International Journal of Innovative Research and Scientific Studies, 8(3) 2025, pages: 5117-5131

IJIRSS
International Journal of Innovative Research & Scientific Studies

ISSN: 2617-6548

URL: www.ijirss.com

Effective intrusion detection approach with ant colony optimization based feature selection and XgBoost classifier

Shweta Bhardwaj[1*], Seema Rawat[2], Hima Bindu Maringanti[3]

[1]*Department of Computer Science & Engineering Amity University Uttar Pradesh Noida, India.*
[2]*Department of Information Technology Amity University Uttar Pradesh Noida, India.*
[3]*Department of Computer Science and Applications North Orissa University, Baripada Odisha, India.*

Corresponding author: Shweta Bhardwaj (*Email: sbhardwaj1@amity.edu*)

Abstract

Secure data communication over the internet and any other network is always at risk; therefore, the intrusion detection system has become a necessary component of computer network systems. Hence, this paper proposes intrusion detection with dimensionality-reduced features, ant colony optimization-based feature selection, and Extreme Gradient Boost (XG-Boost) classifier. For better performance, the proposed system uses the NSL-KDD dataset along with dataset preprocessing. The dimensionality reduction is accomplished using principal component analysis (PCA), and the feature selection is performed using Ant Colony Optimization (ACO). Then, the classification is performed with the Extreme Gradient Boost Algorithm (XG-Boost). The efficiency of the proposed system is evaluated using performance metrics such as F1-score, precision, specificity, accuracy, and recall. Therefore, the obtained results showed 97.6% precision, 97.85% accuracy, 97.88% F1-score, 99.64% specificity, and 97.56% recall.

Keywords: Ant Colony optimization, Intrusion detection, Principal component analysis, XG-Boost.

DOI: 10.53894/ijirss.v8i3.7710
Funding: This study received no specific financial support.
History: Received: 24 April 2025 / **Revised:** 27 May 2025 / **Accepted:** 2 June 2025 / **Published:** 10 June 2025

Competing Interests: The authors declare that they have no competing interests.
Authors' Contributions: All authors contributed equally to the conception and design of the study. All authors have read and agreed to the published version of the manuscript.
Transparency: The authors confirm that the manuscript is an honest, accurate, and transparent account of the study; that no vital features of the study have been omitted; and that any discrepancies from the study as planned have been explained. This study followed all ethical practices during writing.
Publisher: Innovative Research Publishing

1. Introduction

In the modern world, networks play a major role in the day-to-day lives of humans for communication and data sharing. They increase the risk of intrusion, and malicious activity disturbs user data [1]. This leads to cyber security being a dynamic investigation zone. An intrusion detection system is an important cyber security model that monitors the state of hardware and software running in the network. Even after over a decade of development, existing intrusion detection systems face issues in enhancing accuracy and detecting unknown attacks (or) spam. To overcome this issue, many researchers have

focused on improving intrusion detection systems, which capitalize on the machine learning (ML) approach [2]. The ML approach can easily differentiate between the abnormal data and normal data with high accuracy [3].

In addition to this, ML provides strong generalizability, which is also able to detect unknown attacks. Therefore, this stimulates the planning of an effective intrusion detection approach with ant colony optimization based on feature selection and an XG-boost classifier for the efficient detection of intrusions. The NSL-KDD dataset [3-5] is the updated version of KDD Cup 99, which was an effective benchmark for researchers to compare various intrusion detection systems. It supports building an IDS (intrusion detection system) on the network (or) host-based, and it provides original records in the test sets. An adequate amount of records is present in the test and train datasets. Therefore, these aspects stimulate the system to use the NSL-KDD dataset.

The dataset may contain some (or) irrelevant data that may disturb the performance of the proposed system. Therefore, data pre-processing is performed on the selected dataset. The dataset may contain non-numeric values; consequently, with the involvement of a 1 to N encoder, the non-numeric values are converted into numeric values, and normalization is also performed, which reduces redundant data and provides a better overall organization of data [6, 7]. Then, the dimensionality reduction is performed with PCA [8, 9]. It computes the principal components and uses them to achieve a change of basis on the data. It is mainly used in exploratory data evaluation and for making predictive systems. It improves the performance efficiency of the ML algorithm and removes the correlated variables that do not contribute to any decision-making process. It also supports overcoming data overfitting problems [10]. The classification mainly depends on feature selection. Therefore, to enhance feature selection, the system used ant colony optimization. It was designed to solve the computational issues that can be minimized to find decent paths through the graph.

The artificial ants denote the multi-agent approach inspired by the real ant behaviors. It is a population-based metaheuristic that supports finding an approximate solution to difficult optimization problems [11, 12]. Then, the classification was performed using the extreme Gradient Boosting algorithm. It considered the prevalent supervised learning algorithm's support for classification and regression on a huge dataset. It supports a sequentially built shallow decision tree to conserve accurate outcomes and a highly scalable training model that avoids overfitting problems [13, 14]. These aspects have created an interest in proposing this system. With the implementation of these methods, we expect to attain a high level of accuracy in intrusion detection and improve other metrics such as recall, specificity, F1-score, and precision.

The major contribution of the proposed system is to enhance the feature selection process. The proposed system performed dimensionality reduction with Principal Component Analysis, eliminating the correlated variables that do not contribute to any decision-making process and also helping to overcome the data overfitting issue. The classification mainly depends on the feature selection process. Therefore, the proposed system used Ant Colony Optimization for feature selection, which finds an approximate solution to optimization problems. To attain improved and efficient results, the proposed system used the XGBoost classifier, which sequentially built shallow decision trees to provide accurate results and highly scalable training methods that avoid overfitting issues. To analyze the performance of the proposed system by evaluating its efficiency in terms of performance metrics such as precision, specificity, F1-score, accuracy, and recall.

The paper's organization is as follows. Section 1 deliberates the significance of various intrusion detection methods, the importance of the NSL-KDD dataset is explained, and the significance of PCA in dimensionality reduction, ACO in feature selection, and XGBoost in the classification process is discussed. Section 2 explains the information about existing methods related to the proposed system. Section 3 deliberates the research methodology. Section 4 states the results obtained in the proposed system. Section 5 summarizes the conclusion of the paper.

2. Related Work

Recently, network traffic has increased drastically, and various models have been implemented in intrusion detection. Some of them are reviewed in this section. Since the false alarm rates were high and moderate accuracy was attained in recent anomaly detection, various machine learning algorithms were used in the KDD-99 Cup and NSL-KDD dataset to evaluate performance [15]. Similarly, [16, 17] takes the NSL-KDD dataset as the research object, and the existing problems and latest progress were analyzed. Therefore, the overview of data mining and machine learning used in intrusion detection systems was stated, and the CSE-CIC-IDS-2018 & CIC-IDS-2017 recent datasets were discussed [18]. Accordingly, NSL-KDD dataset was used for intrusion detection with a low error rate and high accuracy [19]. To detect the intrusions effectively, the NSL-KDD dataset was used and new patterns were created using a genetic algorithm [20]. Therefore, the system was reviewed by evaluating more recent datasets, and a synopsis was provided for the feature selection approaches [21]. It evaluated several experimental approaches such as time complexity, performance and feature correlation.

The author Abdulhammed et al. [22] used an auto-encoder and principal component analysis to reduce dimensionality. Similarly, the paper [23] introduced a hybrid PCA-firefly algorithm for dimensionality reduction, and the classification of the reduced dataset was performed with the XGBoost algorithm. This increased accuracy and detection efficiency [24] introduced linear discriminant analysis and principal component analysis for dimensionality reduction. Accordingly, Zhang et al. [25] introduced an intrusion detection prototype based on improved PCA combined with Gaussian Naïve Bayes. The dimensionality reduction was achieved by improved PCA.

Similarly, the system [26] used mutual information and principal component analysis for dimensionality reduction and feature selection. Therefore, multi-linear dimensionality reduction was introduced as a feature extraction model to reduce the dimensions. Then, the NSL-KDD dataset was used for evaluation [27]. Feature selection was considered one of the main and important steps in designing an intrusion detection system. Therefore, the paper [28] presented cuttlefish and ant colony optimization for the feature selection since these algorithms were familiar and had solved various optimization problems.

