Vol.3, No.5, Page 1203- 1221, 2025 https://doi.org/10.59628/jast.v3i5.1777 ISSN 2958-9568 JAST

SUN PUBLISHER

مجلة جامعة صنعاء للعلوم التطبيقية والتكنولوجيا

Sana'a University Journal of Applied Sciences and Technology

https://journals.su.edu.ye/index.php/jast/

Enhancing Intrusion Detection System in cloud computing Using Machine Learning Techniques

Khawla Ali Maodah[1] *, Sharaf Alhomdy[2] and Fursan Thabit[3]

[1,2]**Department of Information Technology, Faculty of Computer & Information Technology, Sana'a , Sana'a, Yemen.,**
[3]**Department of Computer Engineering, Faculty of Engineering, Ege University, Turkey.**

*Corresponding author: khawlaa800@gmail.com

Abstract

Strong cybersecurity measures are becoming vitally crucial as cloud computing utilization rises. Advanced and dynamic cyberthreats are frequently difficult for traditional intrusion detection systems (IDS), which rely on preset signatures and rules, to identify. This study improves the detection of known and unknown intrusions in cloud systems using Machine Learning (ML) methods. The UNSW-NB15 dataset was used to train and assess a number of ML classifiers, including Random Forest (RF), Decision Tree (DT), XGBoost, Naïve Bayes (NB), Support Vector Machine (SVM), Logistic Regression (LR), and Gradient Boosting (GB). It uses full feature training across several classifiers and also investigates the implications of feature reduction, in contrast to many other studies that mainly employ full feature sets to train RF alone. Critical metrics including accuracy (ACC), precision, recall, and F1-score are used to analyze classifier performance and provide a thorough evaluation of their efficacy in intrusion detection. The findings show that using all characteristics the RF and DT obtained perfect accuracy (1.00). In the case of less characteristics when using feature selection techniques (RF-based selection, information gain, or mutual information), the RF retained the best accuracy (0.94), whereas NB performed the worst overall. This study emphasizes the significance of feature selection in enhancing IDS performance and shows that ML-based techniques may greatly increase threat detection in cloud settings, even when feature dimensionality is lowered.

Article Info

Keywords:
Cloud Computing Security, Intrusion Detection System, Machine Learning , Feature Selection, UNSW-NB15 dataset, Cybersecurity, Cybersecurity Threat Detection

Article History:
Received: 7-May-2025,
Revised: 1-July-2025,
Accepted: 19-October-2025,
Available online: 28 October 2025.

1. INTRODUCTION

Modern Internet-based technology known as CC provides a variety of IT resources and services, including operating systems, storage, hardware, network infrastructure, and software programs, at affordable prices. It provides a number of advantages, including as decreased management costs, faster development, enhanced cost-effectiveness, and scalability. But because CC infrastructure is so complicated, it is vulnerable to a number of security flaws. The handling and storage of customer data in distant data centers raises concerns about potential dangers. CC choices may be influenced by a number of reasons, such virtualized security threats, difficulties locating physical data storage, potential weaknesses in online storage systems, system compatibility problems, and legal or regulatory constraints [1] [2]. Security is one of the primary issues with cloud systems [3].In reaction to these difficulties [4]. IDSs often prioritize event logging above proactive intrusion prevention, even though they are frequently employed to monitor network activity. Conventional intrusion detection systems, which frequently employ preset rules or signatures, have a tendency to overfit to known threats and may produce a significant number of false alarms when confronted with new attack patterns. Moreover, they cannot react to changing threat circumstances since they are static. By allowing IDSs to identify possible threats through pattern recognition and predictive analysis, ML methods, on the other hand, provide a proactive model. By adjusting to changing behaviors over time rather than depending just on pre-established signatures, ML-driven systems lessen the need for human updates and increase their resistance to threats that have not yet been identified [5].

In some sectors, ML has recently shown to be a useful technology, providing solutions for problems with poor detection rates and a high number of false alarms. ML techniques have been widely used to proactively detect and remove threats and address security flaws in cloud systems [1] [6]. Cybersecurity uses ML, a subfield of artificial intelligence, for prediction systems and zero-day attack detection [7] . The term ML refers to a collection of algorithms that can analyze data, spot trends, and forecast future events based on those trends [6] . Three main learning categories are included in ML: supervised learning, unsupervised learning, and semi-supervised learning [7] . While supervised learning uses labeled training data, unsupervised learning uses unlabeled training data. On the other hand, when training data contains both labeled and unlabeled examples, semi-supervised learning takes place. Numerous IDS techniques are often employed, such as Ensemble Methods, K-Nearest Neighbor (KNN), K-Mean Clustering, Random Forest (RF), XGBoost, Decision Tree (DT), and Support Vector Machine (SVM) [6] [8]. The cybersecurity experts suggested incorporating ML into the design of the IDS to improve its reliability in preventing network intrusions. By using ML, the IDS may improve categorization and handle malware detection issues. As a result, ML-based IDSs provide several benefits, including improved accuracy, more precision in identifying possible threats, and the capacity to identify novel assaults [9]. This study uses sophisticated ML algorithms, such as Random Forest, Decision Tree, XGBoost, and others, to improve cloud security. It uses feature selection methods and the UNSW-NB15 dataset to address the security risks in cloud systems. CC offers scalable and cost-effective services, but it is susceptible to various threats that might compromise data security.

1.1. Problem statement:

Due to the accessibility and usefulness of online services, CC has emerged as a vital tool for both consumers and businesses. The increasing use of cloud services has made them easy targets for hackers that try to steal private and corporate information. Notwithstanding the advancements in IDS designed to monitor and prevent unauthorized access, conventional systems that depend on preset criteria or signatures are becoming outdated and fail to recognize new and advanced assault methods. While ML techniques provide a promising solution by analyzing past attack patterns and detecting unknown threats, the optimal performance of ML-based IDS in cloud environments has not yet been fully realized. There is a critical need for a comprehensive evaluation of different ML algorithms and feature selection techniques to enhance detection accuracy, reduce false alarms, and ensure operational efficiency in cloud computing environments.

