Intrusion Detection Utilizing an Ant Colony Optimization-Based Feature Selection and the XGBoost Classifier

Shweta Bhardwaj
Department of Computer Science & Engineering, Amity University, Uttar Pradesh, Noida, India
sbhardwaj1@amity.edu

Seema Rawat
School of AI and Data Science, Astana IT University, Kazakhstan
seema.rawat@astanait.edu.kz (corresponding author)

Hima Bindu Maringanti
Department of Computer Science & Applications, MSCB University, Baripada, India
m.himabindu@odisha.gov.in

Received: 6 September 2025 | Revised: 15 October 2025 and 29 October 2025 | Accepted: 1 November 2025

 DOI: https://doi.org/10.48084/etasr.14572

ABSTRACT

The Internet of Things (IoT) continues to expand dramatically, connecting a growing number of smart devices such as home automation systems and wearables. However, this growth also introduces significant cybersecurity risks, as attackers increasingly exploit vulnerabilities in these interconnected devices. Protecting IoT networks requires comprehensive Intrusion Detection Systems (IDSs) that can intelligently identify and mitigate malicious activities. The proposed approach integrates dimensionality reduction through Principal Component Analysis (PCA) to streamline data, feature selection using Ant Colony Optimization (ACO) to identify relevant indicators, and classification through the Extreme Gradient Boosting (XGBoost) algorithm for accurate threat detection. The proposed approach achieved far superior results compared to existing IDS methods on three different datasets: 99.2% accuracy, 99.6% precision, 98.8% recall, and 99.2% F1- score on NSL-KDD, 99.3% accuracy, 92.8% precision, 99% recall and 95.8% F1-score on UNSW- NB15, and 99.9% accuracy, 99.5% precision, 99.8% recall, and 99.7% F1-score on CIC-IDS.

Keywords-internet of things; intrusion detection system; XGBoost; NSL-KDD; UNSW-NB15; CIC-IDS; ACO

I. INTRODUCTION

Cyber-attacks can lead to economic losses, sensitive data leaks, and commotion within critical systems [1]. As a result, this network security has become a dynamic and significant research area. Intrusion Detection Systems (IDSs) are important for monitoring hardware and software conditions within a network so that potential threats can be detected. The recent increase in frequency and scale of cyber-attacks highlights the crucial need for timely and effective threat detection. IDSs continuously monitor network traffic and system logs to identify unusual activities and patterns, which may highlight malicious behavior [2]. Many studies have explored the use of Machine Learning (ML) and Deep Learning (DL) methods to improve the accuracy and efficiency of IDSs in detecting such threats, leveraging advanced algorithms to recognize complicated and varied patterns in data and respond to evolving attacks.

IDSs can be classified into two main categories: signature-based and anomaly-based. ML techniques have shown great effectiveness in differentiating between normal and malicious data [3]. Various ML methods, including both supervised algorithms, such as Decision Trees (DT) and Support Vector Machines (SVM), as well as unsupervised methods, such as clustering and autoencoders, remain active areas of research in intrusion detection. These techniques have the advantage of learning over time with minimal human intervention, yet their effectiveness is driven by the quality of training data, feature selection strategies, and the specific algorithms employed [4].

Cyber-attacks pose critical challenges for both individuals, organizations, and governments, leading to economic losses, loss of sensitive information, and disruption of key systems [5]. An IDS serves as a vital component in network security by continuously monitoring the status of both hardware and software within networks to detect potential threats. Traditional security measures, such as firewalls and antivirus software, are

ineffective in detecting recent or unknown attacks, i.e., zero-day attacks or Advanced Persistent Threats (APTs). An IDS addresses this challenge by continuously tracking system traffic and recording anomalies and pattern characteristics of unlawful events [6]. The integration of Artificial Intelligence (AI), particularly through ML and DL, is rapidly becoming a promising area for improving IDSs [7]. Such approaches can learn from data continuously, allowing the system to adapt effectively to evolving cyber threats. Previous studies have highlighted the growing importance of these advanced methods, demonstrating how they can significantly improve the detection capabilities by employing sophisticated learning algorithms [8-9]. Several recent reviews have continued to examine the progress and development of these approaches within IDSs, providing detailed comparisons of various models and highlighting emerging trends and challenges in the field [10].