Therefore, the author Tama et al. [29] introduced a hybrid feature selection comprising of ant colony algorithm, a genetic algorithm, and PSO (particle swarm optimization) and UNSW-NB 15 & NSL-KDD datasets for evaluation.

Similarly, the author Kalimuthan and Renjit [30] reviewed the existing benchmark data to detect unusual attacks and present problems in intrusion detection. The performance of the existing intrusion detection models was evaluated using classification and feature selection of machine learning approaches. The system [31] introduced an improved feature selection algorithm, i.e., FACO was combined with the ant colony optimization algorithm and feature selection.

Further, the classification tends to gain efficient results in detecting intrusion. Therefore, the author [32] implemented an XG-Boost classifier for accurate prediction results with the NSL-KDD datasets. Similarly, to enhance the prediction result, the paper [33] introduced an XG-Boost with ensemble-based intrusion detection; for better outcomes, XG-Boost was based on the tree-boost ML algorithms that support dealing with a smoother "bias-variance" tradeoff. In an effort to increase the overall accuracy rate, the system [34] introduced PSO-XG-Boost using the NSL-KDD dataset. The comparative analysis was performed concerning the mean average precision, recall, precision, and macro-average, particularly in identifying minority groups such as R2L and U2R. The system [35] identified minority and majority modules at the algorithm level without using data balancing methods. The first layer of the system used Siamese Neural Network, Deep Neural Network and extreme binary-Gradient boosting for hierarchical filtration of input specimens; then, the second layer was performed with a multi-class XG-Boosting classifier for efficient classification results using CIDDS-001 and NSL-KDD datasets. The author Kilincer et al. [36] selected LGBM and XGBoost algorithms for classification; furthermore, to improve the classification results, hyperparameter optimization was employed for the XGBoost and LGBM algorithms. In terms of processing time and performance, the LGBM classifier surpassed the XGBoost classifier. The paper [37] demonstrated the usage of ML algorithms for monitoring and detecting malicious activities in the network as part of NIDS in the SDN controller. Various advanced and traditional tree-based ML approaches, such as random forest, XGBoost, and decision tree, were chosen to demonstrate intrusion detection.

Therefore, the author Song et al. [38] introduced an intrusion detection model mainly based on the XGBoost classifier. PCA was used for dimensionality reduction and the WOA-XGBoost algorithm to improve overall intrusion detection. Therefore, the system [39] built a logarithmic autoencoder and Extreme Gradient Boosting. A logarithmic autoencoder was built to learn the hidden features in the input data and generate new data similar to the training samples. Then, the XGBoost was employed to identify the dataset that combined the original dataset with the generated dataset. Accordingly, the system [40] proposed a multi-granularity feature generation+XGBoost approach for enhancing intrusion detection.

In recent times, technology has grown drastically with the increased rate of intrusion. Though the intrusion detection system [37, 40] was implemented efficiently in the detection of intrusion; still, it still faces a low accuracy issue. The detection can be assured only with the high accuracy that various systems lack. Similarly, sudden attack (or) unknown attack detection is still a challenging issue [10]. Many failed to enhance the feature selection since the classification mainly depends on the feature selection.

3. Methodology

The overall work of this research is stated in this section. Figure 1 represents the overall workflow of the proposed system.

The proposed system used the NSL-KDD dataset for evaluation. Initially, the dataset is loaded and pre-processed. The dataset can't be fed directly into the dimensionality reduction process since the original dataset may contain a large amount of data, and there may be some unwanted and irrelevant data. This may affect the prediction rate. Therefore, the dataset is pre-processed, and then the dimensionality reduction is performed on the pre-processed data. The dimensionality reduction reduces the dimension of the data. In this paper, the dimensionality reduction was performed with principal component analysis, eliminating the correlated variables that do not contribute to any decision-making process. Additionally, it helps overcome the data overfitting problems by reducing the number of features. After the dimensionality reduction process, feature selection is performed. The dataset may still contain irrelevant or unwanted data, so feature selection is involved in selecting the relevant features. Mainly, feature selection enhances the efficiency of classification. Therefore, in the proposed work, feature selection is performed with Ant Colony Optimization. ACO is a population-based metaheuristic optimization method used to find the approximate solution for problematic optimization.

The ant colony optimization initially generates the data and evaluates each position of the data; if the data is relevant, the subsets are gathered; if not, the following data is selected for evaluation until the subset is gathered. After the gathering of the subset, an evaluation for stopping the process is done. If the gathered subset data are best and relevant, the subset is returned; if not, the pheromones are updated, and the same process is repeated until the system attains the best subset. After the completion of feature selection, the selected features are randomly chosen for training and testing. In this, 80% of the data are trained, and 20% of the data is used for testing. Then, the trained and test results are fed into classification. In this system, the classification is performed with the XGBoost classifier. It is a widespread supervised learning algorithm employed for classification and regression on huge datasets. It uses a successively built shallow decision tree to provide accurate outcomes and an extremely scalable training methodology that evades overfitting problems. After this, the classification outcomes are trained, and the prediction phase is evaluated. The prediction phase detects whether the intrusion is present or not. Then, the system's performance is evaluated in terms of performance metrics such as precision, accuracy, recall, F1-score, and specificity to determine the efficiency of the proposed system.

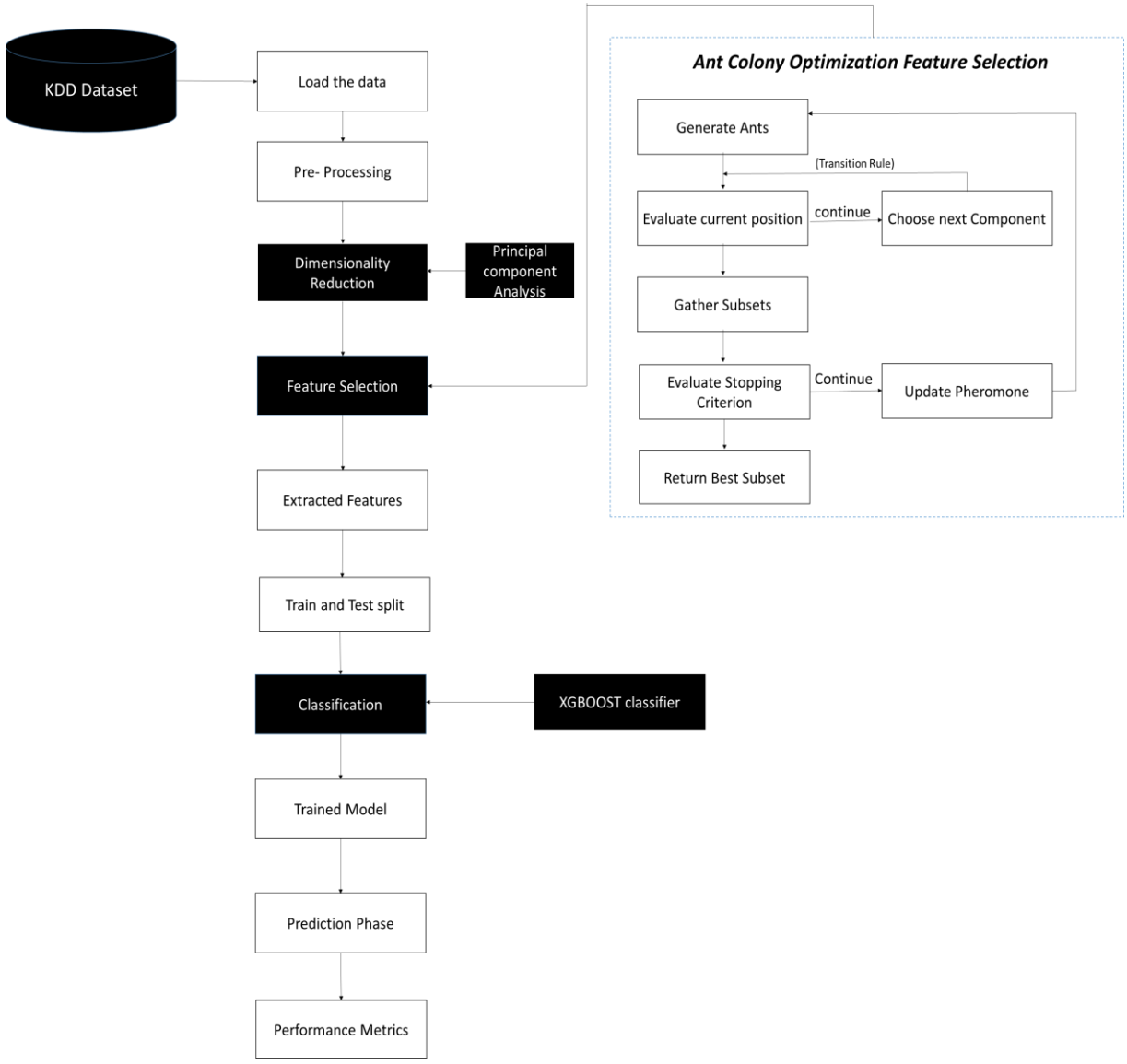

Figure 1.
Overall workflow of the proposed system.