1.2. Objective:

The aim of this study is to enhance IDS in cloud computing by evaluating the impact of feature selection techniques on the system's accuracy, in order to improve the detection of both known and unknown cyber threats using machine learning approaches.

1.3. Contributions:

The Contributions of this study are:

- Comprehensive Evaluation of Classifiers: Conduct a comprehensive study for seven ML classifiers, including RF, DT, XGBoost, NB, SVM, LR, and GB, using the UNSW-NB15 dataset various feature selection techniques. The dataset mimics simulation network traffic and attack scenarios, it offers a trustworthy baseline for assessing IDSs.
- Impact on Cloud Security: enhanced capability to identify both known and unknown attack patterns through an innovative dual-detection mechanism. This analysis provides practical insights for optimizing IDS performance in resource-constrained cloud environments

1.4. The paper's structure:

The rest of paper is organized as follows: Background information is given in section 2. Related works are assessed under section 3. The UNSW-NB15 dataset, feature selection, classifiers used, performance evaluation techniques, and experimental setup are all covered in the description of the suggested model in section 4. and the findings and analysis are covered in section 5. section 6 presents the conclusion and future work.

2. BACKGROUND

2.1. Intrusion Detection System:

IDS is a part of a system that keeps an eye on data flow and system operations without actively averting problems. It searches for abnormalities that could indicate technical problems or unusual use, rather than precisely identifying criminal activity or breaches. Instead of immediately alerting users, it creates records when it finds specific patterns. By providing insights rather than immediately addressing threats, an IDS indirectly enhances network security in contrast to proactive solutions like firewalls, antivirus programs, or access control systems [10] [11]. Monitoring tools are frequently employed in place of active defensive techniques like honeypots, in conjunction with IDS. IDSs use numerous approaches to identify suspicious traffic [12] .Host-based IDS (HIDS) and network IDS (NIDS) are the two primary forms of IDS development that are often recognized. On a single system with host-level changes, HIDS mainly ex-

amines internal operations and file integrity. However, host-specific events may be missed by NIDS for network traffic monitoring. Many times, NIDS detection uses established methods or statistical deviations to identify abnormalities. [13].. IDS can be divided into two broad categories: pattern-matching and behavior-based. Even while behavior-based IDS may not always employ ML or DL algorithms, they rely on departures from predicted activities. Instead of employing dynamic analysis, pattern-matching IDSs look through a library of recognized behaviors to discover matches. Some IDS systems only alert users when specific criteria or conditions are satisfied, rather than running constantly. Based on their responses, IDS systems may be divided into two groups: To prevent such assaults, an active IDS not only identifies threats but also rejects questionable communications. Passive IDS: This kind of IDS only monitors and analyzes traffic, alerting the administrator to threats and possible vulnerabilities but does nothing to fix them. [13] [14]

2.2. Machine Learning

ML is a subfield of artificial intelligence (AI) that leverages mathematical models and algorithms to enable computers to learn and make choices by evaluating large amounts of data and coming to important conclusions on their own [15]. ML may employ several techniques to produce distinct models, and there may be a variety of methods to engage with these models. Depending on the dataset, a network operator may use semi-supervised learning when there is insufficient labeled data or supervised learning when there is a lot of labeled data to train a predictor [16]. ML is commonly divided into three primary categories: supervised learning, unsupervised learning, and reinforcement learning. Supervised ML Algorithm: This method trains on labeled data, where each piece of data has a predefined label. The algorithm uses this tagged data to provide predictions or classifications. For supervised learning to be effective, the data must be accurately labeled. Classification is a crucial tactic in this area. Labels in the training data are not used by the unsupervised ML algorithm. The system automatically identifies structures and patterns in the data, frequently utilizing cosine similarity to evaluate relationships between data points. Clustering is a popular strategy in unsupervised learning.
Reinforcement Learning: This technique teaches an agent how to accomplish certain goals through interactions with the environment. Through trial and error, the agent constantly enhances its performance based on input or incentives from the environment. Reinforcement learning works especially well in dynamic and complicated contexts. For IDS, ML may address a number of issues, including boosting processing speed, cutting down on calculation time, and enhancing security threat detection accuracy [17] [18] [19] [20] .

2.2.1. ML-Based IDS:

Some benefits of integrating ML with IDS include the following:

- Supervised ML-based IDS: These systems can detect a range of assaults by monitoring traffic flow patterns. IDS based on unsupervised ML can identify unknown or novel assaults.
- Resource Efficiency: IDS based on ML usually require little to moderate processing power. Advanced Detection: These systems can detect complex attack behaviors more precisely and respond faster by recognizing them.
- Adaptability: When new threat types appear, ML-based IDS that use clustering and outlier detection algorithms don't need to update attack datasets frequently [21].

2.3. Cloud computing :

CC is a notion of service that provides customers with on-demand access to resources via the Internet, typically with a pay-per-use pricing structure. It enables users to utilize shared computer resources, including as servers, storage, and software, without having to manage or maintain the infrastructure [22]. CC consists of five key players, each with a unique role. The Cloud Consumer (or Cloud Service Consumer, CSC) is the company that acquires and uses the services of a cloud provider, paying on a use-based basis. The cloud provider, sometimes referred to as the cloud service provider or CSP, is in charge of offering cloud services to customers. The cloud auditor conducts an unbiased evaluation of the functionality, security, and performance of cloud services and information systems. The Cloud Broker enables commercial transactions by serving as an intermediary between the customer and the cloud provider. The Cloud Carrier also makes sure that the provider provides the user with a connection and cloud services [23].

