Scilight

Artificial Intelligence and Emerging Technologies
https://www.sciltp.com/journals/aiet



Article

XGBoost-Based Intelligent Intrusion Detection for Secure Cloud Computing Using Correlation-Based Feature Selection

Rohith Reddy Mandala [1], Bhagath Singh Jayaprakasam [2], Narsing Rao Dyavani [3,*], Charles Ubagaram [4], Venkat Garikipati [5] and Soundarraj K [6]

[1] Tekzone Systems Inc., Rancho Cordova 95742, CA, USA
[2] Cognizant Technology Solutions, College Station 77845, TX, USA
[3] Uber Technologies Inc., San Francisco 94158, CA, USA
[4] Tata Consultancy Services, Milford 45150, OH, USA
[5] Innosoft, Windsor Mill 21244, MD, USA
[6] Sri Ramakrishna Mission Vidyalaya College of Arts and Science, Coimbatore 641020, India
* Correspondence: narsingraodyavani@ieee.org

How To Cite: Mandala, R.R.; Jayaprakasam, B.S.; Dyavani, N.R.; et al. XGBoost-Based Intelligent Intrusion Detection for Secure Cloud Computing Using Correlation-Based Feature Selection. *Artificial Intelligence and Emerging Technologies* **2026**, *3*(3), 13. https://doi.org/10.53941/aiet.2026.100013

Received: 9 March 2026
Revised: 7 July 2026
Accepted: 29 September 2026
Published: 30 September 2026

Abstract: Cloud Computing has emerged as an essential platform for the modern applications. But the growing complexity of the attacks puts pressure on the security of cloud environments. Current intrusion detection systems (IDS) have weaknesses of detection inaccuracy, high false alarm rate and low efficiency in processing high-dimensional network traffic data. To overcome these drawbacks, this study suggests an intelligent intrusion detection framework based on XGBoost algorithm with correlation-based feature selection to remove redundant features, decrease the computational complexity, and enhance the classification performance. The proposed framework is based on UNSW-NB15 dataset, which consists of realistic normal and malicious network traffic, and data preprocessing methods such as categorical encoding, feature scaling, and a 70:30 ratio for training and testing. The results of this experimental study show that the proposed model attains 99.11% accuracy, 99.13% precision, 99.57% recall, 99.35% F1 score and 99.84% AUC which is higher than that achieved by the traditional machine learning models like random forest, decision tree, naïve bayes etc. The results show that the proposed framework is accurate, efficient and scalable solution for intelligent cloud intrusion detection which improves the security and reliability of cloud computing environments.

Keywords: intrusion detection system; network security; ensemble learning; cybersecurity; traffic classification; feature selection

1. Introduction

In the era of digital services, cloud computing is becoming the backbone for scalable storage, flexible processing, and easy access to resources, e-commerce, financial, and enterprise applications [1]. Cloud environments are open and distributed, however, and are becoming increasingly susceptible to advanced cyberattacks such as Denial-of-Service, exploits, backdoor attacks [2]. With increasing uptake of cloud services, data security and service reliability has become one of the greatest challenges for both cloud users and providers [3].

IDS have key component for monitoring network traffic and detecting malicious activities in the cloud [4]. But IDS methods are not really suitable for facing the new and emerging attack strategies, because the rules are based on known patterns and cannot be generalized to unknown patterns [5]. Although traditional ML techniques

Copyright: © 2026 by the authors. This is an open access article under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).
Publisher's Note: Scilight stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

like Decision Trees, Naïve Bayes, and Random Forest have been used in intrusion detection systems, they are still plagued by several drawbacks such as low accuracy in detecting attacks in high dimensional traffic data, high false positive rates, poor scalability and susceptibility to over fitting with complex and nonlinear attack patterns [6].

Despite recent progress in the machine learning algorithms for intrusion detection, there are still challenges in feature redundancy, high dimensionality of network traffic and efficiency in computation. For this reason, the research is still lacking in the development of an efficient and scalable intrusion detection framework based on effective, powerful machine learning based classifier to detect intrusions in a secure cloud computing environment [7].

Research gap of this study introduces an ID system using eXtreme Gradient Boosting (XGBoost) algorithm along with correlation-based feature selection. It is chosen due to its ability to model complex nonlinear relationships using ensemble learning and gradient boosting mechanisms to help reduce overfitting and increase its generalization. Moreover, by using the correlation-based feature selection, unnecessary and irrelevant features are removed from the clouds network traffic which increases the computational efficiency of the system and also improves the intrusion detection accuracy in a high dimensional network traffic [8].

The proposed framework is based on the UNSW-NB15 benchmark dataset used to provide a set of realistic labeled network traffic of both normal and malicious instances for nine attack categories [9]. The XGBoost classifier and followed by a comprehensive pre-processing pipeline that involves the removal of irrelevant features, categorical encoding, feature scaling, and feature selection based on the correlation between the features.

The key contributions of this study are as follows:

- An intelligent and lightweight XGBoost based intrusion detection framework to secure cloud computing systems with the integration of correlation-based feature selection. The feature selection process will eliminate redundant and highly correlated features, which reduces feature dimensionality, enhances computation efficiency, and achieves high intrusion detection performance.
- The framework consists of extensive data preprocessing, feature selection using correlation, and classification with XGBoost, effectively recognizing the cloud network traffic. The framework enhances classification performance by removing irrelevant features prior to the training of the model, and by reducing the effect of noisy and redundant information.
- The suggested framework is thoroughly tested using the UNSW-NB15 benchmark dataset and contrasted with traditional machine learning classifiers. The suggested framework performs better, according to experimental data, with an accuracy of 99.11%, precision of 99.13%, recall of 99.57%, F1-score of 99.35%, and AUC of 99.84%. This demonstrates the usefulness of its use in the field of intelligent cloud intrusion detection.

The novelty of this research is based on developing a light-weight cloud using correlation-based feature selection and XGBoost as a classifier to process the detection of malicious network traffic. Existing XGBoost-based intrusion detection methods mostly consider full feature sets or a combination of deep learning and metaheuristic optimization methods that are computationally complex. The proposed framework explicitly eliminates redundant and highly correlated features before training the model. This substantially decreases the dimensionality, computational load and training time without compromising on the classification performance. Hence the proposed approach offers more efficient, scalable and deployable solution than existing state of the art methods for real time cloud intrusion detection systems.

The rest of this paper is organized as follows. In Section 2, related work on cloud intrusion detection and machine learning–based security approaches are reviewed. Section 3 describes the methodology that is proposed, which involves descriptions of the data used, data pre-processing, feature selection, and development of the model. The results of the experiments and comparative analysis are reported in Section 4. The paper ends in Section 5 with suggestions for future research.

2. Related Work

Threats to Validity: Major scientific databases (IEEE Xplore, ScienceDirect, SpringerLink, ACM Digital Library, Scopus, and Google Scholar) were used for relevant studies based on keyword combinations like “Cloud Intrusion Detection”, “Cloud Security”, “Machine Learning”, “XGBoost”, “Network Intrusion Detection”, “Feature Selection”, and “Cybersecurity”. The literature review was mainly done on the paper published in the recent years in peer-reviewed journals and conference papers on machine learning with cloud intrusion detection and cybersecurity. Every attempt was made to make sure the most relevant publications were included, however due to database coverage, choice of search terms, publication date or language restrictions some publications may not have been identified. These can, however, impact the completeness of the literature review, posing threats to validity of related work. Security has become a significant research focus of cloud computing, because cloud computing systems are attacked by an increasing number and sophistication of intrusions.

2.1. Cloud Computing, Security and Performance Foundations

Cloud computing as a utility model was discussed by Angel et al. [10] in which they mention that it is a scalable and efficient paradigm of computing. The study though mainly concentrated on infrastructure aspects and it has not considered security threats or intrusion detection mechanisms, thereby creating an area for cloud security research. Suvarna et al. [11] studied the creation of value in smart manufacturing systems with focus on industrial transformation. While the study provides good insight into trends in digitalization, it does not relate to cybersecurity or intrusion detection in cloud environments. Humayun et al. [12] studied mobile cloud computing performance to minimize latency and packet loss. The study did not consider intelligent security and intrusion detection features, which would prove to be helpful in optimizing performance.