In [11], a detailed systematic review of DL-based IDSs emphasized the increasing significance of AI techniques in enhancing cybersecurity. This review carefully categorized and examined diverse DL methodologies, such as Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), Autoencoders, and Deep Belief Networks (DBNs), that are applied for intrusion detection in both host-centric and network-centric environments. The study in [12] emphasized that DL techniques significantly outperform traditional ML algorithms in handling multi-dimensional, complex, and chaotic network traffic data due to their automatic feature extraction and scalability. Among the models evaluated, CNNs and hybrid approaches achieved higher detection accuracy. In addition, a detailed systematic literature review on IDSs provided a comprehensive overview of research trends, prominent algorithms, methodologies, datasets, and inherent challenges within the field. Commonly used algorithms, such as SVM, DT, and K-Nearest Neighbors (KNN), and various deep Neural Network (NN) architectures were evaluated for their effectiveness based on accuracy, precision, and computational efficiency.

In [13], a detailed survey on the application of ML techniques in IDS development focused on architectural design factors, algorithm selection, and the challenges faced during deployment. This study offered a comprehensive taxonomy of ML methods utilized in IDSs, covering supervised, unsupervised, reinforcement learning, and ensemble approaches. Various algorithms were reviewed, including Random Forests (RF), Gradient Boosting (GB), SVM, and NNs, comparing their effectiveness based on detection accuracy, training speed, and generalizability. Particular emphasis was placed on ensemble and hybrid models, which combine multiple algorithms to optimize robustness and minimize false positive rates. In [14], a thorough and systematic review of recent advances in ML- and DL-based IDSs proposed a clear and structured taxonomy for cyberattack types, including traditional threats such as Denial of Service (DoS) and probing attacks, as well as APTs and insider threats [14]. This taxonomy provides a solid framework for evaluating and comparing the utility of various ML and DL techniques in intrusion detection.

II. METHODOLOGY

This section discusses the proposed model, which was evaluated on the NSL-KDD, UNSW-NB15, and CIC-IDS datasets. The method consists of the following components:

- Data Preprocessing: The raw dataset is carefully pre-processed to eliminate irrelevant information and convert categorical or non-numeric values into numeric formats. For the NSL-KDD and UNSW-NB15 datasets, 80% of the data was used for training and 20% for testing, while the CIC-IDS dataset was split into 70% for training and 30% for testing.
- Feature Selection: Ant Colony Optimization (ACO), inspired by natural ant behavior, is leveraged to select the most relevant features from the dataset, improving model reliability and reducing dimensionality.
- Dimensionality Reduction: Principal Component Analysis (PCA) is applied to further reduce the number of features by transforming them into principal components that capture most of the data variance.
- Classification: The XGBoost algorithm, known for its high performance and scalability, is utilized to classify network traffic as either Normal or Attack. The XGBoost algorithm was configured with Number of trees: 300, Maximum tree depth: 6, Learning rate: 0.1, Subsample ratio: 0.8, Column sampling by tree: 0.8, Regularization: L2 $(\lambda = 1)$, and Objective function: binary logistic.
- Performance Evaluation: The effectiveness of the proposed system is evaluated through multiple performance metrics to ensure reliable detection results.

In the baseline configuration (without PCA), dimensionality reduction was performed using a filter-based feature elimination approach. Initially, constant features were removed. Subsequently, highly correlated features were identified using correlation analysis, and a feature from each highly correlated pair was eliminated. This process reduced redundancy while preserving the original feature space.

Dimensionality reduction using PCA is performed to enhance the feature selection process. Table I shows the dimensionality reduction process applied to the NSL-KDD dataset. The table outlines the step-by-step elimination of redundancy in the data, ultimately retaining only the most relevant features for improving model efficiency and classification accuracy, showing the outcomes of the dimensionality reduction process established with and without the application of PCA. Using PCA, the number of features is reduced from 44 to 40, establishing a major drop in dimensionality. In contrast, the process implemented without PCA reduces the feature set from 44 to 43. This comparison highlights that incorporating PCA into the dimensionality reduction process is more effective.

