sim N(0, D) \\ v \sim N(0, D) \\ \sigma = \left(\dfrac{\Gamma(1+\gamma) \cdot \sin(\pi\gamma/2)}{\Gamma((1+\gamma)/2) \cdot \gamma \cdot 2^{(1+\gamma)/2}}\right)^{\gamma+1} \end{cases}$$

4. Jump strategy: The human exploration stage includes a jump strategy inspired by image cutting and organization techniques. This strategy increases the dispersion of search locations. This approach preserves the essential characteristics of metadata and distributes the search across several regions. The jump factor, represented by f_{jump}, quantifies the degree of jumping and is defined as follows:

$$f_{jump} = \frac{lb(1) - ub(1)}{\delta}, \delta \in [100,2000]$$

2.3.1. Pseudocode of the Human Evolutionary Optimization Algorithm

In summary, HEOA begins the optimization process with a set of predefined candidates referred to as the population. This population is randomly generated using logistic chaotic mapping, which enhances the optimization process. During

the iterative process, the HEOA search strategy examines the proximity of solutions to the optimal solution or the best solution obtained so far. Each solution in HEOA updates its position based on information collected from all current search results. HEOA provides five different search strategies, including one for the exploration stage and four for the development stage. These strategies are designed to establish a balance between exploration and exploitation and ensure an effective search process. The HEOA search process terminates when a specific stopping condition is satisfied. This condition may include reaching the maximum number of iterations or achieving a satisfactory solution.

Input: Population X, maximum iterations Max_iter
Output: Best solution GBestX

Initialization phase:
Generate the initial population X using logistic chaotic mapping;
Compute the initial fitness values;
Start the search;
for i = 1 to Max_iter do
    BestF ← fitness(1);
    WorstF ← fitness(end);
    R ← rand(1);
    for j = 1 to size(X, 1) do
        Human exploration stage:
        if i ≤ 1/4 × Max_iter then
            X_new(j,:) ← GBestX · (1 − i/Max_iter)
            + (mean(X(j,:)) − GBestX)
            · floor(rand()/f_jump) + b · Levy(dim);
        end
        Human development stage:
        else
            Leaders:
            for j = 1 to LN_Number do
                if R < A then
                    X_new(j,:) ← w · X(j,:) · exp(−randn()/rand(1)/Max_iter);
                else
                    X_new(j,:) ← w · X(j,:) + randn() · ones(1, dim);
                end
            end
            Explorers:
            for j = LN_Number + 1 to LN_Number + EN_Number do
                X_new(j,:) ← randn() · exp((X(end,:) − X(j,:))/j²);
            end
            Followers:
            for j = LN_Number + EN_Number + 1 to
            LN_Number + EN_Number + FN_Number do
                X_new(j,:) ← X(j,:) + w · rand(1, dim)
                · (X(1,:) − X(j,:));
            end
            Losers:
            for j = LN_Number + EN_Number + FN_Number + 1 to N do
                X_new(j,:) ← GBestX + (GBestX − X(j,:)) · randn(1);
            end

```
    end
  end
  Update population X and apply boundary limits;
  Update GBestX;
end
```

Return: Best solution GBestX;

3. Findings and Results

The findings of the study are presented in relation to the structure of the datasets, attack-class distribution, feature-selection outputs, optimized hyperparameters, and the final predictive performance of the proposed HEOA-XGBoost model. Two benchmark intrusion-detection datasets, namely NSL-KDD and CIC-IDS 2017, were used to evaluate the proposed approach under different classification settings.

In the NSL-KDD dataset, the attack labels were organized into four main categories: DoS, Probe, U2R, and R2L. The DoS category included attacks such as back, land, neptune, pod, smurf, teardrop, udpstorm, apache2, processtable, and mailbomb. The Probe category included satan, ipsweep, nmap, portsweep, mscan, and saint. The U2R category included buffer_overflow, loadmodule, rootkit, perl, httptunnel, sqlattack, xterm, and ps. Finally, the R2L category included worm, guess_passwd, ftp_write, imap, phf, multihop, warezmaster, warezclient, spy, xlock, xsnoop, snmpguess, snmpgetattack, sendmail, and named. This grouping was applied to reduce the complexity of the classification task and to enable the model to learn more generalizable attack patterns.