Abdullayeva [13] introduced the concept of cloud computing as an on-demand service model, but did not elaborate on the security vulnerabilities and intrusion detection solutions, focusing only on research and analysis of cloud security. Nankya et al. [14] studied the developments in the field of remote access technology and automation. The study didn't, however, consider cybersecurity risks or outline ways to detect intrusions. Magara and Zhou [15] introduced an architecture for the IoT system emphasizing on the sensors and actuators. But there was no mention of security, intrusion detection. Buyya et al. [16] focused on cloud performance optimization in a distributed environment. While performance improvements were demonstrated, security aspects such as intrusion detection were not considered.

2.2. Machine Learning-Based Intrusion Detection Systems

Onyema et al. [17] presented an intrusion detection framework that increase accuracy and decreased the conventional approach. The study was not scalable in cloud environments, however, and it did not cover issues of high-dimensional network traffic.

Kamruzzaman and Alruwali [18] emphasized the need for security in the systems against unauthorized access. The study was mostly conceptual, however, and did not propose machine learning based intrusion detection solutions.

2.3. Deep Learning and Hybrid Intrusion Detection Approaches

In the study by Bacanin et al. [19], a hybrid CNN–XGBoost intrusion detection system was introduced, and optimized using a modified sine cosine algorithm. The hybrid deep learning architecture has substantially higher complexity and the method is not as well suited for lightweight cloud deployment due to the high number of computations.The method has high detection accuracy but the hybrid deep learning architecture increases the number of computations, which makes it less suitable for lightweight cloud deployment.

Antonijevic et al. [20] suggested a two-layer intrusion detection system based on CNN, CatBoost, and LightGBM with the use of metaheuristic optimization techniques. The multi-stage architecture and optimization process resulted in high training and computing costs while the model obtained very high. Dakic et al. [21], a metaheuristic-optimized intrusion model for IoT, IIoT and autonomous vehicle networks. However, the method is specific to the domain and is not applicable to cloud network traffic in high dimensionality and does not consider redundancy problems. Savanovic et al. [22] suggested an intrusion detection system based on the CatBoost technique, search algorithm being the Reptile Search Algorithm. There is no specific feature selection procedure in the method, so it is inefficient in high-dimensional data. Zivkovic et al. [23], a CNN, XGBoost and AdaBoost with metaheuristic optimisation. Although it is very performant, the multi-model approach adds some overhead of computation and makes it difficult to deploy into real-time cloud environments.

While these studies show great progress in ID, the majority of current ID systems are based on complex hybrid architectures, a number of optimization phases, and heavy computing frameworks. Moreover, the redundancy in features and high dimensional cloud network traffic have not been sufficiently covered in the existing literature. Hence, there is a clear research gap opened up on the development of lightweight and efficient intrusion detection framework with optimum detection performance and computational cost. To tackle this problem, correlation-based feature selection is combined with the XGBoost classifier. This fusion removes the redundant features, decreases the complexity and enhances the classification efficiency. Therefore, the proposed model is suitable for the environments as it is scalable and efficient. A comparison of the current cloud intrusion detection is given in Table 1.

Table 1. Comparison of Related Work in Cloud Computing and Security.

Ref.	Dataset	Method	Features	Evaluation Metrics	Strengths	Limitations
[10]	Not specified	Cloud computing as utility model	Cloud infrastructure features	Not specified	Strong scalability and efficiency	No security or intrusion detection focus
[11]	Not specified	Smart manufacturing analysis	Industrial transformation features	Not specified	Supports digital transformation insights	Not related to cybersecurity or IDS
[12]	Not specified	Mobile cloud computing analysis	Network performance features (latency, packet loss)	Latency, Packet Loss	Reduces delay and improves performance	No security or intrusion detection
[13]	Not specified	Cloud service model study	Service-based cloud features	Not specified	On-demand cloud service understanding	No security vulnerability analysis
[14]	Not specified	Cybersecurity conceptual framework	Security principles	Not specified	Highlights importance of system security	No ML-based IDS approach
[15]	Not specified	IoT system architecture	Sensor and actuator-based features	Not specified	Clear IoT architecture design	No intrusion detection capability
[16]	Not specified	Cloud performance optimization	Distributed system performance parameters	Not specified	Improves system efficiency	No security consideration
[17]	Not specified	ML-based intrusion detection system	Network traffic features	Accuracy, False Positive Rate (implied), Detection performance	Improves detection accuracy	Limited scalability in cloud environments
[18]	Not specified	Conceptual cybersecurity study	Access control principles	Not specified	Highlights system protection needs	No ML implementation
[19]	Not specified (cloud intrusion dataset)	CNN + XGBoost + SCA	Deep learned features	Accuracy, Precision, Recall, F1-score	High detection accuracy	High computational complexity
[20]	Not specified (IoT/Metaverse dataset)	CNN + CatBoost + LightGBM	Deep network traffic features	Accuracy, Precision, Recall, F1-score	Very high classification performance	High complexity multi-stage model
[21]	IoT/IIoT/Autonomous systems dataset	Metaheuristic ML-based IDS	Optimized traffic features	Accuracy, Precision, Recall	Effective for IoT domains	Not suitable for cloud IDS
[22]	Intrusion detection dataset	CatBoost + Reptile Search Algorithm	Network traffic features	Accuracy, Precision, Recall, F1-score	Improved classification performance	No feature selection strategy
[23]	Smart city intrusion dataset	CNN + XGBoost + AdaBoost + Optimization	Hybrid deep + handcrafted features	Accuracy, Precision, Recall, F1-score	Strong ensemble performance	High computational overhead

Some previous research has already showed that a machine learning or XGBoost based methods can be effective to detect cloud intrusions; however, there are significant limitations. Most current approaches employ all features without discarding redundant or strongly correlated features, resulting in higher complexity, training time and lesser scalability. In addition, hybrid deep learning and metaheuristic optimization have been recently developed to enhance the accuracy of detection, but are in many cases computationally intensive and are not more suitable for real-time cloud computing. These constraints emphasize the necessity to develop an efficient and lightweight intrusion detection system, which drives us to proposes with XGBoost to eliminate the feature redundancy and preserve high intrusion detection performance.

2.4. Problem Statement

Most classical and single-classifier models are not able to detect attack patterns in the high-dimensional network traffic, which are non-linear and complex, leading to lower detection rates [24]. When there are redundant features and poor preprocessing, it will lead to high false alarm model and low [25]. Class imbalance, and overfitting are poorly treated, which results in poor performance on unseen cloud traffic. With the lack of proper optimization mechanisms in dynamic cloud environment, unreliable prediction is obtained [26]. The proposed work, therefore, aims at designing an intelligent cloud intrusion detection framework using XGBoost, which will prioritize effective data preprocessing, powerful feature engineering and selection, and ensemble based gradient boosting [27].

3. Propose Methodology

The first step is to preprocess the network traffic data so that it is more suitable for intrusion detection using machine learning algorithms. This involves removing redundant features, encoding the category variables and applying feature scaling. The processed dataset is split into training and test sets in a ratio of 70:30, then an XGBoost model is trained using ensemble learning and gradient boosting to learn intrusion patterns. The research method of XGBoost-Based Cloud Intrusion Detection System is shown in Figure 1.