TABLE I. DIMENSIONALITY REDUCTION IN NSL-KDD

	With PCA	**Without PCA**
Number of columns (Features)	40	43

Table II illustrates the results of dimensionality reduction with and without the application of PCA on the UNSW-NB15 dataset. After applying PCA, the number of features was effectively reduced to 25, showing that PCA plays a crucial role by filtering out less informative features. Consequently, using PCA improves efficiency in reducing dimensionality.

TABLE II. DIMENSIONALITY REDUCTION PROCESS IN UNSW-NB15

	With PCA	**Without PCA**
Number of columns (Features)	25	49

Table III illustrates the dimensionality reduction process, both with and without PCA, on the CIC-IDS dataset. The total number of features is 78, but PCA reduced it to 22. The results highlight that PCA is very important in making the dataset more refined. By decreasing the feature count, PCA reduces computational burden tremendously and identifies the most prominent components, reducing noise and concentrating the classifier on essential data points.

TABLE III. DIMENSIONALITY REDUCTION IN CIC-IDS

	With PCA	**Without PCA**
Number of columns (Features)	22	78

Feature selection extracts relevant and efficient features by removing the noise in the dataset. Therefore, to enhance the classification process, the proposed system performed a feature selection process with the ACO algorithm. The tables below show the respective features selected for each dataset.

TABLE IV. SELECTED FEATURES IN NSL-KDD

Features selected with ACO
1,3,5,6,8,10,11,15,18,23,26,28,29,32,36,39

TABLE V. SELECTED FEATURES IN UNSW-NB15

Features selected with ACO
0,1,5,7,9,10,11,12,20,24

TABLE VI. SELECTED FEATURES IN CIC-IDS

Features selected with ACO
3,4,5,6,9,11,13,14,15,16,18,20

These features were recognized as the most effective for differentiating between Normal and Attack traffic, helping the intrusion detection model achieve better performance and accuracy.