2.3.1. CC Security

Cybersecurity is the protection of systems and data from inside and outside the organization, from human and non-human threats, in both the digital and physical domains. According to the CIA triad, it is frequently characterized as:

- Confidentiality refers to preserving user privacy and limiting access to information by unauthorized individuals.
- Integrity: Verifying that data is accurate and, since its creation, hasn't been altered or tampered with.
- Making sure that data is always available to authorized users when they need it is known as availability.

In order to protect user data, CC must incorporate cybersecurity. By giving cybersecurity first priority in cloud

settings, businesses can reduce the risk of cyberattacks while still adhering to relevant security regulations and guidelines [24], [25].

3. RELATED WORK:

To address the problems IDS face, a profusion of research and practical solutions employing AI and ML have lately been proposed.

The study [1] proposed employing an ensemble-based deep learning technique to identify assaults in cloud systems with an SDN-based cloud architecture. In this case, the ensemble model is constructed using K-means and deep learning classifiers. This method reduces the computing costs of deep learning.

classifiers while improving their performance. Two datasets, the SDN-based DDOS attack dataset and the CICIDS 2018 dataset, are used to train and assess this model. In terms of F1 measure, precision, accuracy, and recall, the suggested strategy outperforms previous intrusion detection methods. The recommended technique produced accuracy and precision scores of 99.685 and 0.992, respectively.

Devi and A. Jain et al. [26] Examine the security problems that CC confronts as a result of its dispersed nature, focusing on the complexity of protecting cloud resources. The study focuses on the application of Deep Learning (DL) techniques to improve intrusion detection in cloud systems. As network infrastructures expand, the necessity for effective IDS becomes increasingly crucial. The study also emphasizes the need of expanding the datasets used to train IDS, since more data improves detection accuracy. The research proposes using strong deep learning techniques to improve cloud security by evaluating publicly accessible IDS data. The study found that feature learning algorithms such as soft-max regression (SMR) and STL achieved over 98% accuracy in multi-class identification, solving anomaly detection concerns caused by inadequate normal data patterns in training.

Sanagana et al. [27] Examine security challenges in CC owing to the massive amounts of data handled and stored. IDS are vital for monitoring network traffic and identifying malicious activity. The article offers a new SSAFS-DLID strategy that uses the Salp Swarm Algorithm (SSA) for efficient feature selection, Long Short-Term Memory (LSTM) for anomaly detection, and the Adam optimizer for model optimization. This strategy lowers computing complexity while retaining good accuracy. Empirical results reveal that the model has a 99.71% accuracy rate, suggesting its ability to improve cloud security while reducing false positives. The approach offers a viable solution for protecting cloud infrastructures against cyber threats.

Kumar Samriya et al. [28] provided an enhanced Network IDS (NIDS) to address security issues in CC, which faces increasing cyber dangers due to its dispersed nature. To improve efficiency, the system uses Support Vector Machine (SVM) and XGBoost methods, which have been augmented with a Crow Search Algorithm. XGBoost-based feature selection improves classification accuracy. The system is evaluated on the NSL-KDD and UNR-IDD datasets, and it outperforms earlier systems, proving its suitability for contemporary NIDS applications.

Mghames et al. [29] Developed an ML-based IDS to identify Distributed Denial of Service (DDoS) threats. They trained and tested on the CIC-IDS-2018 dataset with five distinct ML techniques: DT, RF, LR, SVM, and multi-layer neural networks. To boost performance, dimensionality was reduced using principal component analysis (PCA). The multi-layer neural network outperformed all other models, with a classification accuracy of 99.9992% for detecting DDoS attacks.

Eluri et al. [30] addressed the challenge of spotting disturbances in organizational networks by categorizing network activity as normal or abnormal and attempting to eliminate misclassification. They applied two powerful data mining algorithms, SVM, DT, and K-Means, to enhance data organization. This approach was developed and tested on the KDDCUP99 dataset. The results showed that the proposed approach surpassed previous tactics in terms of precision and processing time, implying that it is particularly effective in detecting new threats.

Vibhute et al. [31] studied cloud data security and created a network ID based on the well-known NSL-KDD dataset. They created an ensemble learning-based RF approach to identify the most significant features. The system discovered and diagnosed network intrusions using three ML models: SVM, LR, and K-nearest neighbors (KNN), with validation accuracies of 87.58%, 88.86%, and 98.24%, respectively. The suggested approach has showed potential in detecting cyberattacks in real-time.

Attou et al.[32] recommended a feature-engineered cloud-based IDS based on the Random Forest (RF) classifier. The addition of the RF model significantly improves the accuracy of the recommended detection technique (ACC). The Bot-IoT and NSL-KDD datasets were used to validate the model, which achieved 98.3% and 99.99% accuracy, respectively.

Al-Sharif et al. [4] established an IDS framework for handling security challenges in cloud settings, where standard IDS solutions frequently fail owing to increased complexity and numerous attack vectors. Instead of using a single powerful classifier, they suggested a collective learning approach that combines numerous weaker models to create a more reliable detection system. Their strategy used bagging with Random Forest as the principal model and compared its efficacy to three boosting variants: Ensemble AdaBoost, Ensemble LPBoost, and Ensemble RUSBoost. Evaluations were conducted utilizing several divisions of the CICID2017 dataset. Among the investigated models, Ensemble RUSBoost had the

Figure 1. *Proposed model*

greatest average accuracy at 99.821%, while the bagging approach performed particularly well on the DS2 subgroup, with an accuracy of 99.997%. To further test their technique, the researchers compared their model to an existing solution, emphasizing its comparative benefits and enhanced detection capacity.