Table 1. Class Distribution of the NSL-KDD Dataset

Class	KDDTrain+	KDDTest+	Total
Normal	67,343	9,711	77,054
DoS	45,927	7,458	53,385
Probe	11,656	2,421	14,077
R2L	995	2,754	3,749
U2R	52	200	252
Total attacks	58,630	12,833	71,463
Total samples	125,973	22,544	148,517

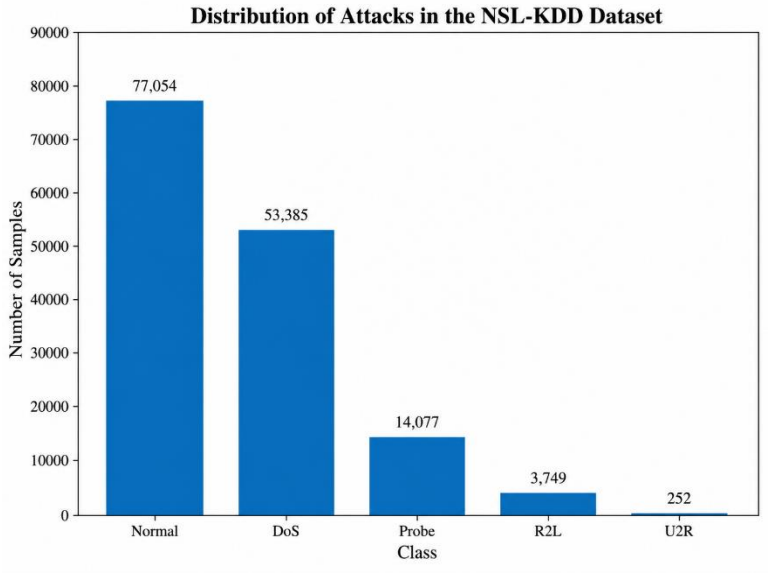

Figure 1. Distribution of Attacks in the NSL-KDD Dataset

To simplify the learning process and reduce problem complexity, the NSL-KDD dataset was used in two different forms. In the first configuration, the training and test subsets were merged and then split for multiclass classification into Normal, DoS, Probe, U2R, and R2L classes. In the second configuration, KDDTrain+ and KDDTest+ were used separately for training and testing, and the labels were converted into binary classes, namely Normal and Attack.

The CIC-IDS 2017 dataset consisted of normal traffic and several categories of network attacks. The original class distribution showed a strong imbalance between normal traffic and attack classes, with normal samples accounting for the largest proportion of the dataset.

Table 2. Class Distribution of the CIC-IDS 2017 Dataset

Attack/Class	Number of Samples
Normal	2,273,097
DoS Hulk	231,073
PortScan	158,930
DDoS	128,027
DoS GoldenEye	10,293
FTP-Patator	7,938
SSH-Patator	5,897
DoS slowloris	5,796
DoS Slowhttptest	5,499
Bot	1,966
Web Attack – Brute Force	1,507
Web Attack – XSS	652
Infiltration	36
Web Attack – SQL Injection	21
Heartbleed	11
Total	2,830,743

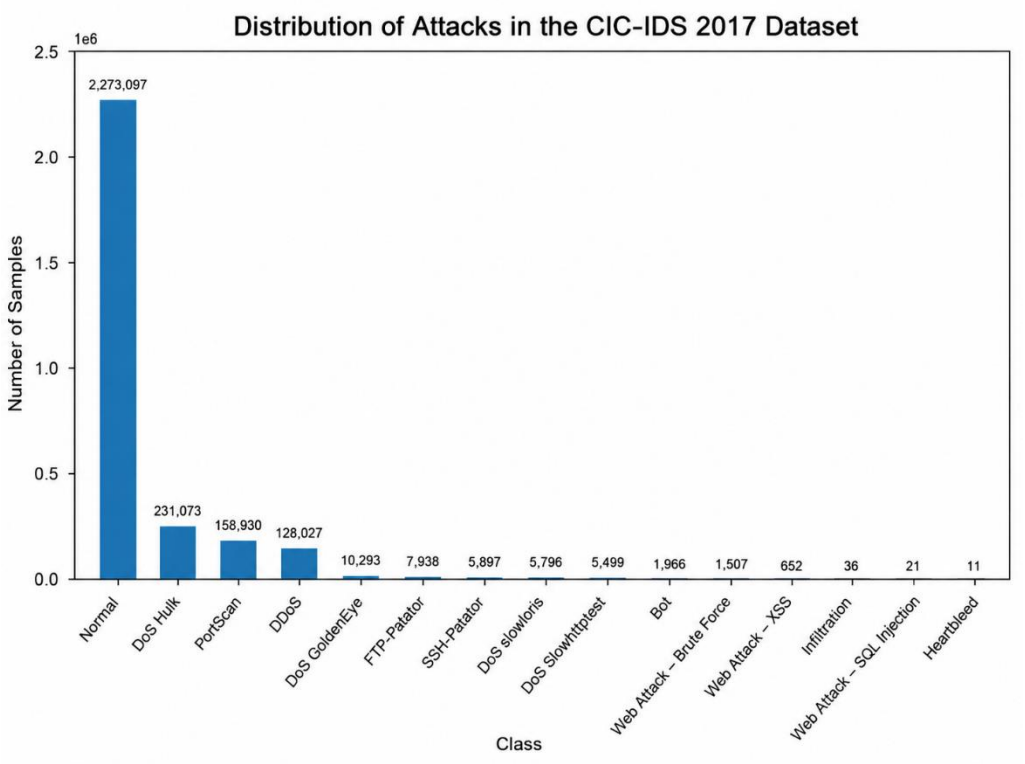

Figure 2. Distribution of Attacks in the CIC-IDS 2017 Dataset

For the CIC-IDS 2017 dataset, the attacks were also grouped into broader categories to facilitate modeling and improve detection performance. The DoS group included DDoS, DoS Hulk, DoS GoldenEye, DoS slowloris, DoS Slowhttptest, and Heartbleed. The BruteForce group included FTP-Patator and SSH-Patator. The Web group included Web Attack – Brute Force, Web Attack – XSS, and Web Attack – SQL Injection. After applying this grouping strategy, the final classes were used for multiclass detection in the CIC-IDS 2017 experimental scenario.