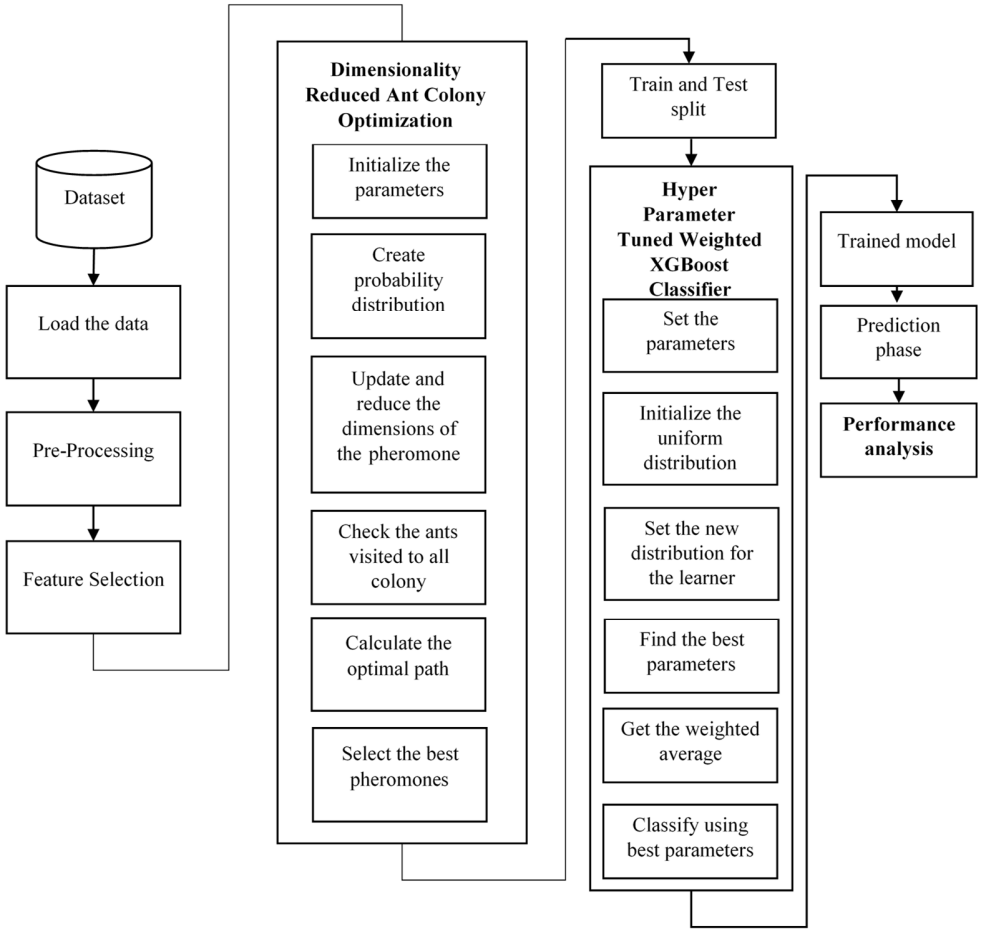

Fig. 1. Overall workflow of the proposed system.

III. RESULTS AND DISCUSSION

The proposed method, Dimensionality Reduced with ACO and Hyperparameter Tuned Weighted XGBoost Classifier (DR-ACO & HP-WXCL), was implemented in Python in Google Colab, and its performance was evaluated using accuracy, precision, recall, and F1 score.

- Accuracy: In classification tasks, accuracy is estimated by dividing the number of appropriately predicted samples by the total number of examined samples.

$$\text{Accuracy(ACC)} = \frac{\text{True Positives+True Negatives}}{\text{Total Instances}}$$

- Precision denotes the ratio of correct positive results to the total positive predictions made by the model.

$$\text{Precision (P)} = \frac{\text{True Positives}}{\text{True Positives+False Positives}}$$

- Recall, also known as sensitivity, is the ratio of true positive samples accurately recognized by the model.

$$\text{Recall(R)} = \frac{\text{True Positives}}{\text{True Positives+False Negatives}}$$

- F1-score is the harmonic mean of precision and recall, which balances both of them.

$$\text{F1} - \text{score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision+Recall}}$$

To ensure fair evaluation, all baseline machine learning models (XGBoost-DNN, LR, NB, DT, and SVM) were trained on the same datasets and feature sets after PCA-based dimensionality reduction. Table VII represents the results of the proposed method compared to prevailing techniques on the NSL-KDD dataset [15].

TABLE VII. COMPARATIVE RESULTS FOR NSL-KDD INTRUSION DETECTION

Model	Accuracy	Precision	Recall	F1-score
XGBoost-DNN	97.60%	97%	97%	97%
LR	87%	87%	87%	87%
NB	52%	28%	52%	36%
SVM	90%	90%	90%	90%
Proposed (DR-ACO & HP-WXC)	**99.20%**	**99.60%**	**98.80%**	**99.20%**

The effectiveness of the proposed model was also evaluated on the UNSW-NB15 dataset [16-20], with results compared against several established techniques, as shown in Table VIII. This dataset contains approximately 2.54 million network flow records with 49 features and 9 attack categories, each labeled as normal or malicious traffic generated using realistic modern network profiles.

TABLE VIII. COMPARATIVE RESULTS FOR UNSW-NB15 INTRUSION DETECTION

Model	Accuracy	Precision	Recall	F1-score
DT	90.4%	90.7%	88.8%	90.4%
LR	70.5%	65.9%	96.1%	78.2%
NB	52%	28%	52%	36%
SVM	85.8%	89.3%	85.9%	88.7%
Proposed (DR-ACO & HP-WXC)	**99.3%**	**92.8%**	**99%**	**95.8%**

Table IX shows that the proposed model achieved superior performance on the CIC-IDS [21] dataset, which includes about 2.8 million labeled traffic instances with 80 features extracted from packet captures, covering a wide range of modern attack types and normal user behavior in real network environments.