4. PROPOSED MODEL

This section explains a ML -based IDS that uses the UNSW-NB15 dataset to identify unusual network activity in cloud settings. Among the methods are:

- Class imbalance may be addressed and model performance enhanced by using data preparation techniques like normalization, encoding, and RandomOverSampler.
- Feature selection methods use all of the UNSW-NB15 dataset's features, and algorithms including Information Gain, Mutual Information, and RF-based significance are used to find pertinent qualities.
- Model Training and Evaluation: Classifiers like RF, DT, SVM, XGBoost, LR, GB, and NB are trained and assessed using metrics like accuracy, precision, recall, and F1-score in order to efficiently categorize network traffic. By taking these steps in a methodical way, the ML-IDS seeks to enhance security threat detection and mitigation in CC settings.

4.1. UNSW-NB15 DATASET:

Using the well-known UNSW-NB15 dataset, which was produced by the University of New South Wales (UNSW) in Australia, we investigated and evaluated intrusion detection techniques in this study. This dataset is a real-time benchmark that simulates actual network traffic and contains a range of attack scenarios that contemporary networks may face. It was created to overcome the NSL-KDD dataset's drawbacks, which include low attack variability and a lack of diversity in traffic patterns. The testing environment is more difficult and representative as the UNSW-NB15 dataset has a wider range of attack characteristics and traffic patterns [33]. Normal, fuzzers, analysis, backdoors, dos, exploits, generic, reconnaissance, shellcode, and worms are the ten categories into which the UNSW-NB15 dataset is divided. Without labels, it contains forty-two attributes. The testing set has 82,332 instances, whereas the training set contains 175,341 occurrences. Furthermore, the training and testing sets have a skewed distribution of classes. [34], [35], [36].

4.2. DATA PREPROCESSING:

The data preparation script will be used to load, clean, normalize, and balance the UNSW-NB15 data set. The following is a summary of the steps:
Data Loading: The load_data() method is used to read datasets from CSV files. It handles mismatched column names by dynamically changing the list of columns and ensuring that the dataset loads properly.
Data Categorization Cleaning: The clean_column_values() method is used to clean and standardize the values of category columns ("proto," "service," "state," and "attack_cat"). It ensures that the data is consistent and properly arranged by eliminating extraneous characters like newlines and extra spaces.
Label Encoding: The label_encode_columns() method uses LabelEncoder to translate category properties into numerical labels. This is important for ML models that need numerical input. Feature Normalization: scikit-learn's StandardScaler is used by the normalize_data() method to normalize the feature set (X). Normalization enhances the performance of a number of methods, such as logistic regression and support vector machines,

by guaranteeing that each feature has the same scale.

Class Distribution and Sampling: Before training the model, the script verifies the target variable's (y) class distribution. To balance the uneven dataset, the RandomOverSampler function of the imblearn module is utilized to oversample the minority class.
Merging Features: To get the final features (X_combined), all normalized features are combined. The completed dataset is in the unsw_45features.csv file. This pretreatment technique makes sure the data is clean, properly structured, standardized, and balanced before it is utilized for ML tasks .

4.3. Features selection:

This method is used to find and describe the relationships between significant data pieces. It makes it easier to simplify the model and reduces the amount of time required to test and train for a range of results [37]. This study makes use of all attributes, as well as those chosen through the application of Random Forest features and filtering methods. Mutual information and information gain. Mutual Information (MI)-based feature selection, a classifier-independent filter strategy for dimensionality reduction, attempts to address these issues by selecting a substantial subset of features [38]. Mutual information-based feature selection is a popular filter method for improving IDS effectiveness. By analyzing the connection between each characteristic and the class label, it ascertains which features have the strongest reciprocal reliance [39]. Information Gain reduces the influence of irrelevant features by classifying them according to their importance. It locates the feature on a given class with the most information [40]. Information Gain (IG) measures the amount of information by using the concepts of entropy and conditional entropy. Random Forest is a popular ML technique for classification and regression. An ensemble approach aggregates forecasts from many decision trees to improve accuracy. Random Forest operates using the bagging technique, which mixes many models to improve overall performance [41].

4.4. Employed classifiers:

Random Forest:
RF is one kind of supervised ML model. It is based on the principles and concepts of classification algorithms for decision trees [42]. It is a cooperative classifier that enhances accuracy by combining two distinct phases: feature selection and classification [43]. RF constructs a large number of decision trees during the training phase in order to provide predictions for either regression or classification problems. It then provides the mean or mode prediction for each tree independently. [44] RF is a fantastic choice for IDS because of its ability to handle noisy data. Compared to other techniques, RF's high accuracy and low false detection rate make it a useful tool for processing noisy data in network packets. Effective data processing may allow RF to regulate continuous data properties in network packets and produce better results [45]. The ability of RF to handle scenarios requiring both regression and classification is one of its numerous advantages. It effectively arranges large, multi-dimensional datasets. Furthermore, Random Forest resolves the overfitting issue and increases model accuracy. The useful information it provides on the relative feature significance makes it easy to choose the most relevant characteristics for the classifier [46].
Naïve bayes :
NB, a variation of Bayes' Theorem, makes the assumption that the qualities are very unrelated to each other. Based on Bayes' theory of probability, this classification approach makes the assumption that the presence of one characteristic has no effect on the probability of another [47]. The foundation of the Naïve Bayes technique is the idea that characteristics are independent and have conditional probability. The classifier assigns the sample to the class with the highest probability after calculating the conditional probabilities for each class for each input [13].
Decision Tree
DT is a commonly used approach for classification issues. It uses a tree structure to arrange data, with classifications based on choices made at each stage. The final categorization is displayed by the leaf nodes, the branches display the test results, and each non-terminal node indicates a test or decision point [47].
Support Vector Machine
For binary classification, SVM is thought to be the best ML technique. IDS uses a binary classification system that classifies transactions as either regular or intrusions, regardless of the attack method [48] . Finding a hyperplane in the n-dimensional feature space that maximizes the margin of separation is the main task of SVMs. SVMs have the advantage of producing results that are acceptable even with short training datasets since they only use a small number of support vectors to build this hyperplane. It is important to remember, nevertheless, that noise close to the hyperplane may have an impact on SVM performance [9] By using kernel functions to transform the original feature space into high-dimensional feature spaces with linearly separable instances, SVM may also address non-linear classification problems [44] [49].
Logistic Regression:
LR is a classification method that may be used for both binary and multiclass classification jobs since it predicts categorical results. It evaluates the likelihood of an event happening using the logistic function, with results ranging from 0 to 1. Two classes are distinguished by a 0.5 threshold: a result is classed as class 1 if it is greater