The parameter configuration of the Human Evolutionary Optimization Algorithm and the selected features differed across the three experimental scenarios. In the first scenario, the NSL-KDD dataset was used in a merged multiclass configuration. In the second scenario, the NSL-KDD dataset was used with separate train and test subsets under binary classification. In the third scenario, the CIC-IDS 2017 dataset was used in a merged multiclass configuration.

Table 3. HEOA Parameter Configuration and Selected Features Across Experimental Scenarios

Scenario	Parameter Configuration	Number of Selected Features	Selected Features
Scenario 1: NSL-KDD, merged data, multiclass classification	$N = 40$; $Max\_iter = 100$; $d = 41$; $lb = 0$; $ub = 1$; $K = 5$	24	duration, service, src_bytes, dst_bytes, land, num_failed_logins, num_compromised, root_shell, su_attempted, num_root, num_access_files, num_outbound_cmds, is_guest_login, count, rerror_rate, srv_rerror_rate, diff_srv_rate, dst_host_count, dst_host_same_srv_rate, dst_host_diff_srv_rate, dst_host_srv_diff_host_rate, dst_host_serror_rate, dst_host_rerror_rate, dst_host_srv_rerror_rate
Scenario 2: NSL-KDD, separate train/test data, binary classification	$N = 40$; $Max\_iter = 100$; d =number of input features; $lb = 0$; $ub = 1$; $K = 5$	25	duration, protocol_type, service, src_bytes, dst_bytes, wrong_fragment, num_failed_logins, logged_in, root_shell, su_attempted, num_root, num_file_creations, num_shells, num_access_files, num_outbound_cmds, is_host_login, is_guest_login, srv_count, serror_rate, srv_serror_rate, dst_host_same_srv_rate, dst_host_serror_rate, dst_host_srv_serror_rate, dst_host_rerror_rate, dst_host_srv_rerror_rate
Scenario 3: CIC-IDS 2017, merged data, multiclass classification	$N = 40$; $Max\_iter = 100$; d =number of input features; $lb = 0$; $ub = 1$; $K = 5$	25	Destination Port, Total Fwd Packets, Total Length of Fwd Packets, Fwd Packet Length Max, Fwd Packet Length Min, Bwd Packet Length Max, Bwd Packet Length Min, Flow Bytes/s, Flow IAT Max, Bwd PSH Flags, Bwd URG Flags, Fwd Packets/s, Max Packet Length, Packet Length Std, FIN Flag Count, Avg Bwd Segment Size, Fwd Avg Bytes/Bulk, Subflow Fwd Packets, Subflow Fwd Bytes, Init_Win_bytes_forward, Init_Win_bytes_backward, Active Mean, Active Std, Idle Mean, Idle Min

The convergence behavior of the HEOA algorithm was evaluated based on the reduction of mean squared error across iterations. In all three scenarios, the decreasing trend of MSE indicated that the algorithm gradually converged toward an optimized subset of features. This convergence pattern confirms the ability of HEOA to search the feature space effectively and identify more informative attributes for intrusion detection.

The hyperparameters of the XGBoost classifier were optimized using a genetic algorithm. The optimized values varied across the three experimental scenarios, reflecting the different structural characteristics of the datasets and classification settings.

Table 4. Optimized XGBoost Hyperparameters Across Experimental Scenarios

Hyperparameter	Scenario 1: NSL-KDD Multiclass	Scenario 2: NSL-KDD Binary	Scenario 3: CIC-IDS 2017 Multiclass
n_estimators	84	16	38
max_depth	17	6	8
learning_rate	0.3195	0.078	0.4
subsample	0.825	0.8624	0.5
colsample_bytree	0.645	0.5816	0.5

The results of model evaluation showed that the proposed HEOA-XGBoost model achieved strong performance across all three experimental conditions. In the first scenario, involving NSL-KDD multiclass classification, the model achieved an accuracy of 99.54%, precision of 96.73%, detection rate of 95.73%, and F1-score of 96.20%. These results indicate that the model was highly effective in distinguishing between normal records and different categories of attacks.

In the second scenario, involving binary classification on the NSL-KDD dataset with separate train and test subsets, the model achieved an accuracy of 95.42%, precision of

95.44%, detection rate of 95.22%, and F1-score of 95.32%. Although the overall performance was lower than that of the merged multiclass configuration, the results still demonstrate the model's acceptable capability in separating attack traffic from normal traffic.