3.1. Data-Set Description

The projected structure used the NSL KDD benchmark dataset [41] to estimate the proposed framework. The NSL-KDD is an upgraded form of the KDD'99 dataset. This dataset is applied as the efficient benchmark dataset that supports researchers in comparing various intrusion detection approaches. Moreover, good records are present in the test and training sets. The reason for using NSL-KDD datasets is that it does not embrace redundant data in the training set; therefore, the classifier may not be biased towards more frequent records. There are no identical records in the introduced testing set; thus, the learner's performance was not prejudiced by the methodology, which has improved detection rates on the frequent records. The NSL-KDD dataset consists of 22 types of cyber-attacks, which are divided into four classes: probing attack (PROBE), Denial of Service (DoS), User to Root (U2R), and Root to Local (R2L). The NSL-KDD consists of three nominal values such as User Datagram Protocol (UDP), Internet Control Message Protocol (ICMP), and Transmission Control Protocol (TCP). Figure 2 shows the overall flags present in the TCP, UDP and ICMP nominal of the NSL-KDD dataset.

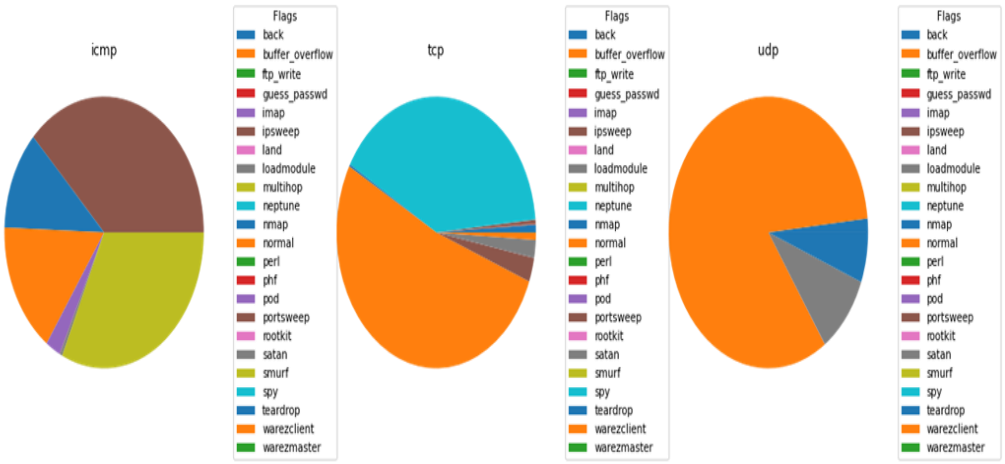

Figure 2.
Total flags present in the ICMP, TCP and UDP nominal of the NSL-KDD dataset.

3.2. Data Pre-Processing

The proposed system used two methods for the data pre-processing, i.e., 1-To-N Encoding and Normalization.

1 to N Encoding

For efficient feature selection, the NSL-KDD dataset cannot be employed directly in training since there is a presence of non-numeric features. Therefore, to overcome this issue, the non-numeric features are transformed into numeric features using $1-n$ numeric coding. In this system, all the non-numeric features like service, ag, and protocol are converted into numeric features.

3.2.1. Normalization

There are features in the NSL-KDD such as DST-bytes, src-bytes, or duration, which make the dataset unbalanced. Unrivaled records in the dataset mislead the classifier and result in an inaccurate outcome. Thus, these features or values are normalized by the following equation functions in Equation 1.

$$\frac{x - Min}{Max - Min} \tag{1}$$

The maximum and minimum values from all available data points are represented as max and min, while x denotes each data point.

3.3. Dimensionality Reduction

The dimensionality reduction is performed to reduce the data dimensions so that the feature selection process can be enhanced. Therefore, the proposed system uses principal component analysis for the dimensionality reduction process.

3.3.1. Principal Component Analysis

Principal component analysis is a well-known statistical method often used in the dimensionality reduction of datasets. The dataset can be denoted as a DM_{n*m} Data matrix, in that n, represents the number of objects with m variables or properties. Accordingly, DM_{n*m} represents the time series of a dataset that has m time series of n length; each row denotes a pattern of observed values for a special time, and the time series is represented in the column.

PCA is considered an orthogonal-linear transformation. It transforms an existing dataset into a new system. The highest variance of data values in the novel system through any projection of the data entities lies on the initial coordinate labelled as the first principal element, then the second modification on the second principal component (second coordinate). In that manner, principal component analysis transforms the DM data-matrix of $n * m$ size into another reduced form of a matrix Y with $n * k$ size for dimension reduction, where k is less than m.

Formally, Y is the reduced form of the dataset with principal components (variables) k, which are orthogonal to each other and $Y_{n*k} = DM_{n*m}V_{m*k}$ in that V_{m*k} are the composed form of the first variables k, and DM is the zero empirical mean value. Accordingly, SVD, $\Sigma = V\Lambda V^{-1}$ where $DMThe$ covariance matrix is Σ that is $\Sigma = DM^T DM$ because it has zero empirical mean value. Therefore, Λ is a rectangular diagonal-matrix m*m through non-negative real numbers on the diagonal, and the real numbers were composed of Σ eigenvalues in descending order. Therefore, the corresponding Σ Eigenvalues is V. However, concerning the energy content for apiece eigenvector, the initial kPrincipal components are selected from the first eigenvectors k of which the content of cumulative energy is not less than a $\in$ threshold. The algorithm of PCA is stated as follows,

Algorithm 1: Principal component analysis

$Step1.\ Organize\ the\ data\ matrix\ (ordataset)\ DM_{n*m}\ .Each\ row\ represents\ as\ an\ observation\ with\ variables\ m\ \&\ each\ column\ represents\ an\ n\ variable.$

$$Step2.\ Calculate\ the\ empirical\ mean\ and\ convert\ the\ data\ matrix\ into\ a\ new\ one\ with\ zero\ mean.\ simultaneously, the\ new\ one\ is\ replaced\ with\ original\ dataset.\ i.e.,\ DM = DM - TV, where\ V(1,i) = \frac{1}{n}\sum_{j=1}^{n} DM(j,i)\ \ and\ i = 1,2,\ldots,m\ and\ T\ is\ a\ n * 1\ column\ vector.$$

$$Step\ 3.\ Compute\ the\ eigenvalues\ and\ eigenvectors\ of\ the\ covariance\ matrix\ \Sigma = DM^T DM.\ According\ to\ SVD, \Sigma = V\Lambda V^{-1}, where\ \Lambda\ is\ the\ diagonal\ matrix\ of\ eigenvalues\ of\ \Sigma\ in\ descending\ order.\ While\ the\ eigenvector\ matrix\ V\ sort\ according\ to\ the\ decreasing\ order\ of\ eigenvalues, that\ is\ \lambda_1 \preccurlyeq \lambda_2 \preccurlyeq \cdots \lambda_m$$

$$Step\ 4.\ Choose\ a\ \ eigenvectors\ subset\ as\ basis\ vectors\ according\ to\ the\ content\ of\ \ cumulative\ energy.\ If\ the\ cumulative\ energy\ content\ h_k\ = \sum_{i=1}^{k} \lambda_i\ \ is\ above\ a\ certain\ value\ \varepsilon\ then\ the\ first\ \ eigenvectorsk\ are\ chosen\ as\ the\ basis\ vectors.\ In\ other\ term, if\ the\ contribution\ rate\ \boldsymbol{\eta} = \frac{\boldsymbol{h_k}}{\boldsymbol{h_m}}\ of\ cumulative\ energy\ is\ larger\ than\ a\ threshold\ \ \in, then\ the\ first\ eigenvectors\ k\ can\ be\ seen\ as\ the\ best\ principal\ components.$$

$$Step\ 5.\ Project\ the\ original\ dataset\ into\ the\ new\ system.\ The\ new\ dataset\ Y\ with\ low\ dimension\ can\ be\ formed\ Y_{n*k} = \ DM_{n*m}V_{m*k}.\ Since\ \ k\ is\ often\ less\ than\ m, that\ is\ k < m, PCA\ attains\ the\ dimensionality\ reduction.$$

From this, the dimension of the datasets is reduced efficiently.