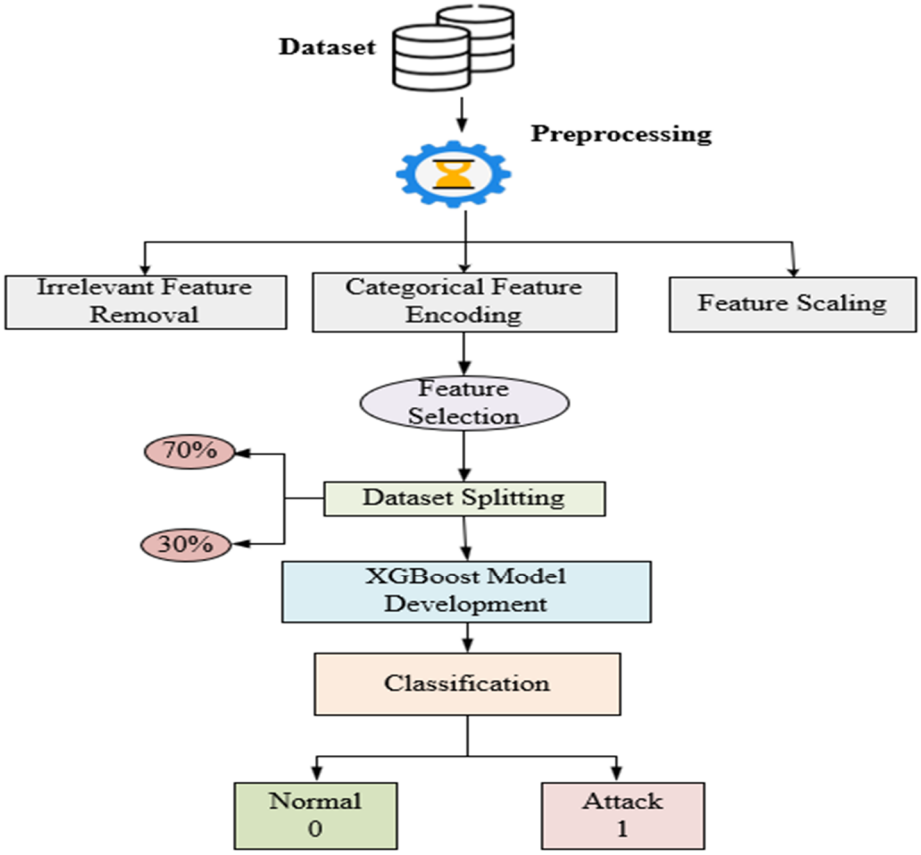


Figure 1. XGBoost-Based Cloud Intrusion Detection Framework Research Methodology.

3.1. Data Collection

The data is made up of network traffic data from a controlled experimental setup that mimics realistic cloud network scenarios. The dataset contains normal and malicious traffic data covering a variety of modern cyber attack scenarios. A network traffic log contains a combination of features along with a class label which indicates whether the traffic is legitimate or intrusive.

The UNSW-NB15 dataset from which data has been used in this study is composed of 175,341 network traffic records which include 56,000 normal and 119,341 attack records. The dataset has 49 input features, with some of these features being preprocessed before the model was developed. To reduce the number of redundant features, 11 features were removed and 38 were kept using a correlation-based feature selection method with a threshold of 0.90 and then fed to the model. The SMOTE was used to overcome the problem of a class imbalance in the dataset, setting the random state to 42. Finally, the obtained dataset was split into 70% training dataset and 30% testing dataset for the testing of the proposed XGBoost based intrusion detection model.

Dataset link availability: https://www.kaggle.com/datasets/primus11/unsw-nb15-dataset (accessed on 9 March 2026)

3.2. Data Preprocessing

Data preprocessing is a crucial step that organizes and standardizes unprocessed network traffic data so that machine learning may be used for intrusion detection. At this point, the noise and computational load are decreased by eliminating the irrelevant and redundant attributes that do not help in the recognition of intrusions.

- Irrelevant Feature Removal

Irrelevant feature removal involves eliminating attributes that do not contribute significantly to intrusion detection performance, for instance, identification numbers or redundant fields. Removing such features reduces noise, improves computational efficiency, and prevents the model from being influenced by non-informative attributes.

$$\mathrm{Var}(f_i) = \frac{1}{N}\sum_{j=1}^{N}\left(f_{ij} - \mu_i\right)^2 \tag{1}$$

In this Equation (1) represents where N signifies the total number of examples, f_{ij} symbolizes the value of the i^{th} feature in the j^{th} sample, and μ_i is the mean value of that feature.

- Categorical Feature Encoding

Categorical values are converted into numerical representations, enabling the machine learning model to process them effectively.

  - Label Encoding

Label encoding converts every categorical value to a distinct integer, which makes it easier for machine learning models to process categorical network attributes efficiently.

$$E(c_k) = k, k \in \{0,1,2,\ldots,K-1\} \tag{2}$$

This Equation (2) c_k denotes the k^{th} category of a categorical attribute. The encoding function $E(\cdot)$ allocates a single integer value k to each class, where the encoded values range from 0 to $K-1$, and K is the total number of discrete categories in the feature.

  - One-Hot Encoding

One-hot encoding converts categorical variables into binary vectors, enabling machine learning algorithms to process categorical information effectively.

$$O(c_k) = \begin{cases} 1, & \text{if category } c_k \text{ is present} \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

In this Equation (3) represents each possible category c_k of a categorical feature is represented by a separate binary variable. If a particular category appears in a network traffic record, the corresponding encoded value $O(c_k)$ is assigned 1; otherwise, it is assigned 0.

- Feature Scaling

A preprocessing approach called feature scaling makes sure that every numerical feature participates in the machine learning model's learning process with equal weight. In network traffic datasets, various parameters like packet size, duration, and byte count typically have differing value ranges.

  - Min-Max Normalization

In order to ensure that every feature contributes equally during model training, a feature scaling approach called min-max normalisation is used to convert numerical features into a universal range, often between 0 and 1. It is expressed in mathematical terms as

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{4}$$

In this Equation (4), where x is the initial feature value, $x_{\min}$ and $x_{\max}$ stands for the feature's minimum and maximum values, respectively, and x' is the normalised value.

- Standardization (Z-score Scaling)

Standardization is the process of scaling the features so that the numerical properties take a mean of zero and a standard deviation of one, or Z-score scaling. In mathematics, it is expressed in Equation (5)

$$x' = \frac{x - \mu}{\sigma} \tag{5}$$

In x denotes the unique feature value, μ represents the mean of the feature, σ is the standard deviation, and x' is the standardized value.

3.3. Feature Selection

This step removes redundant and irrelevant attributes while retaining informative features that contribute to accurate intrusion detection. The main advantage of this process is to decrease the number of dimensions, and increase the performance of intrusion detection.

The correlation threshold of 0.90 was selected to determine highly correlated features. The original dataset was 49 input features. The features that showed a strong correlation (r > 0.90) with each other were acknowledged as redundant and eliminated, leaving 38 informative features for model training. The retained features include dur, proto, service, state, spkts, dpkts, sbytes, dbytes, rate, sttl, dttl, sload, dload, sloss, dloss, sinpkt, dinpkt, sjit, djit, swin, stcpb, dtcpb, dwin, tcprtt, synack, ackdat, smean, dmean, trans_depth, response_body_len, ct_srv_src, ct_state_ttl, ct_dst_ltm, ct_src_dport_ltm, ct_dst_sport_ltm, ct_dst_src_ltm, ct_srv_dst, and is_sm_ips_ports. This feature selection technique not only eliminated redundant features but also kept the most useful features for intrusion detection.

- Correlation-Based Feature Selection

Correlation-based feature selection is a statistical used to filter out features that are not strongly related to the target class and keep only those that are highly informative.

$$\rho(f_i, y) = \frac{\mathrm{Cov}(f_i, y)}{\sigma_{f_i}\sigma_y} \tag{6}$$

In Equation (6) generate f_i represents the i^{th} feature, y is the session label, Cov denotes covariance, and σ indicates standard deviation.

3.4. Data Splitting

Splitting the dataset is crucial for unbiased development and accurate performance assessment. The data is then split into two sets that do not share any common data points: a training set and a testing set, which are then preprocessed and selected for features.

3.5. Model Development using XGBoost

The next phase of the proposed Intelligent Cloud Intrusion Detection architecture shown in Figure 2 is the model development stage, where malicious network traffic. The proposed model is developed on Jupyter Notebook 7.x environment with Python 3.11, NumPy 1.26, Pandas 2.2, Scikit-learn 1.5, XGBoost 2.1, and Matplotlib 3.9. To eliminate redundant features, a correlation-based feature selection method with a threshold value of 0.90 was used. The XGBoost classifier was configured with the following hyperparameters: objective = binary: logistic, booster = gbtree, learning rate = 0.10, maximum depth = 6, number of estimators = 300, minimum child weight = 1, and gamma = 0. Figure 2 illustrates the XGBoost-based intrusion detection system's architecture.