TABLE IX. COMPARATIVE RESULTS FOR CIC-IDS INTRUSION DETECTION

Model	Accuracy	Precision	Recall	F1-score
XGBoost-DNN	99.7%	92%	98.8%	94.9%
LR	84.9%	62%	94.8%	84.2%
NB	94.3%	83.1%	82%	81%
SVM	99.8%	99.9%	99.1%	99.2%
Proposed (DR-ACO & HP-WXC)	**99.9%**	**99.5%**	**99.8%**	**99.7%**

A confusion matrix is a significant tool for evaluating the performance of ML models on classification tasks. It provides a precise overview of the model's predictions on a test set, displaying the number of correct and wrong predictions for every class. The confusion matrix is based on four important metrics:

- True Positives (TP): The count of instances where the model accurately predicts a positive result (i.e., the model outputs Attack and the real label is also Attack).
- True Negatives (TN): The count of instances where the model accurately predicts a negative result (i.e., the model outputs Normal, and the real label is also Normal).
- False Positives (FP): The number of times the model makes an incorrect prediction for a positive situation (i.e., the model outputs Attack, but the real label is Normal).
- False Negatives (FN): The number of cases where the model incorrectly predicts a negative outcome (i.e., the model predicts Normal, but the real label is Attack).

Figure 2 presents the confusion matrix of the proposed model evaluated on the NSL-KDD dataset. For the evaluated model, the values of the confusion matrix are as follows. True Negatives (TN) are 13,414 instances correctly predicted as 'Normal,' False Positives (FP) are 43 instances incorrectly predicted as Attack when they were actually Normal (false alarms), False Negatives (FN) are 145 instances incorrectly predicted as Normal when they were actually Attack (missed attacks). True Positives (TP) are 11,593 instances correctly predicted as Attack.

Figure 3 shows the confusion matrix for UNSW-NB15. 16,840 Normal were predicted as Normal, while 100 Normal were predicted as Attack. From Attack, 13 were predicted as Normal, and 1,293 were predicted as Attack. These results demonstrate that the model performs exceptionally well, with the vast majority of normal (16,840) and attack (1,293) instances being correctly classified. Only a small number of normal cases (100) are misclassified as attacks, and very few Attack cases (13) are missed and classified as Normal.

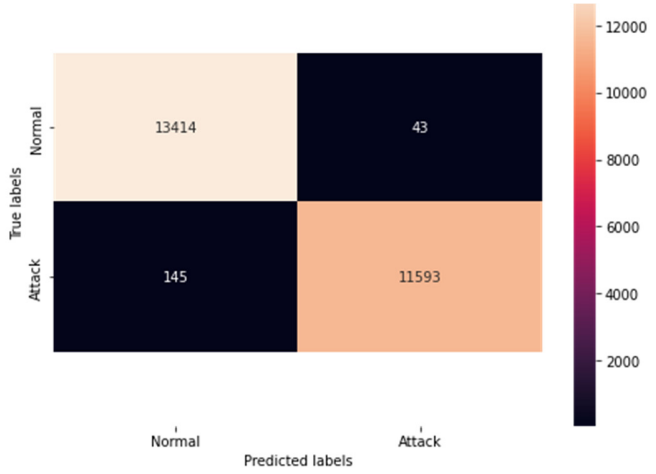

Fig. 2. Confusion matrix for NSL-KDD.

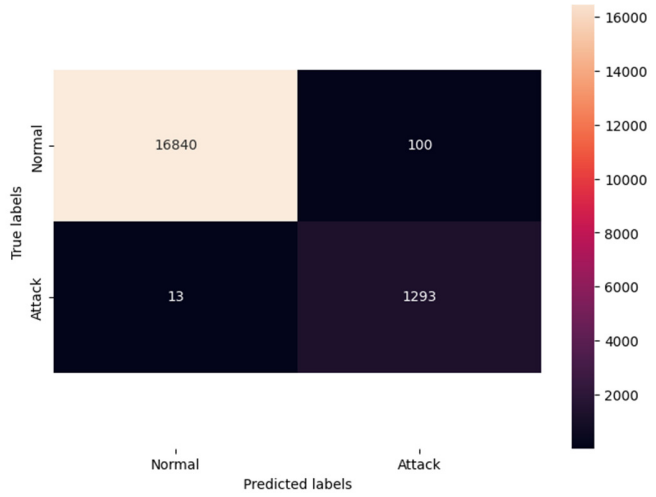

Fig. 3. Confusion matrix for UNSW-NB15.