Table[1]: Confusion Matrix for Intrusion Detection Classification

		Predicted	
Actual		Anomaly	Normal
	Normal	FP (False positive)	TN (True negative)
	Anomaly	TP (True positive)	FN (False negative)

than 0.5 and as class 0 if it is less than 0.5. Using this method, observations are categorized into several groups. The sigmoid function, F(x) = 1 / (1 + e^{-x}), uses input x to output a value between 0 and 1. In contrast, logistic regression uses the logistic sigmoid function to determine the probability of classifying data into several groups [50].

XGBoost:

DTs are used as baseline learners in the boosting technique known as XGBoost. The GR framework serves as the foundation for this incredibly scalable tree-building system. A ML technique called GR is used to solve issues with grouping, regression, and classification [50].

Gradient Boosting:

GB is an ensemble learning technique that builds a more potent and precise prediction model by utilizing several decision trees. ML techniques known as decision trees provide a structure resembling a tree with binary options at each node. Based on the provided data, the final nodes, referred to as leaves, show the anticipated results. With gradient boosting, the model produces trees incrementally, fixing the shortcomings of the ones that came before it, improving prediction performance overall [51].

4.5. Performance evaluation metrics:

A confusion matrix comparing the expected and predicted categories is shown in Tab 1. Four critical states are calculated, and it is divided into two categories: correct and wrong. Critical performance metrics including accuracy (ACC), precision, recall, and the F1-score are computed using these states. The following is a list of concepts used in the confusion matrix. True Positive (TP): The model correctly identifies an assault as such. True Negative (TN): Normal behavior is appropriately defined by the model as normal. A false positive (FP) occurs when the model misinterprets typical behavior as an attack. False Negative (FN): The model incorrectly classifies an assault as normal because it fails to detect it [52] [32] [42] [53] .

F1-score, accuracy, precision, and recall are examples of performance evaluation measures.

$$accuracy = \frac{TP + TN}{TP + TN + FN + FP} \quad (1)$$

$$precision = \frac{TP}{TP + FP} \quad (2)$$

$$recall = \frac{TP}{TP + FN} \quad (3)$$

$$F1 - score = 2 \times \frac{precision \times recall}{precision + recall} \quad (4)$$

4.6. Experimental Environment:

The system is equipped with two AMD EPYC 7742 processors (64 cores @ 2.25 GHz) and 1 TB of DDR5 ECC memory. It has eight NVIDIA, A100 GPUs (40 GB each), 163,840 CUDA cores, and 6,400 Tensor cores for high-performance computation. The connection is controlled by NVIDIA NVSwitch and routed across four InfiniBand 200 Gbps and two 25GbE ports. Storage consists of six 2.4 TB NVMe SSDs arranged in RAID 0, for a total capacity of 14.4TB.

5. RESULTS AND DISCUSSION

This section summarizes the Individual Classifiers' findings, which were assessed using the UNSW-NB15 dataset. It evaluates the performance of several classifiers, such as RF, DT, GB, XGBoost, NB, SVM, and LR. The assessment is based on standard metrics such as accuracy, precision, recall, and F1-score, and it takes three approaches: complete features, feature selection techniques such as Information Gain and Mutual Information, and Random Forest. These results will be addressed as follows:

```
'id', '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', 'is_ftp_login',
'ct_ftp_cmd', 'ct_flw_http_mthd',
'ct_src_ltm', 'ct_srv_dst', 'is_sm_ips_ports',
'attack_cat', 'label'
```

classificationDecisionTree_report

Metric	Value				
precision	recall	f1-score	support		
0	0.99	1.00	1.00	23869	
1	1.00	0.99	1.00	23868	
accuracy	1.00	47737			
macro	avg	1.00	1.00	1.00	47737
weighted	avg	1.00	1.00	1.00	47737

Figure 2. DT

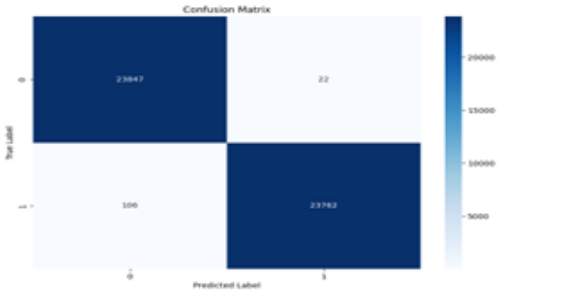

Figure 3. DT

classificationLogisticRegression_report

Metric	Value				
precision	recall	f1-score	support		
0	0.95	0.92	0.94	23869	
1	0.92	0.95	0.94	23868	
accuracy	0.94	47737			
macro	avg	0.94	0.94	0.94	47737
weighted	avg	0.94	0.94	0.94	47737