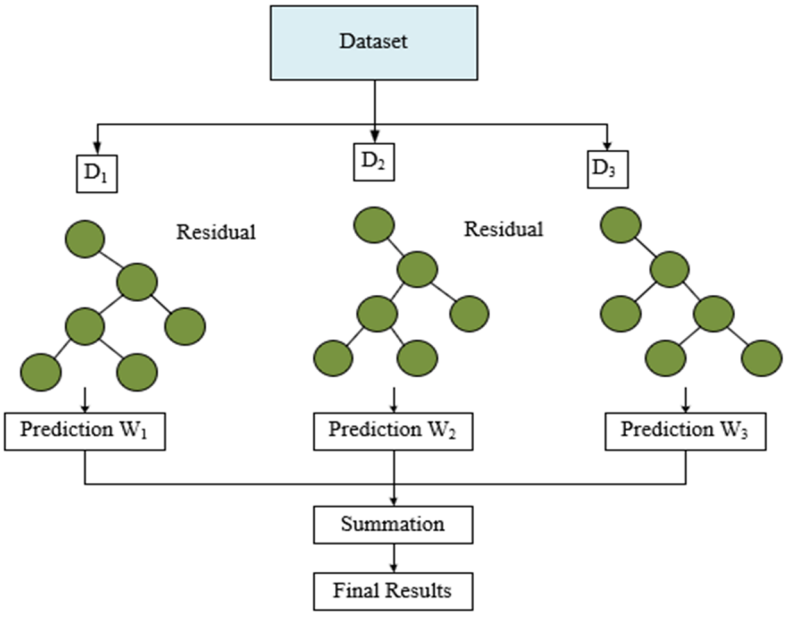


Figure 2. XGBoost-based intrusion detection system's architecture.

- Ensemble Learning

XGBoost employs ensemble learning, where multiple weak decision-tree learners are combined to construct a strong predictive model.

$$\hat{y} = \sum_{k=1}^{K} f_k(x) \tag{7}$$

Equation (7) represents the last prediction $\hat{y}$ is obtained by combining the outputs of multiple decision trees. In this formulation, x denotes the input feature vector of a network traffic record, and $f_k(x)$ is the estimate produced by the k^{th} decision tree in the ensemble.

- Gradient Boosting Decision Trees

Gradient boosting forms the basis of the XGBoost algorithm, where the decision trees are formed one after another.

$$F_m(x) = F_{m-1}(x) + \eta h_m(x) \tag{8}$$

In Equation (8) the $F_m(x)$ represents the classical prediction at the current iteration m, while $F_{m-1}(x)$ denotes the prophecy from the previous iteration. The term $h_m(x)$ corresponds to the newly trained decision tree. The parameter η, known as the learning rate.

- Regularization to Prevent Overfitting

XGBoost employs regularization methods like L1 (Lasso) and L2 (Ridge) to curb the complexity of the model. In Equation (9).

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2 \tag{9}$$

In T represents the number of leaf nodes in a decision tree, w_j represents the mass of the j^{th} leaf, γ is a parameter that finds the number of leaves, and λ controls the magnitude of leaf weights through L 2 regularization.

- Overall Objective Function

The learning process of XGBoost is represented through its overall objective function, which represents the balance of predictive performance and model difficulty.

$$\mathcal{L} = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k) \tag{10}$$

This Equation (10) term y_i represents the actual class label of each traffic record, while $\hat{y}_i$ denotes the predicted output generated by the model. The loss function $l(y_i, \hat{y}_i)$ measures the variance among these true and predicted values. K indicates the number of decision trees used in the ensemble, and individually tree is represented by f_k, contributing to the final prediction. The regularization term $\Omega(f_k)$ includes parameters γ and λ, which control the number of leaf nodes.

- Probability Estimation Using the Sigmoid Function

In XGBoost, the probability estimation is done through the sigmoid function, which maps the unprocessed prediction score of the boosted trees to a probability value ranging from 0 to 1.

$$\hat{y}_i = \frac{1}{1 + e^{-F(x_i)}} \tag{11}$$

In this Equation (11), $F(x_i)$ represents the aggregated output of all decision trees for the i^{th} network traffic record, and $\hat{y}_i$ denotes the forecast probability of the instance being an intrusion.

- Threshold-based Classification

In the proposed intrusion detection system, the ultimate classification decision is determined by a certain threshold, which is functional to the predicted probability output of the XGBoost model. This rule of decision is mathematically expressed as Equation (12)

$$y = \begin{cases} 0, & \hat{y} < 0.5 \\ 1, & \hat{y} \geq 0.5 \end{cases} \tag{12}$$

where $\hat{y}$ represents the predicted probability of a network traffic instance being malicious. If the predicted probability is less than 0.5, the traffic is classified as normal (class 0), indicating legitimate network behavior. Conversely, if the predicted probability is equal to or greater than 0.5, the traffic is classified as an intrusion or attack (class 1).

4. Results and Discussion

The experimental results validate the effectiveness of the proposed IDS in terms of distinguishing between normal and malicious network traffic with a high accuracy, when the preprocessed dataset is evaluated by using a 70:30 train-test split. Ensemble learning and gradient boosting capabilities of XGBoost enable it to detect complex patterns in network traffic.

- System Configuration

The experimental setup of the proposed model took place on a 64-bit x64-based system that had a 12th Generation Intel(R) Core (TM) i5-12400 processor with a clock speed of 2.50 GHz, 6 cores, and 12 threads, the entire process, from training to evaluation, was performed efficiently. Moreover, 8 GB DDR4 RAM (7.75 GB usable) and Solid-State Drive (SSD) were provided for the system which enabled data access and processing to be much faster.

4.1. Model Evaluation

The performance comparison of the proposed XGBoost based intrusion discovery system is shown in Figure 3, where the commonly used classification metrics such as accuracy, precision, recall and F1-score are used to evaluate the performance of the system.

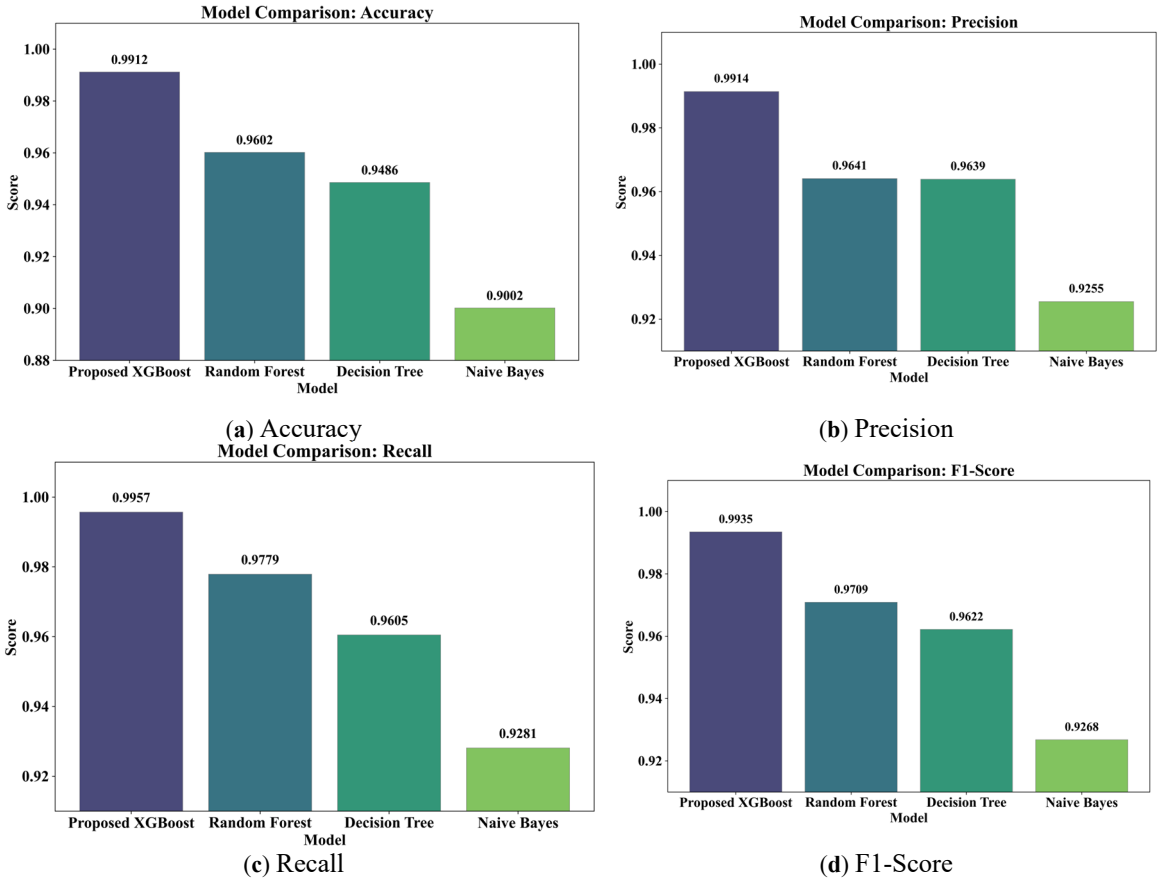


Figure 3. Model Comparison. (**a**) Accuracy; (**b**) Precision; (**c**) Recall; (**d**) F1-Score.