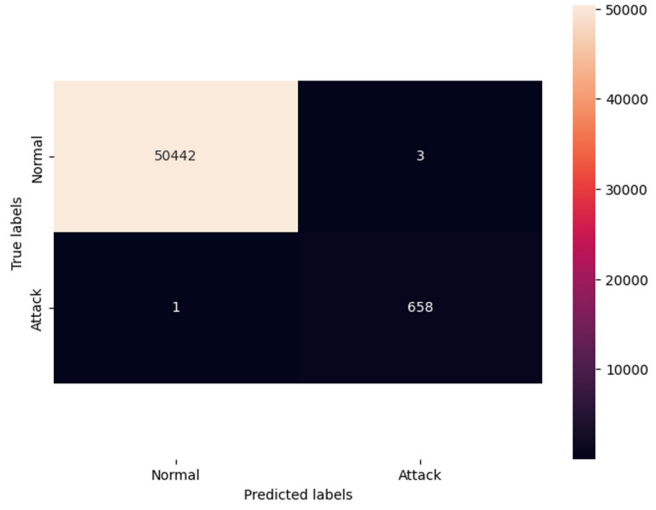

Fig. 4. Confusion matrix for CIC-IDS.

Figure 4 visualizes the confusion matrix for the CIC-IDS 2017 dataset. 50442 true negative instances were correctly identified, while 3 negative instances were incorrectly identified as Attack. Only one instance was misidentified as Normal but was actually Attack, while 658 were correctly identified as Attack.

Another important metric for evaluating classification performance is the Area Under the Curve (AUC), which represents the area beneath the Receiver Operating Characteristic (ROC) curve that plots the True Positive Rate (TPR)—the proportion of correctly identified positive cases—against the False Positive Rate (FPR)—the proportion of negative cases incorrectly classified as positive—across different decision thresholds. AUC provides an aggregate measure of a model's ability to distinguish between positive (e.g., attack) and negative (e.g., normal) instances, regardless of the classification threshold selected. Higher AUC values, closer to 1, indicate superior model performance in differentiating between the two classes.

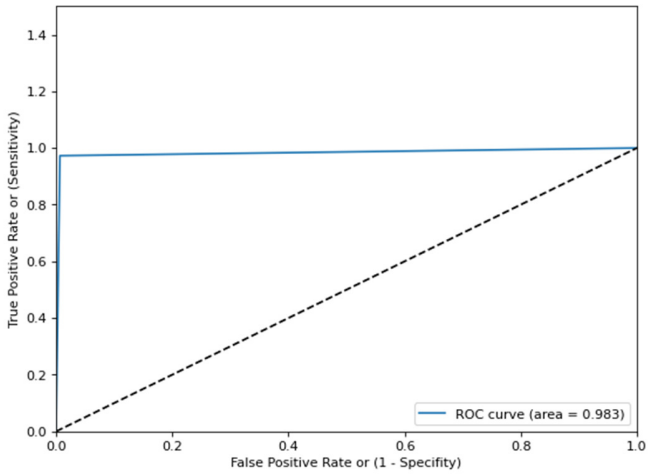

Fig. 5. ROC with NSL-KDD.