3.4. Feature Selection

Feature selection is the most important process in designing intrusion detection systems. Feature selection involves extracting relevant and efficient features by removing the noisy features in the dataset. Therefore, to enhance the classification process, the proposed system performed a feature selection process using the ACO algorithm. Figure 3 shows the framework of the ant colony optimization.

3.4.1. Ant Colony Optimization (ACO)

In the early 90s, an algorithm known as the Ant System, developed by Dorigo and colleagues, was identified as a novel nature-inspired metaheuristic model for resolving combinatorial optimization problems. Initially, the algorithm was employed to solve the problem of a traveling salesman. Recently, it was modified and extended to improve performance and was employed for other optimization issues. The improved versions of the Ant System include the Ant Colony System, AS-Rank System, and MAX-MIN Ant System. The ACO algorithm is essentially a method based on individuals that simulate natural ant behavior, including adaptation and cooperation mechanisms. The foraging behavior of real ants was the main inspiration for ACO. The ACO algorithm is based on a computational paradigm inspired by real ant colonies and the way they perform. The idea was to support various constructive computational ants. Each ant was directed to store the generated solution based on the outcomes of preceding experiments deposited in the ant-dynamic memory. The paradigm was founded on the statement shaped by ethologists about the standard used by ants to connect the shortest path information to food utilizing pheromone traces. Meanwhile, isolated ants transfer practically at random during exploration; an ant that comes across an earlier laid trail may perceive it and adopt it with a high probability of following it, thus reinforcing the trace with its pheromone.

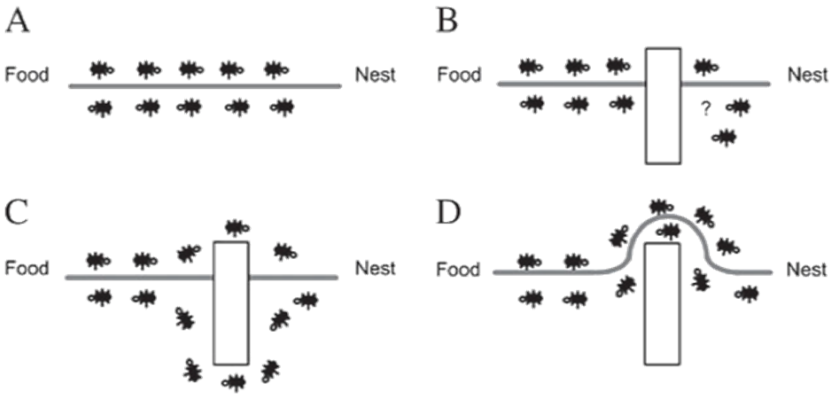

Figure 3.
Architecture of ant colony optimization.

3.4.2. Graph-Representation

The chief aim of the ACO is to optimize the minimum cost search path in a display. The nodes are measured features, with the edges among them referring to the next feature choice. The optimal-feature subset search is essentially an ant-traversal through a graph, where a minimum number of features and nodes are visited, which satisfies the traversal discontinuation condition. To allow any feature to select the next one, the nodes are fully connected. In graph representation reformulation, pheromone update rules and transition rules of standard ant colony optimization algorithms can be employed. Therefore, heuristic values and pheromone are not associated with links. As an alternative, each feature has its own heuristic and pheromone values.

3.4.3. Heuristic Information

The heuristic value representation is the feature attractiveness, and the rudimentary ingredient of any ant colony optimization algorithm is a constructive probabilistic solution. The solutions of constructive heuristics assemble as a feature order from the finite feature set. The construction of a subset twitches with an empty set. Then, at each phase of production, the present subset is protracted by adding the features commencing from the feature set. An appropriate traversing heuristic desirability among features might be any subset estimation. The traversal information of heuristics and nodes of pheromone levels are collected to process the probabilistic statute of transition, representing the possibility that may comprise feature in the subset at time phase:

$$P_i^{ant}(ts) = \begin{cases} \frac{[\tau_i(ts)]^{\alpha}.[\eta_i]^{\beta}}{\sum_{u \in J^{ant}}[\tau_u(ts)]^{\alpha}.[\eta_u]^{\beta}} & if\ i \in ff^{ant} \\ 0 & otherwise \end{cases} \quad (2)$$

Where $ff^{ant}$the feasible feature set is that ant can be added to its subset; η_i & τ_i are the heuristic and pheromone values associated with the feature β , α and i are the two parameters which conclude the pheromone and heuristic relative weight information. Therefore, the probability of transition used by ant colony optimization is a balance among τ_i the intensity of pheromones and η_i Heuristic-information. This competently equilibriums the manipulation to exploration tradeoff. The finest balance between exploitation & exploration is achieved by correct selection β , α constraints. When $\alpha = 0$, no-pheromone information is used; that is, the preceding exploration experience is ignored. The search formerly reduces into a stochastic greedy search if $\beta = 0$ neglects the latent benefits.

3.4.4. Pheromone Update

Updating the pheromone is a significant part of functioning the ACO suitably. After the completion of resolutions, pheromone vanishing on entire nodes is stimulated using Equation 3 quantity of pheromones are deposited by all ants, $\Delta\tau_i(ts)$, they have used on each node

$$T_i(ts) = (1-\rho)\tau_i(ts) \quad (3)$$

$$\tau_i(ts+1) = \tau_i(ts) + \Delta\tau_i(ts) \quad (4)$$

with

$$\Delta\tau_i(ts) = \sum_{k=1}^{m} \Delta\tau_i^{ant}(ts) \quad (5)$$

Where m is the amount of ants at each iteration, and the trail of pheromone decay coefficient is denoted as$p \in (0,1)$. Avoiding stagnation is the main role of pheromone evaporation, i.e., the circumstance in which all ants construct the same solution. According to Equations 4, 5, entire ants can inform the pheromone. Where the number of pheromones deposited by ant on i node at ts time step is calculated by $\Delta\tau_i(ts)$.

$$\Delta\tau_i^{ant}(ts) = \begin{cases} \frac{\omega.\gamma\left(S^{ant}(ts)\right) + \varphi.\left(\frac{n}{|S^{ant}(ts)|}\right) if\ i \in S^{ant}(ts)}{} & if\ i \in ff^{ant} \\ 0 \qquad otherwise \end{cases} \quad (6)$$

Where S^{ant} is the subset of features found by ant at t interaction, and its length is$|S^{ant}(ts)|$. According to the evaluation of classifier performance $\gamma\left(S^{ant}(ts)\right)$, the performance is updated, and the feature subset length. The parameters ω and φ, which control the relative significance of performance of classifier and the subset length, $\varphi = 1-\omega$ and $\omega \in [0,1]$.This formulation determines that the classifier performance and the feature subset length has different significance for the feature selection process. The system assumed that the classifier's performance is more important than the length of the subset, so $\omega = 0.7$ & $\varphi = 0.3$.