In the third scenario, involving multiclass classification on the CIC-IDS 2017 dataset, the proposed model achieved the highest overall performance, with an accuracy of 99.76%, precision of 99.64%, detection rate of 95.87%, and F1-score of 97.43%. These results show that the model performed particularly well on the CIC-IDS 2017 dataset despite its high class imbalance and heterogeneous attack structure.

Table 5. Performance of the Proposed HEOA-XGBoost Model Across Experimental Scenarios

Model	Dataset	Classification Type	Accuracy (%)	Precision (%)	Detection Rate (%)	F1-Score (%)
HEOA-XGBoost	NSL-KDD	Multiclass	99.54	96.73	95.73	96.20
HEOA-XGBoost	NSL-KDD	Binary	95.42	95.44	95.22	95.32
HEOA-XGBoost	CIC-IDS 2017	Multiclass	99.76	99.64	95.87	97.43

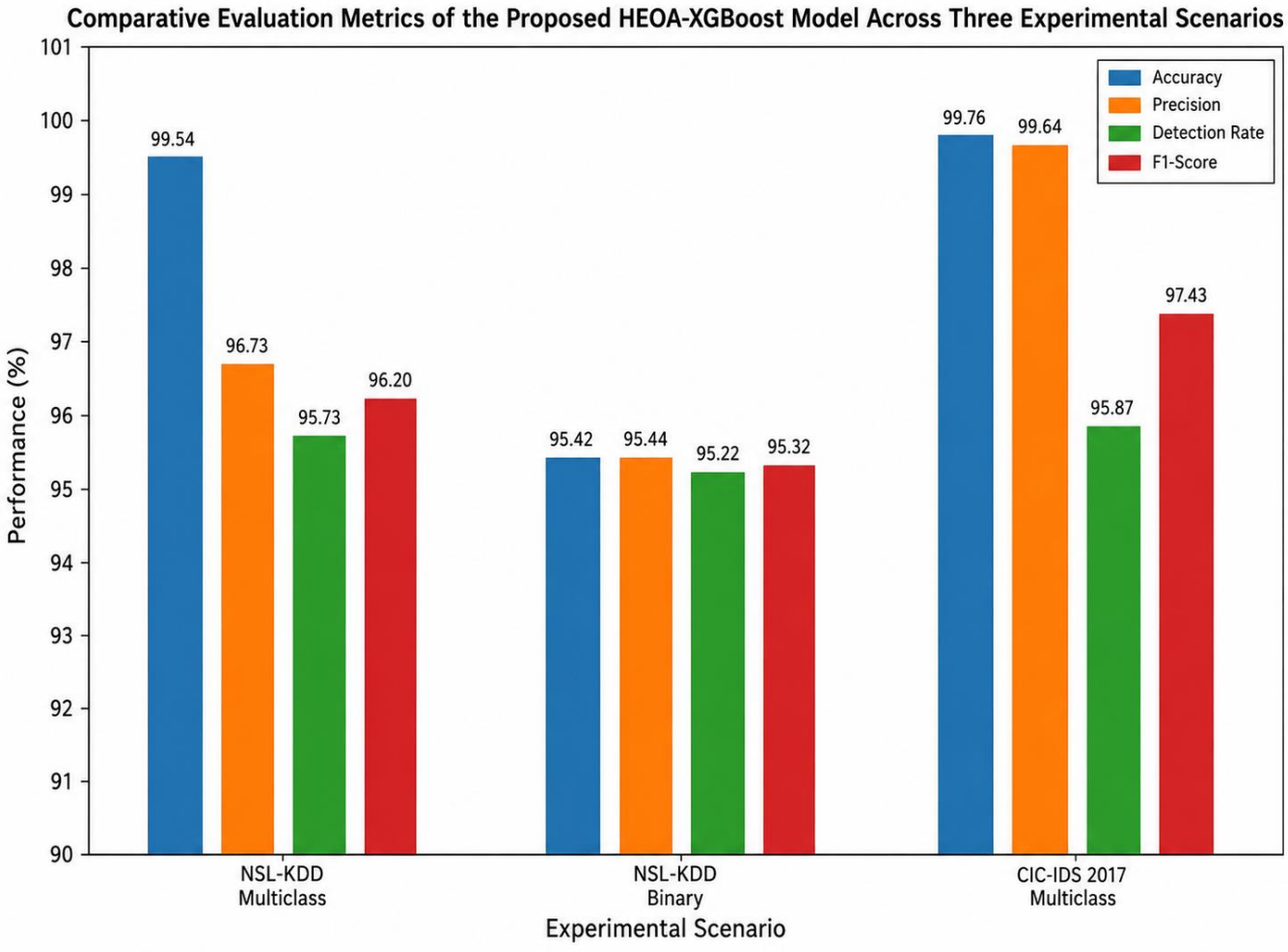

Figure 3. Comparative Evaluation Metrics of the Proposed HEOA-XGBoost Model Across the Three Experimental Scenarios

The comparison of the evaluation metrics showed that the proposed model obtained the best accuracy and precision in the CIC-IDS 2017 multiclass scenario. The NSL-KDD multiclass scenario also produced highly competitive results, particularly in terms of accuracy and false-positive rate. By contrast, the NSL-KDD binary scenario showed relatively lower performance and a higher false-positive rate, which may be attributed to the use of separate training and test subsets and the inherent distributional differences between KDDTrain+ and KDDTest+.