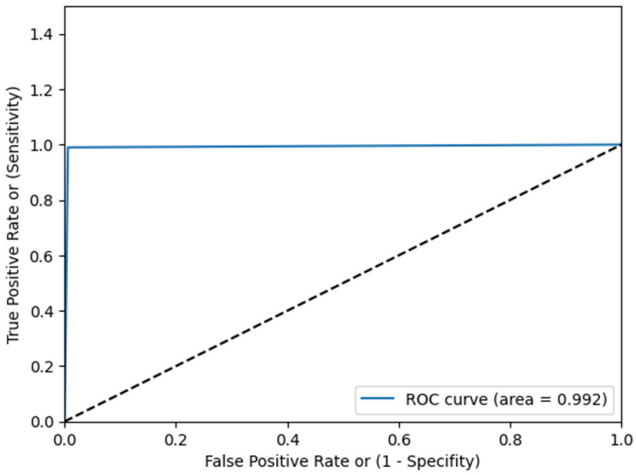

Fig. 6. ROC with UNSW-NB15.

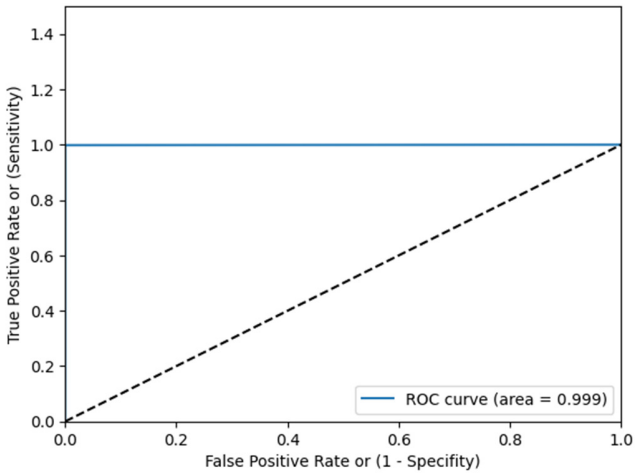

Fig. 7. ROC with CIC-IDS.

IV. CONCLUSION

A thorough examination of available IDS methodologies has highlighted perennial problems such as overfitting, lack of interpretability, and costly computation. This study placed special emphasis on feature relevance and efficient processing. The proposed model was thoroughly evaluated using three benchmarking datasets: NSL-KDD, UNSW-NB15, and CIC-IDS-2017. Preprocessing was performed to ensure data quality and ML equivalence, after which PCA and ACO effectively reduced noise and dimensionality, enhancing detection performance and interpretability. XGBoost was chosen for its robustness, regularization, and effectiveness in dealing with class imbalance.

For future research, the combination of LLMs and GANs presents a formidable and complementary method for intrusion detection. The aim is to develop smarter, more accurate, and interpretable security solutions that can identify, respond, and even forecast cyber threats in real time. This opens the doors for autonomous, self-improving IDS solutions that can reliably safeguard against the constantly changing threat landscape.

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, Oct. 2019, Art. no. 4396, 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, Feb. 2020, Art. no. 105124, https://doi.org/10.1016/j.knosys.2019.105124.

[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, Jan. 2022, Art. no. e232, https://doi.org/10.1002/itl2.232.

[4] F. Samson and S. Iseal, "Machine Learning Techniques for Enhancing Intrusion Detection Systems (IDS)," ResearchGate, 2025.

[5] L. Diana, P. Dini, and D. Paolini, "Overview on Intrusion Detection Systems for Computers Networking Security," *Computers*, vol. 14, no. 3, Mar. 2025, https://doi.org/10.3390/computers14030087.

[6] H. Dong and I. Kotenko, "Cybersecurity in the AI era: analyzing the impact of machine learning on intrusion detection," *Knowledge and Information Systems*, vol. 67, no. 5, pp. 3915–3966, May 2025, https://doi.org/10.1007/s10115-025-02366-w.

[7] S. K. R. Mallidi and R. R. Ramisetty, "Optimizing Intrusion Detection for IoT: A Systematic Review of Machine Learning and Deep Learning Approaches With Feature Selection and Data Balancing," *WIREs Data Mining and Knowledge Discovery*, vol. 15, no. 2, 2025, Art. no. e70008, https://doi.org/10.1002/widm.70008.