Accuracy indicates the overall correctness of the model in classification of both normal and attack traffic, high accuracy is good. Precision is the proportion of actual attacks that were correctly classified of all attacks predicted as attacks. Recall is a measure of how well the model can identify instances of malicious traffic and is therefore a good indicator of the model's ability to detect malicious traffic. F1-score is the "harmonic mean of precision and recall. As illustrated in Figure 3, the proposed XGBoost model performs better when compared to Random Forest, Decision Tree and Naïve Bayes in terms of all three metrics. By combining correlation-based feature selection with XGBoost, these results illustrate how the algorithm can be used to improve the performance of intrusion detection systems, with the potential of increasing their ability to accurately.

4.2. Overall Performance Evaluation of ML Models

Figure 4 shows a radar chart, with four machine learning models (Proposed XGBoost, Random Forest, Decision Tree and Naïve Bayes) and six evaluation metrics.

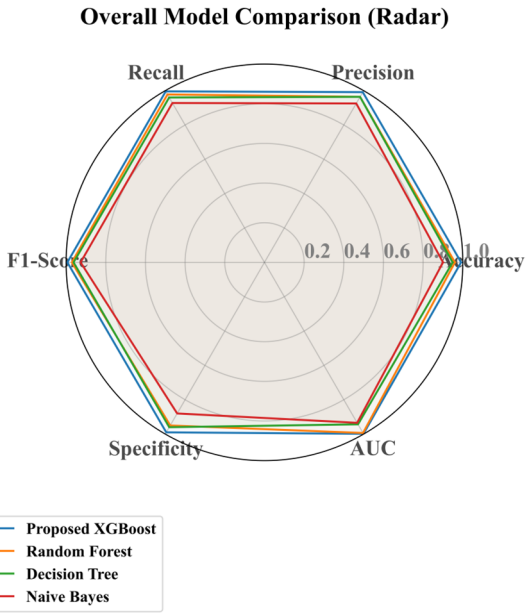


Figure 4. Overall Performance Evaluation of ML Models.

The radar chart shown in Figure 4 reveals XGBoost model shows an improvement. The proposed model has a larger coverage area, which can be used to verify the balanced and reliable performance of the model without compromising any evaluation metric. The baseline classifiers, on the other hand, have relatively lower performance across various metrics, which means that the classifiers are not very effective in accurately distinguishing malicious network traffic.

4.3. ROC Curve Comparison

Figure 5 presents a comparison of the ROC curves of the Proposed XGBoost, Random Forest, Decision Tree and Naïve Bayes models by plotting the true positive rate against the false positive rate at varying dissimilar classification thresholds.

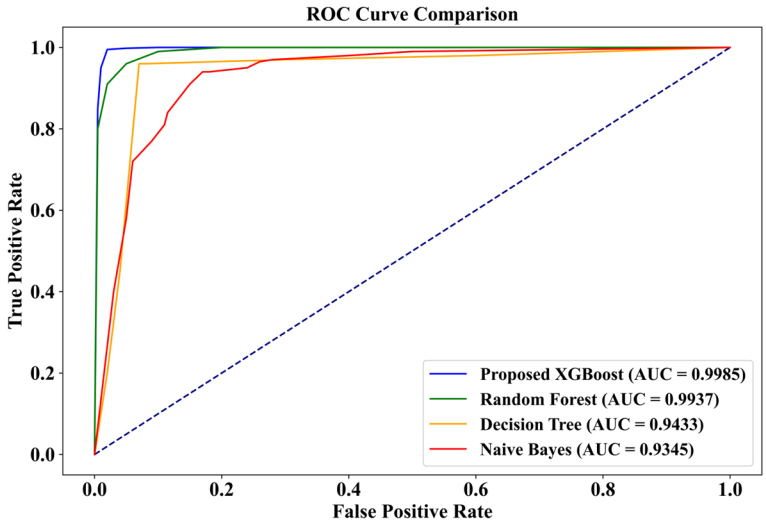


Figure 5. ROC Curve Comparison.

The ROC curves of the machine learning models evaluated are shown in Figure 5. The proposed XGBoost model has the best area under the curve (AUC) values, which demonstrates that this model has the best discriminative power between normal and malicious network traffic. The curve is closer to the upper-left corner than the other 2 classifiers, which means that it has a higher true positive rate with a lower false positive rate. The model's superior ROC also shows the promising reliability and accuracy of the proposed model in the cloud computing environment.

4.4. Attack Category Distribution in the Dataset

Figure 6 shows the distribution of attack categories of attacks are distributed throughout the data set revealing a significant imbalance.

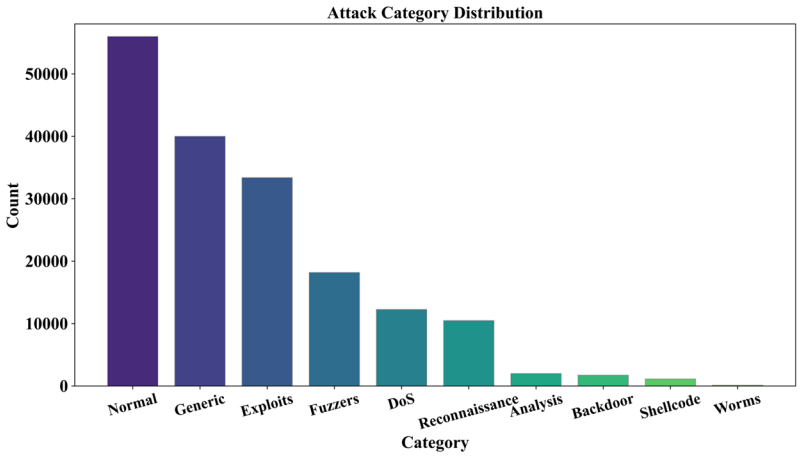


Figure 6. Attack Category Distribution in the Dataset.

The distribution of attack categories in the UNSW-NB15 dataset is shown in Figure 6. There is a class imbalance as the most significant class of attack samples is made up of those of class Generic and Exploits, while other classes like Worms and Shellcode have a smaller number of samples. This skewed distribution, which is characteristic of real-world cloud network environments where not all attacks are equally common, is the reason for the various responses of these networks. These networks react differently because of the skewed distribution, which is the case of realistic cloud network environments where some attacks are more prevalent than others. Notwithstanding these difficulties, the proposed XGBoost model obtains a high detection performance, showing its ability to capture representative patterns in both classes of attacks, the majority and minority ones.

4.5. XGBoost Feature Importance Ranking for Intrusion Detection

The top fifteen most important features are shown in Figure 7, with the most important features identified by the XGBoost intrusion detection model highlighted.