Therefore, the algorithm of ant-colony-optimization founded on FS (feature selection) for intrusion detection system is given as follows,

Algorithm 2: Ant colony optimization

1. $Begin$
2. Initialize all parameters, i. e. m, β, α, ρ, , τ_0, φ, ω, T

3. Let t $= 1$.
4. for Each node i do
5. $\tau_i(ts) = \tau_0$
6. $end\ for$
7. $Place\ m\ ants, ant = 1,,m.//\ Initialize\ a\ population\ of\ ants\ with\ random\ positions$
8. $while\ ts \leq T\ do$
9. $for\ Each\ ant = 1,\ldots,m\ do$
10. $S^k\ (ts) = \{\}$
11. $while\ Ant\ is\ able\ to\ increase\ the\ detection\ rate$
12. do
13. $From\ current\ node, select\ next\ node\ i\ using$
14. $Equation\ (2)$
15. $Add\ node\ i\ to\ subset\ S_k(ts)$
16. $end\ while$
17. $Calculate\ the\ subset\ length\ |S_k\ (ts)|.$
18. $Calculate\ the\ classifier\ performance\ \gamma\big(S_k(ts)\big)$
19. $end\ for$
20. $for\ Each\ node\ i\ do$
21. $Apply\ pheromone\ evaporation\ using\ Equation\ (3).$
22. $Calculate\ \Delta\tau i(ts) using\ Equations\ (5,6)$
23. $Update\ pheromone\ using\ Equation\ (4)$
24. $end\ for$
25. $ts = ts + 1$
26. end while
27. $Return\ the\ subset\ S_{ant}\ (ts)\ with\ highest\ \gamma(S_{ant}\ (ts))\ as\ the\ solution.$
28. End

By this feature selection is accomplished efficiently.

3.5. Classification

The proposed system enhances the classification performance using the XG-Boost classifier.

3.5.1. XgBoost Classifier

The XGBoost is an open-source software that facilitates a regularized gradient boosting architecture for Python, Java, C++, Perl, Scala, and Julia. It mainly aims to provide a distributed, portable, and scalable gradient boosting library. The XGBoost classifier works as Newton-Raphson in the space function. Unlike gradient boosting, which operates as gradient descent in the space of the Newton-Raphson model. Therefore, the XGBoost classifier algorithm is given as,

A generic un-regularized XG-Boost algorithm is:

$\boldsymbol{Input}$: $training\ set\ \ a\ differentiable\ loss\ function\ LF\big(y, F(x)\big), a\ number\ of\ weak\ learners\ M\ and\ a\ learning\ rate\ \ \alpha.$

Algorithm 3: XG-Boost classifier

Step 1. Model initialization with a constant value

$$f_{(0)}(x) = \arg\min \sum_{i=1}^{N} LF\ (y_i, \emptyset)$$

Step 2. For m=1 to WL

1. Computes the "gradients $g_m(x_i)$" and "hessians $h_m(x_i)$ "

$$g_m(x_i) = \left[\frac{\partial LF(y_i f(x_i))}{\partial f(x_i)}\right]_{f(x)=\hat{f}_{(m-1)}(x)}$$

$$h_m(x_i) = \left[\frac{\partial^2 LF(y_i f(x_i))}{\partial f(x_i)^2}\right]_{f(x)=\hat{f}_{(m-1)}(x)}$$

2. Using a training set fits a base learner or weak learner

$$\left\{x_i, -\frac{g_m(x_i)}{h_m(x_i)}\right\}_{i=1}^{N}$$

$$\partial_m = \arg\min \sum_{i=1}^{N} \frac{1}{2} h_m(x_i) \left[-\frac{g_m(x_i)}{h_m(x_i)} - \emptyset(x_i) \right]^2$$

$$f_m(x) = \alpha \partial_m \, (m)$$

3. Model update

$$f_m(x) = f_{(m-1)}(x) + f_{(m)}(x)$$

Step 3. Final output

$$f_m(x) = f_{(M)}(x) = \sum_{m=0}^{WL} f_m(x)$$

Therefore, from this, the proposed system attained a high classification rate in the intrusion detection system. Section 3 deliberates the overall workings of the research methodology with a flow graph. Section 3.1 describes the dataset. Then, the pre-processing is stated in section 3.2; the approach used in the pre-processing, i.e., the 1 to N encoding, is stated in section 3.2.1, and normalization is stated in section 3.2.2. Then, the dimensionality reduction is explained in section 3.3, and the principal component analysis is explained with the proposed algorithm in section 3.3.1. Section 3.4 explains the feature selection process, and the ant-colony optimization is explained in detail with the planned algorithm in section 3.4.1. Finally, the classification process is explained in section 3.5; along with this, the XG-Boost classifier is explained in section 3.5.1. The implemented system results are explained in the upcoming section 4.

4. Results and Discussion

This section deliberates the proposed system's performance by analyzing the obtained results. The evaluation of the proposed system is conducted with the NSL-KDD dataset concerning the performance metrics such as precision, accuracy, F1-score, specificity, and recall.

4.1. Performance Metrics

Concerning the performance metrics, the performance of the planned system is determined. The considered performance metrics are precision, accuracy, recall, specificity, and F1 score.

4.1.1. Accuracy

Accuracy denoted the fraction of correctly predicted results from the total samples. The below-shown equation was used for calculating accuracy.

$$Accuracy = \frac{no.of.true\ negative + no.of.true\ positive}{no.of.true\ negative + no.of.false\ positive + no.of.true\ positive + no.of.false\ negative} \quad (7)$$

4.1.2. Precision

It is defined as the proportion of tuple through which the exact prediction can be made for the presence of intrusion, and it is premeditated by the below-shown equation.

$$Precision = \frac{(TP)}{(TP + FP)} \quad (8)$$

In the above-shown representation, TP was termed a true positive, and FP was termed a false positive.

4.1.3. Recall

The recall is the fraction of tuples, where the intrusion is correctly detected or not. Its calculation is done in the equation mentioned below.

$$Recall = \frac{TP}{TP + FN} \quad (9)$$

Similarly, in the above-represented equation, TP was signified as a true positive, and FN was signified as a false negative.

4.1.4. F1-Score

It is defined by the harmonic mean of precision and recall. The below-mentioned equation does the computation.

$$F1 - Score = \frac{2 * precision * recall}{precision + recall} \quad (10)$$

4.1.5. Specificity

Specificity was the metrics which estimated the model's ability to predict the true negatives of each available category. The mathematical representation of specificity is shown below.

$$specificity = \frac{No.Of.True\ negatives}{No.Of.false\ positives + No.Of.True\ negatives} \quad (11)$$

According to these metrics, the evaluation of the proposed system is performed.

4.2. Performance Analysis

The evaluation of the proposed system's performance is discussed in this section. The evaluation is performed concerning the accuracy and the number of features extracted. The total number of features involved in the proposed system is 44 columns (features).

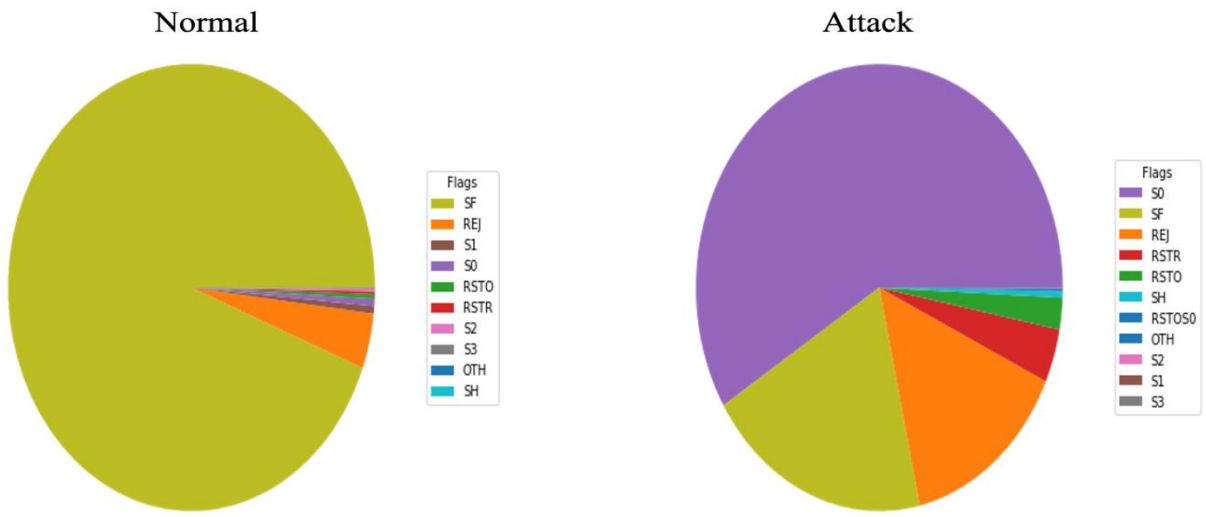

Figure 4.
Flags present in normal and the flags present in the attack.

4.3. Performance of PCA

The performance results obtained with the PCA are deliberated in this section.