The confusion-matrix heatmaps for the three experimental scenarios also confirmed the quantitative results. In the NSL-KDD multiclass setting, the stronger concentration of values along the main diagonal indicated the accurate classification of most attack categories. In the NSL-KDD binary setting, the diagonal dominance of the confusion matrix showed that most normal and attack samples were correctly classified. In the CIC-IDS 2017 multiclass setting, the strong diagonal pattern further demonstrated the model's ability to distinguish among different attack classes, while the non-diagonal cells represented relatively limited classification errors.

Table 6. Comparative Summary of Model Performance Across Experimental Scenarios

Model	Dataset	Data Merging	Classification Type	Accuracy (%)	Precision (%)	Detection Rate (%)	F1-Score (%)	FPR (%)
HEOA-XGBoost	NSL-KDD	Yes	Multiclass	99.54	96.73	95.73	96.20	0.13
HEOA-XGBoost	NSL-KDD	No	Binary	95.42	95.44	95.22	95.32	4.78
HEOA-XGBoost	CIC-IDS 2017	Yes	Multiclass	99.76	99.64	95.87	97.43	0.40

Overall, the findings indicate that the proposed HEOA-XGBoost model achieved high detection performance across both NSL-KDD and CIC-IDS 2017 datasets. The differences in performance among the three scenarios can be attributed to the classification type, the use or non-use of data merging, the degree of class imbalance, and the intrinsic complexity of each dataset. The results also confirm that combining HEOA-based feature selection with genetically optimized XGBoost hyperparameters can improve the accuracy, stability, and reliability of intrusion detection systems.

4. Discussion and Conclusion

The present study aimed to improve the performance of intrusion detection systems by combining the Human Evolutionary Optimization Algorithm (HEOA) for feature selection with genetically optimized XGBoost classification. The findings demonstrated that the proposed HEOA-XGBoost model achieved strong performance across all three experimental scenarios. In the first scenario, using the NSL-KDD dataset in the merged multiclass setting, the model achieved an accuracy of 99.54%, precision of 96.73%, detection rate of 95.73%, F1-score of 96.20%, and false-positive rate of 0.13%. In the second scenario, using the original train-test separation of NSL-KDD in binary classification, the model achieved an accuracy of 95.42%, precision of 95.44%, detection rate of 95.22%, F1-score of 95.32%, and false-positive rate of 4.78%. In the third scenario, using the CIC-IDS 2017 dataset in the merged multiclass setting, the model achieved the highest overall performance, with an accuracy of 99.76%, precision of 99.64%, detection rate of 95.87%, F1-score of 97.43%, and false-positive rate of 0.40%. These results indicate that the integration of HEOA-based feature selection and genetic algorithm-based hyperparameter tuning can substantially enhance the classification capability of XGBoost in intrusion detection tasks.

The superior performance of the proposed model can first be explained by the role of feature selection in reducing dimensionality and improving the quality of the learning space. Intrusion detection datasets usually contain many redundant, irrelevant, or weakly informative features, and the inclusion of such features may increase computational complexity and reduce classification accuracy. In this study, HEOA selected 24 effective features in the NSL-KDD multiclass scenario and 25 effective features in the NSL-KDD binary and CIC-IDS 2017 multiclass scenarios. This reduction enabled the classifier to focus on the most discriminative network characteristics while avoiding unnecessary noise. The findings are consistent with previous IDS research showing that feature selection is a decisive factor in improving detection efficiency and reducing model complexity. Ambusaidi et al. emphasized that filter-based feature selection can improve IDS performance by identifying the most relevant attributes for attack detection [7]. Similarly, Zhou et al. showed that combining feature selection with ensemble classification can produce efficient intrusion detection models with stronger predictive performance [8]. The present results support this line of evidence and suggest that HEOA can be considered an effective search-based mechanism for selecting meaningful features in high-dimensional cybersecurity datasets.

The results also confirm the importance of metaheuristic optimization in enhancing machine learning-based IDS models. HEOA contributed to the search for an optimized feature subset through a balance between exploration and

exploitation. The observed convergence of mean squared error across iterations indicated that the algorithm gradually moved toward more suitable feature combinations. This behavior is theoretically consistent with the broader logic of metaheuristic algorithms, which are designed to solve complex optimization problems where exact search is impractical. Previous studies have emphasized that metaheuristics are especially useful for combinatorial optimization because they can search large solution spaces and identify high-quality approximate solutions within a reasonable computational time [9, 10]. More recent reviews have also argued that new-generation metaheuristic algorithms are particularly valuable in complex, nonlinear, and high-dimensional machine learning tasks [11]. Therefore, the high performance observed in this study can be attributed partly to the capacity of HEOA to identify compact and informative feature subsets.