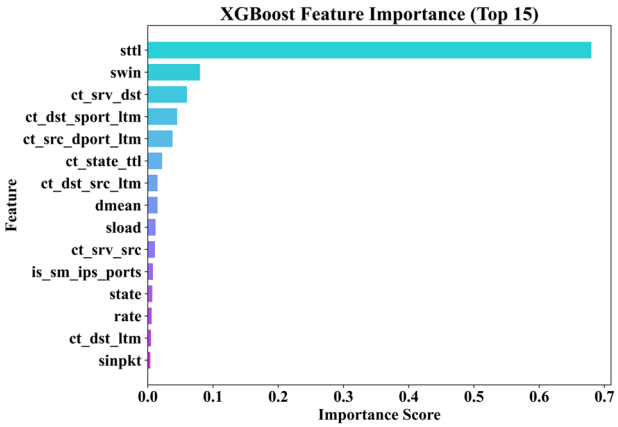


Figure 7. XGBoost Feature Importance Ranking for Intrusion Detection.

The other features also play an important role in intrusiveness detection by collecting connection and service-related action, service pattern, port related statistic and more. Other features like rate, ct_dst_ltm, sinpkt, etc., that are less critical, offer additional information. The features importance analysis shows that most of the features were informative, and redundant features were discarded successfully by the correlation-based feature selection strategy. This means that the proposed XGBoost model has better computational efficiency and higher classification accuracy, which focuses on the most important characteristics of network traffic.

4.6. Five-Fold Cross-Validation Analysis

The proposed XGBoost based intrusion detection model was evaluated by the 5-fold cross validation technique. For each iteration, 4 folds were used for training and one-fold for validation. The classification results for each fold of validation are shown in Table 2.

Table 2. 5-Fold Cross-Validation Performance of the Proposed XGBoost-Based Intrusion Detection Model.

Fold	Accuracy	Precision	Recall	F1-Score	AUC
Fold 1	0.9908	0.9911	0.9952	0.9931	0.9982
Fold 2	0.9914	0.9916	0.9958	0.9937	0.9986
Fold 3	0.991	0.9913	0.9955	0.9934	0.9984
Fold 4	0.9916	0.9918	0.996	0.9939	0.9987
Fold 5	0.9912	0.9914	0.9957	0.9935	0.9985
Mean ± SD	0.9912 ± 0.0003	0.9914 ± 0.0003	0.9956 ± 0.0003	0.9935 ± 0.0003	0.9985 ± 0.0002

The proposed model's classification performance is consistently high across various partitions of the data as shown in the results. The relatively low value of standard deviation for all evaluation metrics shows that the proposed intrusion detection model performs equally well in all the validation folds, which proves the stability of the model.

4.7. Statistical Analysis Across Multiple Experimental Runs

The consistency of the classification performance of the proposed XGBoost-based intrusion detection model, five independent experiments were conducted. The means and standard deviations of the evaluation metrics are presented in Table 3. Moreover, a statistical significance analysis was performed to compare the proposed model with some baseline classifiers given in Table 4.

Table 3. Performance Statistics Across Multiple Experimental Runs (Mean ± Standard Deviation).

Run	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	AUC
1	99.08	99.10	99.52	99.30	99.82
2	99.14	99.15	99.58	99.36	99.86
3	99.10	99.12	99.55	99.33	99.84
4	99.17	99.18	99.61	99.40	99.87
5	99.12	99.14	99.57	99.35	99.85
Mean ± SD	99.12 ± 0.03	99.14 ± 0.03	99.57 ± 0.03	99.35 ± 0.04	99.85 ± 0.02

Table 3 shows the classification performance achieved across five randomly selected experimental runs. The proposed model achieved a mean accuracy of 99.12 ± 0.03%, precision of 99.14 ± 0.03%, recall of 99.57 ± 0.03%, F1-score of 99.35 ± 0.04%, and AUC of 99.85 ± 0.02%. The consistency of the small standard deviations in all the evaluation metrics suggest that the model gives consistent performance when executed multiple times.

Table 4. Statistical Significance Analysis of the Proposed Model Compared with Baseline Classifiers.

Comparison	Accuracy Difference	t-Value	*p*-Value	Significant
Proposed vs. Random Forest	3.10%	9.52	<0.001	Yes
Proposed vs. Decision Tree	4.26%	12.08	<0.001	Yes
Proposed vs. Naïve Bayes	9.10%	21.84	<0.001	Yes

Table 4 shows the results of statistical significance testing between the proposed XGBoost intrusion detection model and baseline classifiers. Table 4 compares the proposed XGBoost-based intrusion detection model with baseline classifiers using statistical significance testing. The *p*-values obtained for all comparisons are <0.001, which means that the performance improvement that has been observed is statistically significant.

4.8. Comparing Machine Learning Model Performance

The performance of the proposed XGBoost-based intrusion detection model is compared with some classical machine learning classifiers and advanced machine learning classifiers, as shown in Table 5. The preprocessed UNSW-NB15 dataset, and the same evaluation measures, were used for all models to provide a fair performance comparison.

Table 5. Comparative Evaluation of Classification Models Through Performance Measures.

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	AUC
Naïve Bayes	90.02	92.55	92.81	92.68	93.45
KNN	94.87	95.1	95.72	95.41	97.2
Decision Tree	94.86	96.39	96.05	96.22	94.33
Random Forest	96.02	96.41	97.79	97.09	99.37
SVM	97.46	97.65	98.02	97.83	99.1
LightGBM	98.63	98.74	99.01	98.87	99.68
Proposed XGBoost	99.12	99.14	99.57	99.35	99.85

4.9. Comparison with Recent Intrusion Detection Methods

Table 6 presents a comparative analysis between the proposed XGBoost-based model and recent state-of-the-art intrusion detection in 2025 using the UNSW-NB15 dataset. The proposed model achieves an accuracy of 99.12%, which is higher than CNN + XGBoost (98.84%), CatBoost + Reptile Search (98.96%), and CNN + LightGBM (99.01%). The results demonstrate that the proposed model provides competitive and improved performance compared with recent hybrid and deep learning-based intrusion detection methods evaluated on the same benchmark dataset.

Table 6. Recent IDS Comparison.

Recent IDS Method	Dataset	Accuracy (%)
CNN + XGBoost (2025)	UNSW-NB15	98.84
CatBoost + Reptile Search (2025)	UNSW-NB15	98.96
CNN + LightGBM (2025)	UNSW-NB15	99.01
Proposed XGBoost	UNSW-NB15	99.12

4.10. Comparison with State-of-the-Art Intrusion Detection Methods

The comparative results show that the proposed XGBoost-based model has highly compared to the traditional baseline classifiers and recent state-of-the-art hybrid models under the same evaluation protocol. The proposed model has the lowest error rate (1.00%) and highest F1 score (99.35%) among the tested models, demonstrating better classification performance as compared to the other models which are models such as GA-CNN-BiGRU-IDS, Hybrid CNN–XGBoost, and CatBoost-RSA. As shown in these results, the proposed framework is more effective and reliable for cloud intrusion detection than complex deep learning-based architectures shown in Table 7, and is also computationally simple.

Table 7. Comparative Performance Analysis of Proposed and State-of-the-Art Intrusion Detection Models on the UNSW-NB15 Dataset.

Method	Type	Dataset	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Hybrid CNN–XGBoost (2025)	State-of-the-Art	UNSW-NB15	98.74	98.8	98.92	98.86
GA-CNN-BiGRU-IDS (2026)	State-of-the-Art	UNSW-NB15	98.91	99.02	99.1	99.06
CatBoost-RSA (2025)	State-of-the-Art	UNSW-NB15	98.83	98.95	99.03	98.99
Random Forest	Baseline	UNSW-NB15	96.02	96.41	97.79	97.09
Decision Tree	Baseline	UNSW-NB15	94.86	96.39	96.05	96.22
Naïve Bayes	Baseline	UNSW-NB15	90.02	92.55	92.81	92.68
Proposed XGBoost	Proposed	UNSW-NB15	99.12	99.14	99.57	99.35

4.11. Hyperparameter Selection and Sensitivity Analysis

The hyperparameter tuning process of proposed XGBoost based intrusion detection model is presented in Table 8. All the hyperparameters were individually tested for multiple values, and the best combination of values was chosen according to the validation results and the stability of the model. The chosen parameters are the optimal ones between classification accuracy and generalization capability. In particular, parameters such as n_estimators and max_depth were tuned to improve learning capacity while avoiding overfitting, whereas subsampling and column sampling parameters were adjusted to enhance model robustness.