Table 1.
Dimensionality reduction process.

	With PCA	Without PCA
Number of Columns (Features)	40	43

Table 1 shows the dimensionality reduction process performed with and without PCA. The total number of features used is 44, but the PCA-involved system showed that the features were reduced from 44 to 40. The process performed without PCA showed that the features diminished were reduced from 44 to 43. The dimensionality reduction reduces the dimension of features. Therefore, the system performed with PCA showed better results in dimensionality reduction by reducing the features from 44 to 40. It showed that the implementation of PCA in dimensionality reduction yielded better results than the system performed without PCA.

4.3.1. Performance of ACO

The results of the feature selection process with ant colony optimization are deliberated in these sections.

Table 2.
Selected Features.

Number of features Selected with ACO
1,3,5,6,8,10,11,15,18,23,26,28,29,32,36,39

The dimension reduced features, i.e., the 40 columns, were then processed in the feature selection, which selected only the relevant features. Therefore, the selected features are represented in Table 2.

Table 3.
Feature selection with and without ACO.

Accuracy with ACO (%)	Accuracy without ACO (%)
97.8	95.36

Table 3 represents the accuracy obtained with and without the involvement of ACO. The system performed with the ACO feature selection showed 97.8% accuracy, whereas the system performed without the involvement of ACO showed 95.36% accuracy. Therefore, the system performed with ACO feature selection showed 97.8% accuracy, which demonstrated better results than the system performed without ACO. Hence, it proves that the feature selection with ACO showed better results.

4.4. Comparative Analysis

The overall performance of the projected system concerning recall, accuracy, precision, specificity, and F1-score is compared with various existing methods. The comparative results are stated in this section.

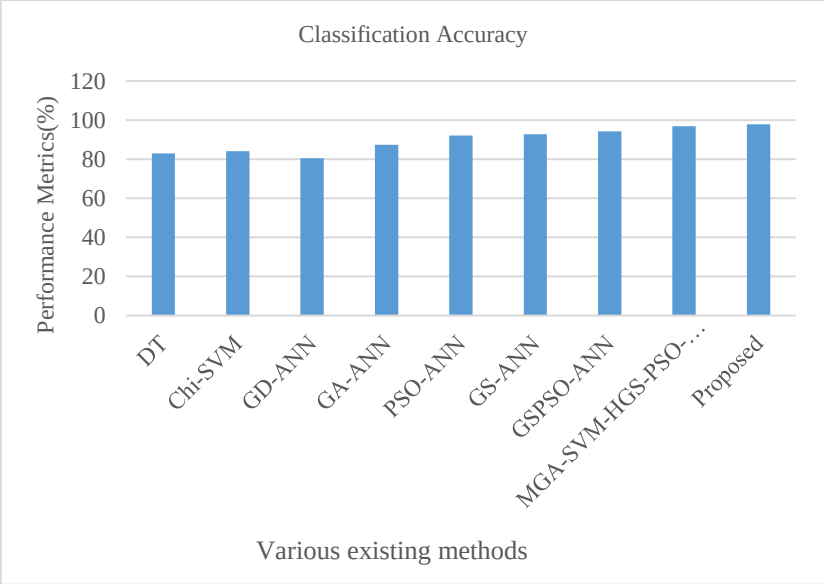

Figure 5.
Comparison of classification accuracy with earlier methods.

Figure 5 deliberates the results of classification accuracy compared with various existing methods. The DT showed 83% accuracy, the chi-SVM model showed 84% accuracy, GD-ANN showed 80.45% accuracy, GA-ANN showed 87.37% accuracy, PSO-ANN showed 92.06% accuracy, GS-ANN showed 92.81% accuracy, GSPSO-ANN showed 94.26% accuracy, and MGA-SVM-HGS showed 96.8% accuracy, while the proposed system showed 97.80% accuracy. This comparison shows that the proposed XG-Boost results in a 97.80% accuracy rate, which is higher than the other existing methods. It demonstrates that the classification shows its efficiency for intrusion detection.

Further, the comparative analysis with earlier methods concerning precision, F1-score, accuracy, and recall is stated in the following section. The XGBoost-DNN system showed 97.6% accuracy, 97% precision, 97% recall, and 97% F1-score. Then, the LR system showed 87% accuracy, 87% precision, 87% recall, and 87% F1-score. The NB system showed 52% accuracy, 28% precision, 52% recall, and 36% F1-score. The SVM system showed 90% accuracy, 90% precision, 90% recall, and 90% F1-score. Therefore, the present system showed 97.8% accuracy, 97.6% precision, 97.56% recall, and 97.88% F1-score. The proposed system showed improved results compared to the existing methods. Thus, it proves that the projected system is efficient and effective in intrusion detection. Figure 6 shows the graphical representation of the comparative analysis.

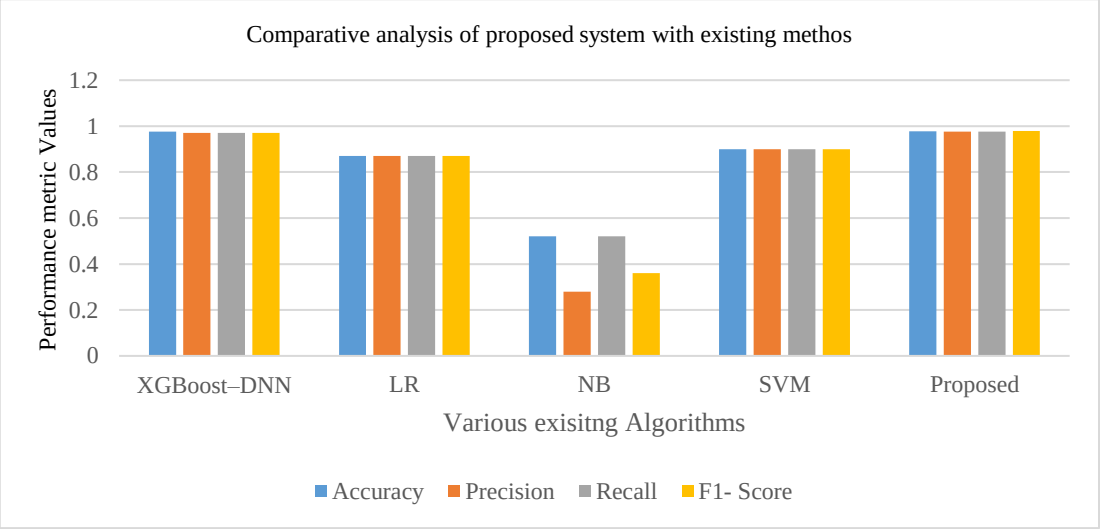

Figure 6.
Comparative analysis of the proposed system with various existing algorithms.

Figure 7 shows the specificity and accuracy of several methods employed for intrusion detection on the NSL-KDD dataset.

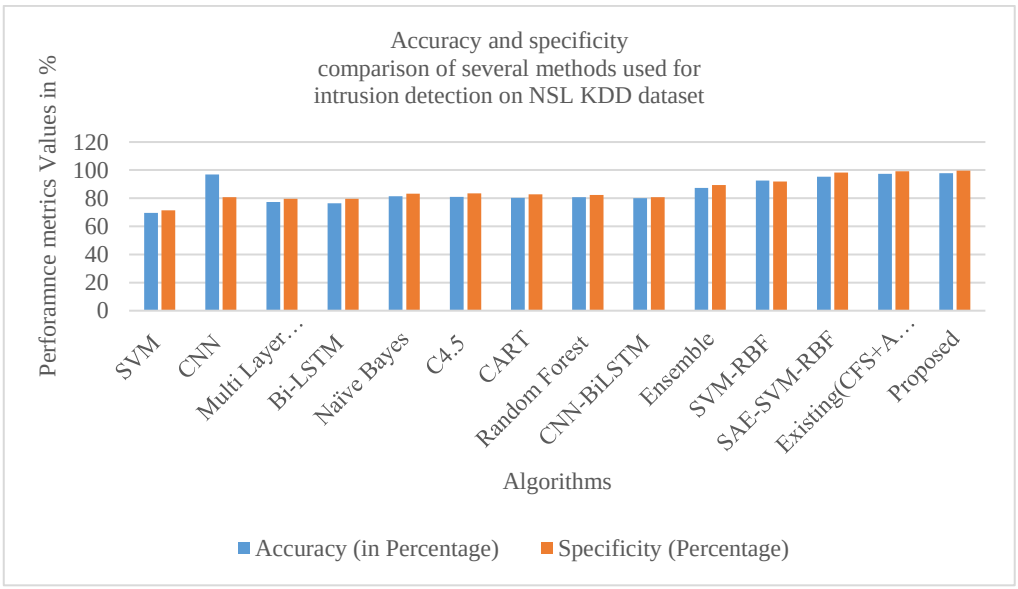

Figure 7.
Specificity and Accuracy comparison of several systems supports intrusion detection on the NSL-KDD dataset.