The findings are also aligned with previous studies that have applied evolutionary and swarm-based optimization methods to intrusion detection. Bostani and Sheikhan used a hybrid binary particle swarm optimization and genetic algorithm approach for feature selection in IDS and reported that hybrid optimization can improve classification performance by selecting more relevant attributes [12]. Mahboob and Moghaddam similarly applied the butterfly optimization algorithm to anomaly-based intrusion detection and showed that metaheuristic optimization can strengthen detection performance [13]. Dinesh and Kalaivani also demonstrated that combining metaheuristic methods with machine learning algorithms can enhance IDS performance on the NSL-KDD dataset [14]. The present study extends this research direction by using HEOA as the feature-selection mechanism and genetic optimization for XGBoost hyperparameter tuning. The strong results across NSL-KDD and CIC-IDS 2017 indicate that the proposed hybrid strategy is not limited to a single dataset structure but can perform effectively under different classification conditions.

Another important finding concerns the contribution of hyperparameter optimization. XGBoost is a powerful ensemble classifier, but its performance depends heavily on hyperparameters such as the number of estimators, maximum depth, learning rate, subsampling rate, and column subsampling ratio. In this study, the genetic algorithm produced different optimized hyperparameter configurations for each scenario, indicating that the best model structure depends on dataset characteristics and classification type. For example, the NSL-KDD multiclass scenario required a higher number of estimators and deeper trees, while the binary NSL-KDD scenario performed best with fewer estimators and a lower learning rate. The CIC-IDS 2017 scenario achieved the best performance with moderate depth and a higher learning rate. These differences indicate that fixed hyperparameter values are unlikely to be optimal across heterogeneous IDS datasets. This finding is consistent with studies emphasizing that machine learning-based intrusion detection requires careful model optimization to achieve high accuracy and generalizability [2, 3].

The high performance of the proposed model in the CIC-IDS 2017 multiclass scenario is particularly noteworthy because this dataset contains a much larger number of samples and more diverse traffic patterns than NSL-KDD. The model achieved 99.76% accuracy and 99.64% precision, suggesting that the selected features and optimized XGBoost structure were highly effective in distinguishing normal traffic from different attack categories. This result is important because modern network environments, especially IoT and industrial IoT infrastructures, produce large volumes of heterogeneous data and require scalable intrusion detection solutions. Chaabouni et al. highlighted the importance of learning-based IDS models for IoT security because of the complexity, heterogeneity, and vulnerability of connected devices [4]. Gyamfi and Jurcut also emphasized the need for online network intrusion detection in industrial IoT environments, where rapid and reliable detection is required to protect operational systems [15]. The strong performance of HEOA-XGBoost on CIC-IDS 2017 therefore suggests that the proposed approach has potential relevance for modern, data-intensive cybersecurity environments.

The lower performance observed in the NSL-KDD binary scenario compared with the two merged multiclass scenarios requires careful interpretation. Although the model still achieved acceptable results, including an accuracy of 95.42% and an F1-score of 95.32%, the false-positive rate increased to 4.78%. This difference may be related to the use of the original train-test split, where KDDTrain+ and KDDTest+ differ in attack distribution and difficulty. Tavallaee et al. previously emphasized that KDD-based datasets contain structural issues that may influence IDS evaluation and that differences between training and testing subsets can affect performance interpretation [21]. Therefore, the lower binary performance does not necessarily indicate weakness in the proposed model; rather, it may reflect distributional divergence between training and testing data. This finding also reinforces the importance of

robust validation strategies, preprocessing, and realistic evaluation protocols in IDS research [6].

The findings further show that the proposed approach can reduce false-positive rates, especially in the merged NSL-KDD multiclass setting, where the false-positive rate reached only 0.13%. This is a significant result because high false-positive rates are one of the main barriers to the operational adoption of IDS technologies. In practical security management, false alarms can overwhelm analysts, reduce trust in automated systems, increase response costs, and delay the identification of real threats. Therefore, a model that combines high detection accuracy with a low false-positive rate has direct managerial and operational value. This finding is consistent with the broader cybersecurity management literature, which emphasizes that effective information security strategies must not only detect attacks but also support efficient decision-making and resource allocation [1]. From this perspective, the proposed HEOA-XGBoost model contributes to both technical detection performance and organizational cyber-risk management.

The results should also be interpreted in relation to the growing concern about adversarial and next-generation cyber threats. Although the present model achieved high accuracy, modern IDSs must operate in adversarial environments where attackers may intentionally manipulate traffic patterns to evade detection. Alhajjar et al. emphasized that adversarial machine learning is a major challenge for IDS because attackers can exploit model vulnerabilities and generate deceptive inputs [17]. Ibitoye et al. also noted that adversarial attacks against machine learning in network security are becoming increasingly important and require more resilient detection frameworks [18]. In addition, emerging generative artificial intelligence-based attacks may create new threat patterns in next-generation wireless networks [19]. Although the present study did not directly test adversarial robustness, the use of optimized feature selection and tuned classification may provide a stronger foundation for future adversarially resilient IDS designs.