Figure 4. LR

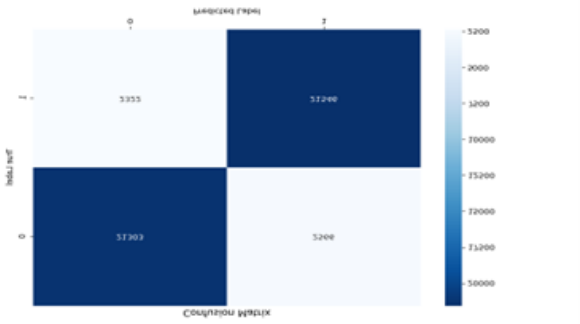

Figure 5. LR

5.1. Individual Classifiers using full features

The performance of standalone classifiers shows that ensemble models routinely beat simpler models in terms of accuracy, precision, recall, and F1-scores.

a. DT
The DT classifier also functioned flawlessly, with an accuracy of 1.00, precision of 1.00, recall of 1.00, and F1-score of 1.00, showing that it can accurately identify both positive and negative examples, as seen in Figure 2 and 3.

b. LR
Figure 4 and 5 demonstrate that the LR had an accuracy of 0.94, as well as a balanced precision, recall, and F1-score.

c. NB
The NB fared the worst, with an accuracy of 0.86 and balanced metrics (precision, recall, and F1-score all 0.86), showing that it struggles with the dataset's intricacies and is less fit for the task than the other classifiers, as seen in Figure 6 and 7.

d. RF
The RF performed flawlessly, with 1.00 accuracy, recall, and F1-score, accurately predicting all outcomes and true positives, as seen in Figure 8 and 9.

e. SVM
Figure 10 and 11 demonstrate that the SVM performed well, with 0.94 accuracy and balanced metrics, indicating that it is a dependable but not top-performing classifier for intrusion detection.

f. XGBoost
The XGBoost obtained remarkable results with 0.99 accuracy, recall, and F1-score, however it performed somewhat worse than Random Forest and Decision Tree, as seen in Figure 12 and 13.

g. GB
The GB achieved an accuracy of 0.9669 and good, albeit somewhat lower, precision, recall, and F1-score. While not as perfect as the best models, it is nonetheless helpful, particularly in cases demanding great accuracy, as seen in Figure 14 and 15.

The following table shows the performance metrics (accuracy, precision, recall, and F1-score) for the several classifiers used in the challenge. Table 2 focuses on individual classifiers.

Table 2 shows the performance evaluation of different classifiers based on Accuracy, Precision, Recall, and F1-score. In all measures, the Random Forest and Decision Tree classifiers obtained perfect 1.00 ratings. With scores of 0.99 for each criteria, XGBoost did somewhat worse. Naive Bayes had the weakest performance, with

classificationNaiveBayes_report

Metric	Value				
precision	recall	f1-score	support		
0	0.91	0.79	0.85	23869	
1	0.82	0.93	0.87	23868	
accuracy	0.86	47737			
macro	avg	0.87	0.86	0.86	47737
weighted	avg	0.87	0.86	0.86	47737

Figure 6. NB

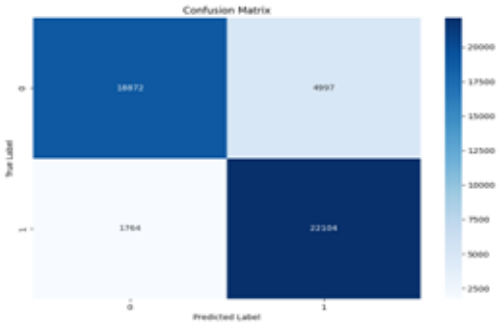

Figure 7. NB

classificationRandomForest_report

Metric	Value				
precision	recall	f1-score	support		
0	1.00	1.00	1.00	23869	
1	1.00	1.00	1.00	23868	
accuracy	1.00	47737			
macro	avg	1.00	1.00	1.00	47737
weighted	avg	1.00	1.00	1.00	47737

Figure 8. RF

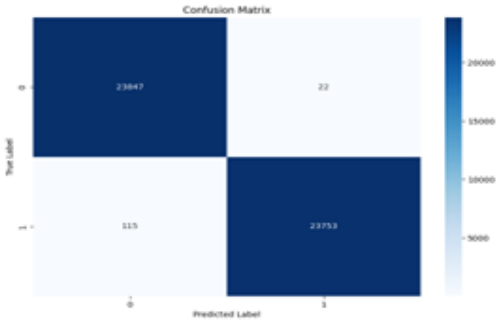

Figure 9. RF

classificationSVM_report

Metric	Value				
precision	recall	f1-score	support		
0	0.95	0.92	0.94	23869	
1	0.92	0.95	0.94	23868	
accuracy	0.94	47737			
macro	avg	0.94	0.94	0.94	47737
weighted	avg	0.94	0.94	0.94	47737

Figure 10. SVM

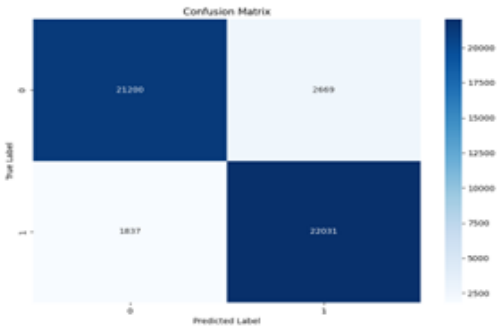

Figure 11. SVM

values around 0.86. Support Vector Machine, Logistic Regression, and Gradient Boosting produced consistent results, with scores ranging from 0.94 to 0.97 across the four criteria.