[8] B. Alwasel, A. Aldribi, M. Alreshoodi, I. S. Alsukayti, and M. Alsuhaibani, "Leveraging Graph-Based Representations to Enhance Machine Learning Performance in IIoT Network Security and Attack Detection," *Applied Sciences*, vol. 13, no. 13, June 2023, https://doi.org/10.3390/app13137774.

[9] A. Thakkar and R. Lohiya, "Role of swarm and evolutionary algorithms for intrusion detection system: A survey," *Swarm and Evolutionary Computation*, vol. 53, Mar. 2020, Art. no. 100631, https://doi.org/10.1016/j.swevo.2019.100631.

[10] V. Pai, Devidas, and N. D. Adesh, "Comparative analysis of Machine Learning algorithms for Intrusion Detection," *IOP Conference Series: Materials Science and Engineering*, vol. 1013, no. 1, Jan. 2021, Art. no. 012038, https://doi.org/10.1088/1757-899X/1013/1/012038.

[11] J. Lansky *et al.*, "Deep Learning-Based Intrusion Detection Systems: A Systematic Review," *IEEE Access*, vol. 9, pp. 101574–101599, 2021, https://doi.org/10.1109/ACCESS.2021.3097247.

[12] M. M. Issa, M. Aljanabi, and H. M. Muhialdeen, "Systematic literature review on intrusion detection systems: Research trends, algorithms, methods, datasets, and limitations," *Journal of Intelligent Systems*, vol. 33, no. 1, Jan. 2024, https://doi.org/10.1515/jisys-2023-0248.

[13] A. H. Ali *et al.*, "Unveiling machine learning strategies and considerations in intrusion detection systems: a comprehensive survey," *Frontiers in Computer Science*, vol. 6, June 2024, https://doi.org/10.3389/fcomp.2024.1387354.

[14] A. Momand, S. U. Jan, and N. Ramzan, "A Systematic and Comprehensive Survey of Recent Advances in Intrusion Detection Systems Using Machine Learning: Deep Learning, Datasets, and Attack Taxonomy," *Journal of Sensors*, vol. 2023, no. 1, 2023, Art. no. 6048087, https://doi.org/10.1155/2023/6048087.

[15] "NSL-KDD." Canadian Institute for Cybersecurity - UNB, [Online]. Available: https://www.unb.ca/cic/datasets/nsl.html.

[16] N. Moustafa and J. Slay, "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *2015 Military Communications and Information Systems Conference (MilCIS)*, Nov. 2015, pp. 1–6, https://doi.org/10.1109/MilCIS.2015.7348942.

[17] N. Moustafa and J. Slay, "The evaluation of Network Anomaly Detection Systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set," *Information Security Journal: A Global Perspective*, vol. 25, no. 1–3, pp. 18–31, Apr. 2016, https://doi.org/10.1080/19393555.2015.1125974.

[18] N. Moustafa, J. Slay, and G. Creech, "Novel Geometric Area Analysis Technique for Anomaly Detection Using Trapezoidal Area Estimation on Large-Scale Networks," *IEEE Transactions on Big Data*, vol. 5, no. 4, pp. 481–494, Sept. 2019, https://doi.org/10.1109/TBDATA.2017.2715166.

[19] N. Moustafa, G. Creech, and J. Slay, "Big Data Analytics for Intrusion Detection System: Statistical Decision-Making Using Finite Dirichlet Mixture Models," in *Data Analytics and Decision Support for Cybersecurity: Trends, Methodologies and Applications*, I. Palomares Carrascosa, H. K. Kalutarage, and Y. Huang, Eds. Springer International Publishing, 2017, pp. 127–156.

[20] M. Sarhan, S. Layeghy, N. Moustafa, and M. Portmann, "NetFlow Datasets for Machine Learning-Based Network Intrusion Detection Systems," in *Big Data Technologies and Applications*, vol. 371, Z. Deze, H. Huang, R. Hou, S. Rho, and N. Chilamkurti, Eds. Springer International Publishing, 2021, pp. 117–135.

[21] "CIC-IDS 2017." Canadian Institute for Cybersecurity - UNB, [Online]. Available: https://www.unb.ca/cic/datasets/ids-2017.html.