The comparative analysis showed the results obtained from various existing methods with respect to the accuracy and specificity. The accuracy of SVM is 69.52%, 97.01% in CNN, 77.41% in multi-layer perceptron, 76.37% in Bi-LSTM, 81.47% in Naïve Bayes, 81% in C4.5, 80.3% in CART, 80.67% in Random Forest, 80.05% in CNN-BiLSTM, 87.28% in ensemble, 92.55% in SVM-RBF, 95.27% SAE-SVM-RBF, 97.49% in CFS+ANN and the proposed system showed 97.80%. Similarly, the specificity result showed in the systems are, 71.41% in SVM, 80.75% in CNN, 79.57% in Multi-layer perceptron, 79.64% in Bi-LSTM, 83.21% in Naïve Bayes, 83.44% in C4.5, 82.71% in CART, 82.35% in random forest, 80.83% in CNN-BiLSTM, 89.41% in Ensemble, 91.84% in SVM-RBF, 98.35% in SAE—SVM-RBF, 99.31% in CFS+ANN and the proposed system showed 99.64% of specificity rate. Therefore, this comparative analysis clearly stated that the proposed system showed high rate of accuracy and specificity than the various existing methods.

Table 4.
Comparison Chart

Model	Accuracy	Precision	Recall	F-1 Score
XGBoost-DNN	97.6%	97%	97%	97%
LR	87%	87%	87%	87%
NB	52%	28%	52%	36%
SVM	90%	90%	90%	90%
Proposed	99.2%	99.6%	98.8%	99.2%

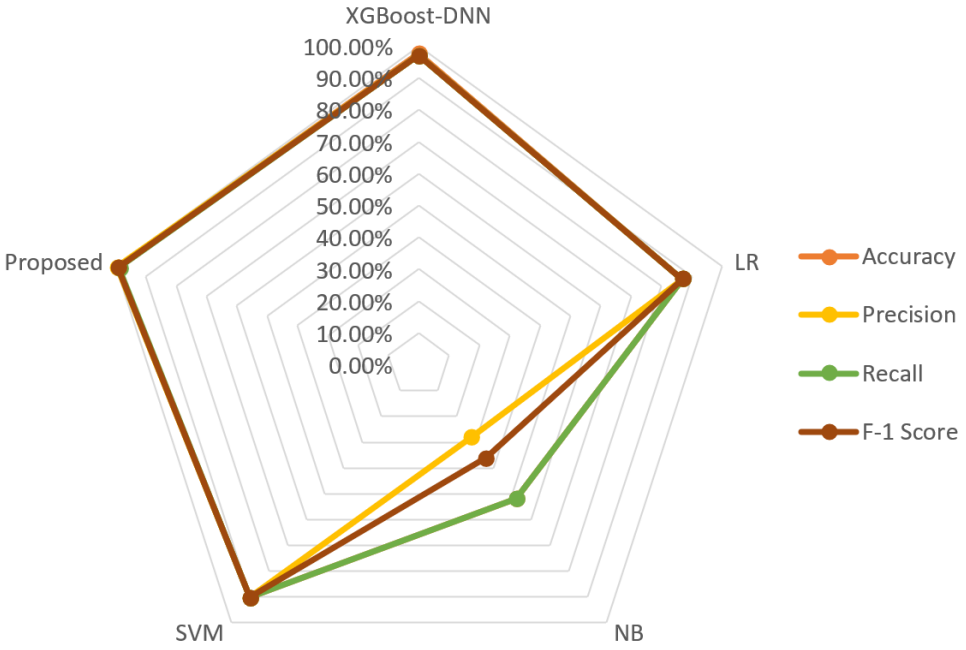

Figure 8.
Radar Chart to compare performance metrics of proposed work with other state of art methods.

The proposed system used the NSL-KDD dataset and performed the preprocessing. The dimensionality reduction is performed efficiently with PCA. The feature selection is enhanced with ACO, and the classification is performed with the XGBoost algorithm. The performance was estimated based on performance metrics such as precision, F1-score, recall, and accuracy. Also, the efficiency of each system is stated in the performance analysis. Then, a comparative analysis was performed to determine the efficiency and effectiveness of the proposed system. The proposed system showed 97.8% accuracy, 97.6% precision, 97.56% recall, 97.88% F1-score, and 99.64% specificity. The obtained results were better than the existing methods. The overall performance and the individual system performance also showed their efficiency in enhancing the prediction result. Thus, it proves that the present system showed its competence and effectiveness in intrusion detection. Therefore, the proposed system shows improved results compared to various existing methods.

5. Conclusion

The growth of technology also leads to the risk of intrusion and misuse of information. Therefore, to overcome these problems and to protect the information, the intrusion detection system was employed. The system proposed an effective intrusion detection approach with ant colony optimization based on feature selection and an XGBoost classifier to enhance the system's security. The paper provides a brief introduction to the intrusion detection system. Then, the various existing systems in intrusion detection were reviewed to find the strengths and weaknesses of the proposed system; the weaknesses are revealed in the proposed system to improve its efficiency. Following this, the proposed methodology was explained with the overall working flow graph. Then, the introduction to the NSL-KDD dataset was given along with the dataset pre-processing. The dimensionality reduction performed with principal component analysis was explained with the proposed algorithm, and then to enhance the classification process, feature selection was performed with the ant colony optimization, which was explained in detail with the proposed algorithm. Then, the classification performed with the XGBoost classifier algorithm was also explained. Therefore, to conclude the effectiveness of the proposed system, the performance evaluation was done using performance metrics such as F1-score, precision, specificity, accuracy, and recall. Each employed model's efficiency was also explained in the performance evaluation. Along with this, a comparative investigation was performed to conclude the proficiency and effectiveness of the proposed system. Thus, the proposed system showed 97.8% accuracy, 97.6% precision, 97.56% recall, 97.88% F1-score, and 99.64% specificity, which are better than the existing intrusion detection systems. Hence, it proves that the proposed system was efficient and effective in intrusion detection.

References

[1] H. Liu and B. Lang, "Machine learning and deep learning methods for intrusion detection systems: A survey," *Applied Sciences*, vol. 9, no. 20, p. 4396, 2019. https://doi.org/10.3390/app9204396

[2] A. Aldweesh, A. Derhab, and A. Z. Emam, "Deep learning approaches for anomaly-based intrusion detection systems: A survey, taxonomy, and open issues," *Knowledge-Based Systems*, vol. 189, p. 105124, 2020.