From a broader infrastructure and management perspective, the results support the strategic importance of intelligent intrusion detection in digital transformation. Modern organizations and national infrastructures increasingly depend on secure and reliable network systems. Research on distributed energy resource management systems has shown that smart and secure energy networks require strong standards, protocols, and security mechanisms to protect complex digital infrastructures [16]. Similarly, scenario-based analysis of national information networks emphasizes that security consequences must be considered in long-term digital infrastructure planning [20]. The proposed model, by improving detection performance and reducing feature complexity, can contribute to such security architectures by supporting more accurate and efficient monitoring. Moreover, the relevance of machine learning to decision-oriented domains such as financial risk assessment shows that intelligent classification systems can support management decisions across multiple fields [5].

Overall, the findings indicate that the proposed HEOA-XGBoost model provides a strong and efficient framework for intrusion detection. Its strength lies in the integration of three complementary mechanisms: systematic preprocessing, HEOA-based feature selection, and genetically optimized XGBoost classification. The model performed especially well in the CIC-IDS 2017 multiclass scenario and the merged NSL-KDD multiclass scenario, while maintaining acceptable performance in the more challenging NSL-KDD binary train-test setting. These results are consistent with previous studies showing that machine learning, feature selection, and metaheuristic optimization can improve IDS performance [3, 8, 14]. Therefore, the study contributes to the IDS literature by demonstrating that human-inspired evolutionary optimization can effectively enhance feature selection and improve the stability and accuracy of machine learning-based intrusion detection.

The present study has several limitations. First, the evaluation was conducted using two benchmark datasets, NSL-KDD and CIC-IDS 2017, which, although widely used, may not fully represent the complexity and variability of real-time enterprise networks. Second, the study focused on classification performance metrics such as accuracy, precision, detection rate, F1-score, and false-positive rate, while computational cost, memory consumption, and real-time processing latency were not examined in detail. Third, although the proposed model showed strong detection performance, its robustness against adversarial examples, encrypted traffic, zero-day attacks, and concept drift was not empirically tested. Finally, the study relied on a specific combination of HEOA, genetic algorithm optimization, and XGBoost; therefore, the comparative advantage of this framework against all possible deep learning, online learning, and hybrid ensemble architectures remains open to further investigation.

Future research should evaluate the proposed model on additional modern datasets and real organizational network

traffic to examine its generalizability under practical deployment conditions. Further studies should also investigate the computational efficiency of HEOA-XGBoost, including training time, inference speed, scalability, and resource requirements in large-scale and real-time environments. Another important direction is to examine the adversarial robustness of the model by testing it against evasion attacks, poisoning attacks, and generative artificial intelligence-based attack patterns. Future work may also compare HEOA with other recent metaheuristic algorithms and integrate it with deep learning, online learning, or federated learning architectures. In addition, future studies can explore adaptive feature selection mechanisms that update selected features over time as network behavior and attack distributions evolve.

From a practical perspective, cybersecurity managers and network administrators can use the findings of this study to improve IDS deployment strategies by combining optimized feature selection with carefully tuned machine learning models. The proposed framework can reduce unnecessary input features, improve detection accuracy, and support more efficient monitoring of network traffic. In operational environments, the model should be integrated into a broader security management system that includes continuous data collection, periodic retraining, performance monitoring, incident response procedures, and analyst feedback. Organizations should also evaluate the model under their own traffic conditions before full deployment, because attack patterns, network architecture, and operational priorities differ across contexts. Finally, practitioners should treat optimized IDS models as decision-support tools rather than standalone security solutions and combine them with complementary security controls such as firewalls, endpoint protection, threat intelligence, access management, and security awareness programs.

Authors' Contributions

Authors equally contributed to this article.

Acknowledgments

Authors thank all participants who participate in this study.

Declaration of Interest

The authors report no conflict of interest.

Funding

According to the authors, this article has no financial support.

Ethical Considerations

All procedures performed in this study were under the ethical standards.

References

[1] M. Hosseingholipour, "Information Security Management Strategies to Counter Cyberattacks in Computer Networks," in *Twenty-Fourth National Conference on Applied Research in Electrical, Computer, and Biomedical Engineering Sciences*, 2025.

[2] Y. Tang, "The Application of Machine Learning in the Field of Network Security," *Tcsisr,* vol. 7, pp. 363-369, 2024, doi: 10.62051/cw085k64.

[3] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Communications Surveys & Tutorials,* vol. 18, no. 2, pp. 1153-1176, 2016, doi: 10.1109/COMST.2015.2494502.