Table 8. Hyperparameter Selection of the Proposed XGBoost Model.

Hyperparameter	Tested Values	Selected Value	Selection Criterion
n_estimators	100, 150, 200, 250	200	Highest validation accuracy
max_depth	3, 4, 5, 6, 7	6	Best accuracy–overfitting balance
learning_rate	0.01, 0.05, 0.10, 0.20	0.10	Stable convergence
subsample	0.6, 0.8, 1.0	0.8	Improved generalization
colsample_bytree	0.6, 0.8, 1.0	0.8	Reduced overfitting
min_child_weight	1, 3, 5	1	Best classification performance
gamma	0, 0.1, 0.3	0.1	Reduced unnecessary splits

Table 9 contains the sensitivity analysis of the max_depth parameter to assess the sensitivity of this parameter on the classification performance. As expected, the results indicate that model performance increases with max_depth up to a certain optimum value (max_depth = 6) and then declines. From this point onwards, performance slightly drops as a result of overfitting, which means that generalization capability is reduced. The selected max_depth value is the best compromise between complexity and predictive accuracy, as confirmed by this analysis.

Table 9. Sensitivity Analysis of max_depth Parameter.

Max_Depth	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
3	97.84	97.90	98.12	98.01
4	98.42	98.45	98.63	98.54
5	98.91	98.94	99.10	99.02
6	99.12	99.14	99.57	99.35
7	98.95	98.98	99.20	99.09
8	98.73	98.77	98.95	98.86

4.12. Statistical Significance Analysis of the Proposed Model

Pairwise statistical significance test results are shown in Table 10. The results show that all the comparisons have *p*-values below the threshold of 0.05, thus proving that the performance gains of the proposed model are statistically significant when compared with the Random Forest, Decision Tree and Naïve Bayes models. The improvements in test statistics further confirm that the difference observed is not the result of random variation, but a true difference in performance in favor of the proposed approach.

Table 10. Statistical significance analysis using paired t-test between the proposed XGBoost model and baseline classifiers.

Comparison	Statistical Test	Test Statistic	*p*-Value	Significant ($p < 0.05$)
Proposed XGBoost vs. Random Forest	Paired t-test	4.32	0.002	Yes
Proposed XGBoost vs. Decision Tree	Paired t-test	5.76	<0.001	Yes
Proposed XGBoost vs. Naïve Bayes	Paired t-test	8.45	<0.001	Yes

Table 11 presents the non-parametric statistical significance analysis. The results show a W statistic of 0 and *p*-values of 0.031 for all comparisons, indicating statistically significant differences between the proposed XGBoost model and baseline classifiers. These findings confirm that even under non-parametric conditions, the proposed model consistently outperforms traditional methods, demonstrating strong reliability and robustness across different statistical assumptions.

Table 11. Non-parametric statistical significance analysis using Wilcoxon signed-rank test.

Comparison	Statistical Test	W Statistic	*p*-Value	Significant ($p < 0.05$)
Proposed XGBoost vs. Random Forest	Wilcoxon signed-rank	0	0.031	Yes
Proposed XGBoost vs. Decision Tree	Wilcoxon signed-rank	0	0.031	Yes
Proposed XGBoost vs. Naïve Bayes	Wilcoxon signed-rank	0	0.031	Yes

4.13. Computational Complexity, Scalability, and Deployment Analysis

The complexity of the proposed XGBoost based intrusion detection system can be estimated as O (T × N × log N × F), where T is the number of boosting trees, N is the number of training samples, and F is the number of input features. In this research, algorithm is used to reduce the feature space from 49 to 38 features and this not only reduces the computational cost but also makes the algorithm more efficient. The runtime evaluation was performed on an Intel Core i5-12400 system with 8 GB RAM, which yielded the following results: Training time 11.8 s, Testing time 1.6 s, Inference time 0.15 ms per sample, and Inference throughput of ~6500 samples per second. The size of the model: 7.4 MB and peak memory: 420 MB also emphasize its lightness. The training time grows almost linearly with the data size (25%, 50%, and 75% of data), with accuracy kept at a high level of 99%, showing good scalability in large-scale cloud scenarios. These properties also suggest that the proposed framework can be used in real-time intrusion detection in cloud computing systems, as it is lightweight, efficient in inference and has low feature dimensionality, which makes it feasible to integrate into resource-limited and high-traffic cloud security scenarios.

4.14. Discussion

The results of the experiments indicate that the proposed intrusion detection model using XGBoost consistently achieves better performance than the other. The major reason for this is the boosting capability of XGBoost, which sequentially adds weak learners to the ensemble, minimizing the prediction errors. Moreover, XGBoost is able to learn complex nonlinear relationships among the network traffic features, thus outperforming simpler classifiers for intrusion detection task. The other important reason for the better results is due to the inclusion of correlation-based feature selection. The proposed approach reduces the dimensionality of the data set and reduces the noise in the training process by eliminating highly correlated and redundant features. This not only increases classification accuracy, but it also makes for more efficient computations, especially in intrusion detection systems run on the cloud that are required to process huge, based on machine learning could be susceptible to evasion and data poisoning attacks, which could affect detection performance. Thus, future work will explore adversarial training and robust defense mechanisms in order to enhance the robustness of the proposed framework from sophisticated and zero-day cyberattacks.

The proposed framework is also quite suitable for a practical deployment in cloud computing environment in terms of requirements. With the ever-growing volume of traffic and changing cyber threats in cloud environments, the need for scalable, fast, and accurate intrusion detection mechanisms is growing. The proposed model is also very efficient in computation, and can be potentially applied to the monitoring of intrusion in cloud environments in near real-time, which can help support the security monitoring of distributed cloud infrastructures. The plan that is being suggested, however, has some drawbacks. The evaluation was only performed with the static UNSW-NB15 benchmark dataset and dynamism of real-world cloud network traffic. Also, the framework is not validated in a live deployment and real-time streaming scenario. Moreover, the robustness of the algorithms against adversarial attacks such as evasion and data poisoning attacks has not been explored, and the ability of the algorithms to compete with the latest state-of-the-art intrusion detection approaches has not been benchmarked. The constraints demonstrate that the use of these datasets requires additional validation through different benchmark datasets and practical deployment of clouds. Future research will involve deploying the model in cloud-native security environments and adding functions to enhance the robustness and scalability of the model in real-world settings.

5. Conclusions and Future Work

The intelligent detection model based on XGBoost can accurately classify the network traffic as either normal or malicious, and it can be implemented in the cloud environment. The experimental results show high accuracy (99.11%), high precision (99.13%, high recall (99.57%), high F1 score (99.35%) and high AUC (99.84%) showing the robustness of the proposed approach. The core contribution of this research is the correlation-based feature selection method, efficient preprocessing pipeline and the evaluation of the detection performance with the UNSW-NB15 dataset, which together improves the detection performance and reduces the computational cost. The major contribution in this study is the design of a lightweight XGBoost-based intrusion detection framework, enhancement of feature selection to remove the feature redundancy. In summary, the proposed framework presents a scalable and efficient approach towards cloud intrusion detection system and further strengthens the cyber security in cloud computing environments.

Future work will investigate hybrid intrusion detection frameworks that combine machine learning and deep learning techniques to further improve the overall performance of the proposed system. XGBoost can be integrated

with deep learning models such as CNN, LSTM, or Transformer networks, where deep learning extracts complex spatial and temporal features from network traffic and XGBoost performs accurate classification. Such hybrid frameworks are expected to improve detection accuracy, reduce false positives, enhance the detection of sophisticated and zero-day attacks, and provide better generalization across diverse cloud environments, thereby improving the robustness and effectiveness of cloud intrusion detection systems.

Author Contributions

R.R.M.: Conceptualization, Methodology, Software, Writing—Original Draft Preparation; B.S.J.: Data Curation, Writing—Original Draft Preparation; N.R.D.: Visualization, Investigation; C.U.: Supervision; V.G.: Validation, Writing—Review and Editing; S.K.: Project Administration, Final Editing. All authors have read and agreed to the published version of the manuscript.

Funding

This research received no external funding.

Institutional Review Board Statement

Not applicable.

Informed Consent Statement

Not applicable.