[3] S. Rawat, A. Srinivasan, V. Ravi, and U. Ghosh, "Intrusion detection systems using classical machine learning techniques vs integrated unsupervised feature learning and deep neural network," *Internet Technology Letters*, vol. 5, no. 1, p. e232, 2022. https://doi.org/10.1002/itl2.232

[4] P. Bedi, N. Gupta, and V. Jindal, "Handling class imbalance problem in intrusion detection systems using siamese neural network," *Procedia Computer Science*, vol. 171, pp. 780-789, 2024. https://doi:10.1016/j.procs.2020.04.107

[5] M. Artur, "Review the performance of the Bernoulli Naïve Bayes classifier in intrusion detection systems using recursive feature elimination with cross-validated selection of the best number of features," *Procedia Computer Science*, vol. 190, pp. 564-570, 2021.
[6] J. K. Samriya, R. Tiwari, X. Cheng, R. K. Singh, A. Shankar, and M. Kumar, "Network intrusion detection using ACO-DNN model with DVFS based energy optimization in cloud framework," *Sustainable Computing: Informatics and Systems*, vol. 57, p. 100746, 2022. https://doi.org/10.1016/j.suscom.2022.100746
[7] D. Preethi and N. Khare, "Sparse auto encoder driven support vector regression based deep learning model for predicting network intrusions," *Peer-to-Peer Networking and Applications*, vol. 14, no. 4, pp. 2419-2429, 2021.
[8] F. Laghrissi, S. Douzi, K. Douzi, and B. Hssina, "IDS-attention: An efficient algorithm for intrusion detection systems using attention mechanism," *Journal of Big Data*, vol. 8, no. 1, pp. 1-21, 2021.
[9] Y. Hamid and M. Sugumaran, "A t-SNE based non linear dimension reduction for network intrusion detection," *International Journal of Information Technology*, vol. 12, no. 1, pp. 125-134, 2020.
[10] H. Rajadurai and U. D. Gandhi, "An empirical model in intrusion detection systems using principal component analysis and deep learning models," *Computational Intelligence*, vol. 37, no. 3, pp. 1111-1124, 2021.
[11] H. Bangui and B. Buhnova, "Lightweight intrusion detection for edge computing networks using deep forest and bio-inspired algorithms," *Computers and Electrical Engineering*, vol. 100, p. 107901, 2022.
[12] A. Thakkar and R. Lohiya, "Role of swarm and evolutionary algorithms for intrusion detection system: A survey," *Swarm and Evolutionary Computation*, vol. 53, p. 100631, 2020.
[13] Z. Ashi, L. Aburashed, M. Al-Qudah, and A. Qusef, "Network intrusion detection systems using supervised machine learning classification and dimensionality reduction techniques: A systematic review," *Jordanian Journal of Computers and Information Technology*, vol. 7, no. 04, pp. 373–390, 2021. https://doi.org/10.5455/jjcit.71-1629527707
[14] D. Raman, G. V. Reddy, A. Kumar, and S. Vuyyala, "Efficient machine learning model for intrusion detection—a comparative study," in *Machine Learning and Information Processing: Proceedings of ICMLIP 2020*, 2021: Springer, pp. 435-444.
[15] R. D. Ravipati and M. Abualkibash, "Intrusion detection system classification using different machine learning algorithms on KDD-99 and NSL-KDD datasets-a review paper," *International Journal of Computer Science & Information Technology*, vol. 11, no. 3, pp. 45–54, 2019.
[16] X. Gao, C. Shan, C. Hu, Z. Niu, and Z. Liu, "An adaptive ensemble machine learning model for intrusion detection," *IEEE Access*, vol. 7, pp. 82512-82521, 2019.
[17] Ü. Çavuşoğlu, "A new hybrid approach for intrusion detection using machine learning methods," *Applied Intelligence*, vol. 49, pp. 2735-2761, 2019.
[18] A. Thakkar and R. Lohiya, "A review of the advancement in intrusion detection datasets," *Procedia Computer Science*, vol. 167, pp. 636-645, 2020.
[19] F. Yihunie, E. Abdelfattah, and A. Regmi, "Applying machine learning to anomaly-based intrusion detection systems," presented at the 2019 IEEE Long Island Systems, Applications and Technology Conference (LISAT), 2019.
[20] K. Pradeep Mohan Kumar, M. Saravanan, M. Thenmozhi, and K. Vijayakumar, "Intrusion detection system based on GA-fuzzy classifier for detecting malicious attacks," *Concurrency and Computation: Practice and Experience*, vol. 33, no. 3, p. e5242, 2021.
[21] M. Di Mauro, G. Galatro, G. Fortino, and A. Liotta, "Supervised feature selection techniques in network intrusion detection: A critical review," *Engineering Applications of Artificial Intelligence*, vol. 101, p. 104216, 2021.
[22] R. Abdulhammed, H. Musafer, A. Alessa, M. Faezipour, and A. Abuzneid, "Features dimensionality reduction approaches for machine learning based network intrusion detection," *Electronics*, vol. 8, no. 3, p. 322, 2019. https://doi.org/10.3390/electronics8030322
[23] S. Bhattacharya *et al.*, "A novel PCA-firefly based XGBoost classification model for intrusion detection in networks using GPU," *Electronics*, vol. 9, no. 2, p. 219, 2020. https://doi.org/10.3390/electronics9020219
[24] Z. Shen, Y. Zhang, and W. Chen, "A bayesian classification intrusion detection method based on the fusion of PCA and LDA," *Security and Communication Networks*, vol. 2019, no. 1, p. 6346708, 2019.
[25] B. Zhang, Z. Liu, Y. Jia, J. Ren, and X. Zhao, "Network intrusion detection method based on PCA and Bayes algorithm," *Security and Communication Networks*, vol. 2018, no. 1, p. 1914980, 2018.
[26] F. Laghrissi, S. Douzi, K. Douzi, and B. Hssina, "Intrusion detection systems using long short-term memory (LSTM)," *Journal of Big Data*, vol. 8, no. 1, pp. 1-16, 2021.
[27] B. N. Kumar, R. MSVSB, and B. V. Vardhan, "Enhancing the performance of an intrusion detection system through multi-linear dimensionality reduction and Multi-class SVM," *International Journal of Intelligent Engineering and Systems*, vol. 11, no. 1, pp. 181-192, 2018.
[28] V. Balasaraswathi and M. Sugumaran, "A hybrid algorithm using ant colony optimization and cuttlefish algorithm for feature selection in intrusion detection systems," in *Proceedings of the 2019 International Conference on Computer Networks and Communication Technologies (CNCT)*, 2009.
[29] B. A. Tama, M. Comuzzi, and K.-H. Rhee, "TSE-IDS: A two-stage classifier ensemble for intelligent anomaly-based intrusion detection system," *IEEE Access*, vol. 7, pp. 94497-94507, 2019.
[30] C. Kalimuthan and J. A. Renjit, "Review on intrusion detection using feature selection with machine learning techniques," *Materials Today: Proceedings*, vol. 33, pp. 3794-3802, 2020.
[31] H. Peng, C. Ying, S. Tan, B. Hu, and Z. Sun, "An improved feature selection algorithm based on ant colony optimization," *IEEE Access*, vol. 6, pp. 69203-69209, 2018.
[32] S. S. Dhaliwal, A.-A. Nahid, and R. Abbas, "Effective intrusion detection system using XGBoost," *Information*, vol. 9, no. 7, p. 149, 2018. https://doi.org/10.3390/info9070149
[33] B. S. Bhati, G. Chugh, F. Al-Turjman, and N. S. Bhati, "An improved ensemble based intrusion detection technique using XGBoost," *Transactions on Emerging Telecommunications Technologies*, vol. 32, no. 6, p. e4076, 2021. https://doi.org/10.1002/ett.4076
[34] H. Jiang, Z. He, G. Ye, and H. Zhang, "Network intrusion detection based on PSO-XGBoost model," *IEEE Access*, vol. 8, pp. 58392-58401, 2020.

[35] P. Bedi, N. Gupta, and V. Jindal, "I-SiamIDS: An improved Siam-IDS for handling class imbalance in network-based intrusion detection systems," *Applied Intelligence*, vol. 51, no. 2, pp. 1133-1151, 2021.

[36] I. F. Kilincer, F. Ertam, and A. Sengur, "A comprehensive intrusion detection framework using boosting algorithms," *Computers & Electrical Engineering*, vol. 100, p. 107869, 2022.

[37] A. O. Alzahrani and M. J. Alenazi, "Designing a network intrusion detection system based on machine learning for software defined networks," *Future Internet*, vol. 13, no. 5, p. 111, 2021. https://doi.org/10.3390/fi13050111

[38] Y. Song, H. Li, P. Xu, and D. Liu, "A method of intrusion detection based on WOA-XGBoost algorithm," *Discrete Dynamics in Nature and Society*, vol. 2022, no. 1, p. 5245622, 2022.

[39] W. Xu and Y. Fan, "Intrusion detection systems based on logarithmic autoencoder and XGBoost," *Security and Communication Networks*, vol. 2022, no. 1, p. 9068724, 2022.

[40] Y. Niu, C. Chen, X. Zhang, X. Zhou, and H. Liu, "Application of a new feature generation algorithm in intrusion detection system," *Wireless Communications and Mobile Computing*, vol. 2022, p. 1234567, 2022.

[41] UNB, "NSL-KDD dataset," Retrieved: https://www.unb.ca/cic/datasets/nsl.html, 2024.