[4] N. Chaabouni, M. Mosbah, A. Zemmari, C. Sauvignac, and P. Faruki, "Network intrusion detection for IoT security based on learning techniques," *IEEE Communications Surveys & Tutorials,* vol. 21, no. 3, pp. 2671-2701, 2019, doi: 10.1109/COMST.2019.2896380.

[5] Tafreshi, "Credit risk assessment using machine learning algorithms: Logistic regression, artificial neural network, and support vector machine," *Biannual Journal of Financial Knowledge of Securities Analysis,* vol. 18, no. 65, pp. 45-62, 2025.

[6] S. Kotsiantis, D. Kanellopoulos, and P. Pintelas, "Data preprocessing for supervised learning," *International Journal of Computer Science,* vol. 1, no. 2, pp. 111-117, 2007.

[7] M. A. Ambusaidi, X. He, P. Nanda, and Z. Tan, "Building an intrusion detection system using a filter-based feature selection algorithm," *IEEE Transactions on Computers,* vol. 65, no. 10, pp. 2986-2998, 2016, doi: 10.1109/TC.2016.2519914.

[8] Y. Zhou, G. Cheng, S. Jiang, and M. Dai, "Building an efficient intrusion detection system based on feature selection and ensemble classifier," *Computer Networks,* vol. 174, p. 107247, 2020, doi: 10.1016/j.comnet.2020.107247.

[9] C. Blum and G. R. Raidl, "Metaheuristics in combinatorial optimization: Overview and conceptual comparison," *European Journal of Operational Research,* vol. 285, no. 3, pp. 801-814, 2020, doi: 10.1016/j.ejor.2020.02.013.

[10] R. Marti, *Handbook of Heuristics*. Springer, 2021.

[11] T. Dokeroglu, E. Sevinc, T. Kucukyilmaz, and A. Cosar, "A survey on new generation metaheuristic algorithms," *Computers & Industrial Engineering,* vol. 137, pp. 105-127, 2022, doi: 10.1016/j.cie.2019.106040.

[12] H. Bostani and M. Sheikhan, "Hybrid of binary PSO and GA for feature selection in intrusion detection systems," *Neural Computing and Applications,* vol. 29, no. 11, pp. 1043-1057, 2018, doi: 10.1007/s00521-016-2558-2.

[13] A. S. Mahboob and M. R. O. Moghaddam, "An anomaly-based intrusion detection system using butterfly optimization algorithm," in *2020 6th Iranian Conference on Signal*

Processing and Intelligent Systems (ICSPIS), 2020, pp. 1-6, doi: 10.1109/ICSPIS51611.2020.9349537.

[14] K. Dinesh and D. Kalaivani, "Enhancing performance of intrusion detection system in the NSL-KDD dataset using meta-heuristic and machine learning algorithms-design thinking approach," in *2023 International Conference on Sustainable Computing and Smart Systems (ICSCSS)*, 2023, pp. 1471-1479, doi: 10.1109/ICSCSS57650.2023.10169845.

[15] E. Gyamfi and A. Jurcut, "Novel online network intrusion detection system for industrial IoT based on OI-SVDD and AS-ELM," *IEEE Internet of Things Journal,* vol. 9, no. 23, pp. 23934-23947, 2022, doi: 10.1109/JIOT.2022.3172393.

[16] P. K. Kurella and S. Mikkili, "A Comprehensive Review of DERMS for Smart and Secure Energy Networks: Standards, Protocols, and Security Considerations," (in English), *International Journal of Ambient Energy,* vol. 46, no. 1, 2025, doi: 10.1080/01430750.2025.2506703.

[17] E. Alhajjar, P. Maxwell, and N. Bastian, "Adversarial machine learning in network intrusion detection systems," *Expert Systems with Applications,* vol. 186, p. 115782, 2021, doi: 10.1016/j.eswa.2021.115782.

[18] O. Ibitoye, R. Abou-Khamis, M. e. Shehaby, A. Matrawy, and M. O. Shafiq, "The Threat of Adversarial Attacks Against Machine Learning in Network Security: A Survey," *J. Electron. Electric. Eng.,* 2025, doi: 10.37256/jeee.4120255738.

[19] C. Chaccour and W. Saad, "Securing next-generation wireless networks against native GenAI attacks: An evidence-theoretic approach," in *IEEE International Conference on Communications*, Abu Dhabi, UAE, 2025.

[20] M. Shirvani Naghani, K. Koolivand, E. Ejabi, and M. Heidari, "Drawing Scenarios for the National Information Network of the Islamic Republic of Iran on the Horizon of 2028 with an Emphasis on Security Consequences," *Majlis and Rahbord,* vol. 31, no. 120, 2024. [Online]. Available: https://www.magiran.com/paper/2834591/outlining-scenarios-for-the-national-information-network-of-the-islamic-republic-of-iran-by-2028-with-emphasis-on-security-implications?lang=en.

[21] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *Proceedings of the IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA)*, Ottawa, ON, Canada, 2009, pp. 1-6, doi: 10.1109/CISDA.2009.5356528.