5.2. Individual Classifiers by using Gain Information and Manual Information

This section compares numerous classifiers using metrics like as accuracy, precision, recall, and F1-score, providing insight into their benefits and downsides for a

classificationXGBoost_report

Metric	Value				
precision	recall	f1-score	support		
0	0.99	0.99	0.99	23869	
1	0.99	0.99	0.99	23868	
accuracy	0.99	47737			
macro	avg	0.99	0.99	0.99	47737
weighted	avg	0.99	0.99	0.99	47737

Figure 12. XGBoost

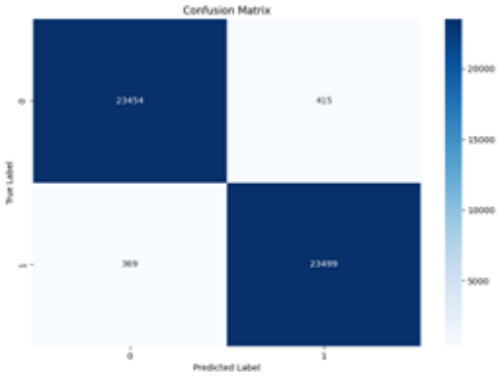

Figure 13. XGBoost

classificationGradientBoosting_report

	precision	recall	f1-score	support	accuracy
0	0.9921398922547030	0.9413046210566010	0.9660539610878210	23869.0	
1	0.9441632457853410	0.9925423160717280	0.9677485242754140	23868.0	
macro avg	0.9681515690200220	0.9669234685641640	0.9669012426816180	47737.0	
weighted avg	0.9681520715300930	0.9669229318976890	0.9669012249326680	47737.0	
0					0.9669229318976890

Figure 14. GB

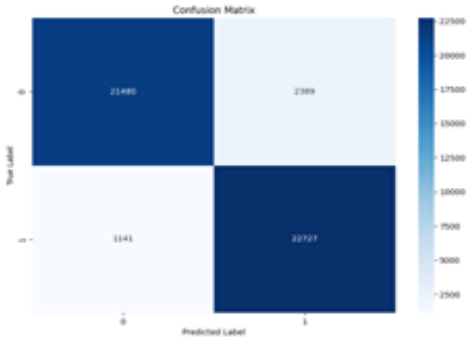

Figure 15. GB

Table[2]: Classifier Performance Evaluation using Accuracy, Precision, Recall, and F1-Score

Classifiers	Accuracy	Precision	Recall	F1-score
Random Forest	1.00	1.00	1.00	1.00
Decision Tree	1.00	1.00	1.00	1.00
XGBoost	0.99	0.99	0.99	0.99
Naive Bayes	0.86	0.87	0.86	0.86
Support Vector Machine	0.94	0.94	0.94	0.94
Logistic Regression	0.94	0.94	0.94	0.94
Gradient Boosting	0.9669	0.9682	0.9669	0.9669

variety of classification problems. Here's a list of features:

```
'sjit', 'dload', 'dinpkt', 'sload', 'id',
'rate', 'sinpkt', 'dur', 'sbytes',
'labelsjit', 'dload', 'dinpkt', 'sload',
'id', 'rate', 'sinpkt', 'dur', 'sbytes',
'label'
```

Individual classifiers were assessed based on four essential metrics: accuracy, precision, recall, and F1-score. The findings provide important insights into the efficacy of classifiers and the advantages of combining multiple models.

a. XGBoost
The XGBoost model performed perfectly, with 0.98 accuracy, 0.98 precision, 0.98 recall, and 0.98 F1 score. This demonstrates that, while not perfect, XGBoost has strong performance and dependable threat detection capabilities, as seen in Figure 16 and 17.

b. DT
The DT classifier performed flawlessly, with 1.00 accuracy, precision, recall, and F1-score, accurately classifying both positive and negative examples, as seen in Figure 18 and 19.

c. RF
Figure 20 and 21 show that the RF classifier performed flawlessly, with 1.00 accuracy, precision, recall, and F1-score, correctly recognizing all positive instances and separating them from negatives.

classificationXGBoost_report

Metric	Value				
precision	recall	f1-score	support		
0	0.98	0.98	0.98	23869	
1	0.98	0.98	0.98	23868	
accuracy	0.98	47737			
macro	avg	0.98	0.98	0.98	47737
weighted	avg	0.98	0.98	0.98	47737

Figure 16. XGBoost

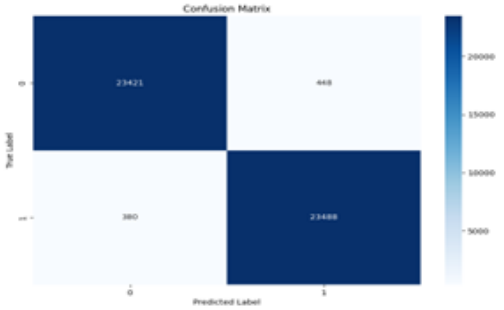

Figure 17. XGBoost

classificationDecisionTree_report

Metric	Value				
precision	recall	f1-score	support		
0	0.99	1.00	1.00	23869	
1	1.00	0.99	1.00	23868	
accuracy	1.00	47737			
macro	avg	1.00	1.00	1.00	47737
weighted	avg	1.00	1.00	1.00	47737

Figure 18. DT

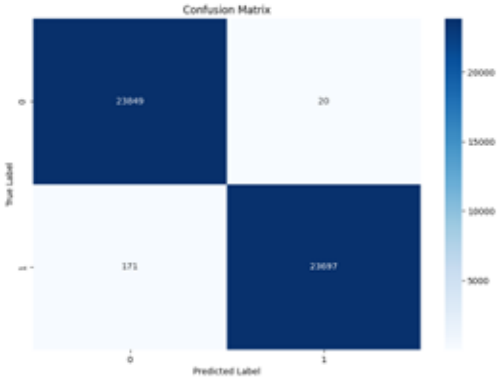

Figure 19. DT

classificationRandomForest_report

Metric	Value				
precision	recall	f1-score	support		
0	0.99	1.00	1.00	23869	
1	1.00	0.99	1.00	23868	
accuracy	1.00	47737			
macro	avg	1.00	1.00	1.00	47737
weighted	avg	1.00	1.00	1.00	47737