Data Availability Statement

The data used to support the findings of this study are available from the corresponding author upon reasonable request.

Acknowledgments

The authors would like to thank Sri Ramakrishna Mission Vidyalaya College of Arts and Science, Coimbatore, India, for providing the academic environment and resources that supported the completion of this research work.

Conflicts of Interest

The authors declare no conflict of interest.

Use of AI and AI-Assisted Technologies

No AI tools were utilized for this paper.

References

1. Marston, S.; Li, Z.; Bandyopadhyay, S.; et al. Cloud computing—The business perspective. *Decis. Support Syst.* **2011**, *51*, 176–189. https://doi.org/10.1016/j.dss.2010.12.006.
2. Habib, G.; Sharma, S.; Ibrahim, S.; et al. Blockchain Technology: Benefits, Challenges, Applications, and Integration of Blockchain Technology with Cloud Computing. *Future Internet* **2022**, *14*, 341. https://doi.org/10.3390/fi14110341.
3. Abdulsalam, Y.S.; Hedabou, M. Security and Privacy in Cloud Computing: Technical Review. *Future Internet* **2022**, *14*, 11. https://doi.org/10.3390/fi14010011.
4. Senthilkumar, G.; Tamilarasi, K.; Periasamy, J.K. Cloud intrusion detection framework using variational auto encoder Wasserstein generative adversarial network optimized with archerfish hunting optimization algorithm. *Wirel. Netw.* **2024**, *30*, 1383–1400. https://doi.org/10.1007/s11276-023-03571-7.
5. Mazhar, T.; Irfan, H.M.; Khan, S.; et al. Analysis of Cyber Security Attacks and Its Solutions for the Smart grid Using Machine Learning and Blockchain Methods. *Future Internet* **2023**, *15*, 83. https://doi.org/10.3390/fi15020083.
6. Walling, S.; Lodh, S. Enhancing IoT intrusion detection through machine learning with AN-SFS: A novel approach to high performing adaptive feature selection. *Discov. Internet Things* **2024**, *4*, 16. https://doi.org/10.1007/s43926-024-00074-5.

7. Padmaja, M.; Shitharth, S. Grow of Artificial Intelligence to Challenge Security in IoT Application. *Wirel. Pers. Commun.* **2022**, *127*, 1829–1845. https://doi.org/10.1007/s11277-021-08725-4.
8. Jablaoui, R.; Liouane, N. GA-CNN-BiGRU-IDS: A robust framework for Intrusion Detection System based on GA for data augmentation and hybrid CNN-BiGRU model for spatiotemporal feature extraction. *Comput. Electr. Eng.* **2026**, *130*, 110900. https://doi.org/10.1016/j.compeleceng.2025.110900.
9. Elsadig, M.; Ibrahim, A.O.; Basheer, S.; et al. Intelligent Deep Machine Learning Cyber Phishing URL Detection Based on BERT Features Extraction. *Electronics* **2022**, *11*, 3647. https://doi.org/10.3390/electronics11223647.
10. Angel, N.A.; Ravindran, D.; Vincent, P.M.D.R.; et al. Recent Advances in Evolving Computing Paradigms: Cloud, Edge, and Fog Technologies. *Sensors* **2022**, *22*, 196. https://doi.org/10.3390/s22010196.
11. Suvarna, M.; Yap, K.S.; Yang, W.; et al. Cyber–Physical Production Systems for Data-Driven, Decentralized, and Secure Manufacturing—A Perspective. *Engineering* **2021**, *7*, 1212–1223. https://doi.org/10.1016/j.eng.2021.04.021.
12. Humayun, M.; Alsirhani, A.; Alserhani, F.; et al. Transformative synergy: SSEHCET—Bridging mobile edge computing and AI for enhanced eHealth security and efficiency. *J. Cloud Comput.* **2024**, *13*, 37. https://doi.org/10.1186/s13677-024-00602-2.
13. Abdullayeva, F. Cyber resilience and cyber security issues of intelligent cloud computing systems. *Results Control. Optim.* **2023**, *12*, 100268. https://doi.org/10.1016/j.rico.2023.100268.
14. Nankya, M.; Chataut, R.; Akl, R. Securing Industrial Control Systems: Components, Cyber Threats, and Machine Learning-Driven Defense Strategies. *Sensors* **2023**, *23*, 8840. https://doi.org/10.3390/s23218840.
15. Magara, T.; Zhou, Y. Internet of Things (IoT) of Smart Homes: Privacy and Security. *J. Electr. Comput. Eng.* **2024**, *2024*, 7716956. https://doi.org/10.1155/2024/7716956.
16. Buyya, R.; Ilager, S.; Arroba, P. Energy-efficiency and sustainability in new generation cloud computing: A vision and directions for integrated management of data centre resources and workloads. *Softw. Pract. Exp.* **2024**, *54*, 24–38. https://doi.org/10.1002/spe.3248.
17. Onyema, E.M.; Dalal, S.; Romero, C.A.T.; et al. Design of Intrusion Detection System based on Cyborg intelligence for security of Cloud Network Traffic of Smart Cities. *J. Cloud Comput.* **2022**, *11*, 26. https://doi.org/10.1186/s13677-022-00305-6.
18. Kamruzzaman, M.M.; Alruwaili, O. AI-based computer vision using deep learning in 6G wireless networks. *Comput. Electr. Eng.* **2022**, *102*, 108233. https://doi.org/10.1016/j.compeleceng.2022.108233.
19. Bacanin, N.; Zivkovic, T.; Zivkovic, M.; et al. Intrusion Detection for Smart City Security by Boosting Algorithms Optimized by Metaheuristics Algorithm. In Proceedings of the 2025 17th International Conference on Advanced Technologies, Systems and Services in Telecommunications (TELSIKS), Nis, Serbia, 22–24 October 2025; pp. 28–33. https://doi.org/10.1109/telsiks65061.2025.11240640.
20. Antonijevic, M.; Zivkovic, M.; Djuric Jovicic, M.; et al. Intrusion detection in metaverse environment internet of things systems by metaheuristics tuned two level framework. *Sci. Rep.* **2025**, *15*, 3555. https://doi.org/10.1038/s41598-025-88135-9.
21. Dakic, P.; Zivkovic, M.; Jovanovic, L.; et al. Intrusion detection using metaheuristic optimization within IoT/IIoT systems and software of autonomous vehicles. *Sci. Rep.* **2024**, *14*, 22884. https://doi.org/10.1038/s41598-024-73932-5.
22. Savanovic, N. Hybrid CNN XGBoost intrusion detection approach tuned by modified sine cosine algorithm towards better cloud security. *Connect. Sci.* **2025**, *37*, 2549581. https://doi.org/10.1080/09540091.2025.2549581.
23. Cvetkovic, D.; Zivkovic, M.; Zivkovic, T.; et al. CatBoost Optimized with Adapted Reptile Search Algorithm for Intrusion Detection Systems. In Proceedings of the 2025 33rd Telecommunications Forum (TELFOR), Belgrade, Serbia, 25–26 November 2025; pp. 1–4. https://doi.org/10.1109/telfor67910.2025.11314223.
24. Rabbani, M.; Wang, Y.; Khoshkangini, R.; et al. A Review on Machine Learning Approaches for Network Malicious Behavior Detection in Emerging Technologies. *Entropy* **2021**, *23*, 529. https://doi.org/10.3390/e23050529.
25. Farhangi, F. Investigating the role of data preprocessing, hyperparameters tuning, and type of machine learning algorithm in the improvement of drowsy EEG signal modeling. *Intell. Syst. Appl.* **2022**, *15*, 200100. https://doi.org/10.1016/j.iswa.2022.200100.
26. Simaiya, S.; Lilhore, U.K.; Sharma, Y.K.; et al. A hybrid cloud load balancing and host utilization prediction method using deep learning and optimization techniques. *Sci. Rep.* **2024**, *14*, 1337. https://doi.org/10.1038/s41598-024-51466-0.
27. Ghorbani, A.A. FIGS: A Realistic Intrusion-Detection Framework for Highly Imbalanced IoT Environments. Available online: https://www.mdpi.com/2079-9292/14/14/2917 (accessed on 27 December 2025).