Figure 20. RF

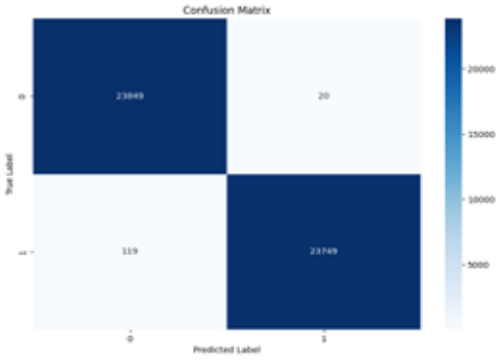

Figure 21. RF

d. SVM

The SVM performed well, with 0.90 accuracy, precision, recall, and F1-score, but was somewhat less effective than models such as Random Forest and Decision Tree, as illustrated in Figure 22 and 23.

e. NB

The NB classifier scored the worst, with 0.77 accuracy, 0.83 precision, 0.77 recall, and 0.76 F1-score, showing that it struggled to recognize threats and was less trustworthy for the job, as seen in Figure 24 and 25.

f. GB

The GB classifier obtained 0.91 accuracy, precision, recall, and F1-score, suggesting a balanced and successful threat detection. Figure 26 and 27 indicate that it outperformed Naive Bayes and Logistic Regression while falling significantly behind XGBoost.

classificationSVM_report

Metric	Value				
precision	recall	f1-score	support		
0	0.90	0.90	0.90	23869	
1	0.90	0.90	0.90	23868	
accuracy	0.90	47737			
macro	avg	0.90	0.90	0.90	47737
weighted	avg	0.90	0.90	0.90	47737

Figure 22. SVM

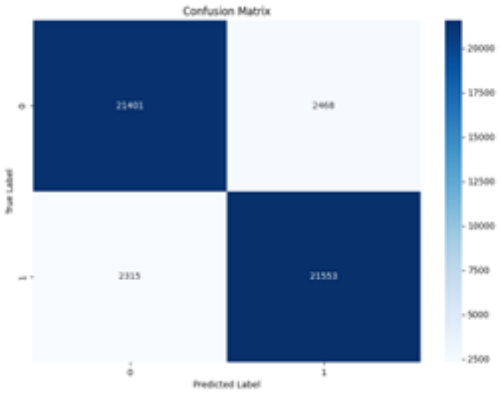

Figure 23. SVM

classificationNaiveBayes_report

Metric	Value				
precision	recall	f1-score	support		
0	0.96	0.57	0.71	23869	
1	0.69	0.98	0.81	23868	
accuracy	0.77	47737			
macro	avg	0.83	0.77	0.76	47737
weighted	avg	0.83	0.77	0.76	47737

Figure 24. NB

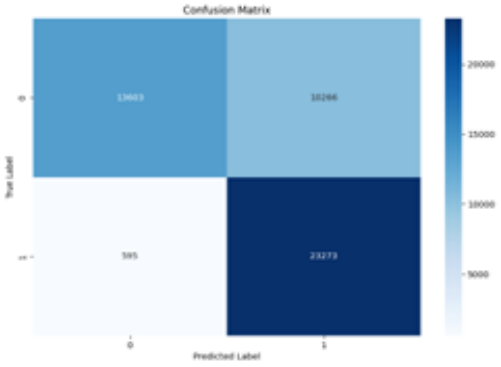

Figure 25. NB

classificationGradientBoosting_report

Metric	Value				
precision	recall	f1-score	support		
0	0.94	0.87	0.90	23869	
1	0.88	0.94	0.91	23868	
accuracy	0.91	47737			
macro	avg	0.91	0.91	0.91	47737
weighted	avg	0.91	0.91	0.91	47737

Figure 26. GB

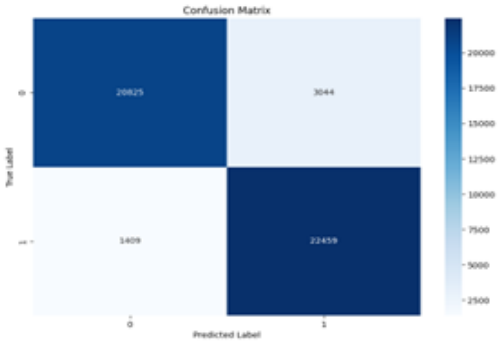

Figure 27. GB

g. LR

Figure 28 and 29 indicate that Logistic Regression works pretty well, with reduced Accuracy (0.83) but acceptable Precision (0.86), Recall (0.83), and F1-Score (0.83).

Table 3 compares individual classifiers' performance in terms of accuracy, precision, recall, and F1-score, as well as their efficiency.

Table 3 shows that individual classifiers perform differently, with Random Forest and Decision Tree getting perfect scores (1.00 across all categories), indicating flawless classification. XGBoost outscored Naive Bayes, which had a 0.77 accuracy and lower related metrics. Support Vector Machine and Logistic Regression had reasonable accuracy scores of 0.90 and 0.83, respectively.

classificationLogisticRegression_report

Metric	Value				
precision	recall	f1-score	support		
0	0.97	0.68	0.80	23869	
1	0.76	0.98	0.85	23868	
accuracy	0.83	47737			
macro	avg	0.86	0.83	0.83	47737
weighted	avg	0.86	0.83	0.83	47737

Figure 28. LR

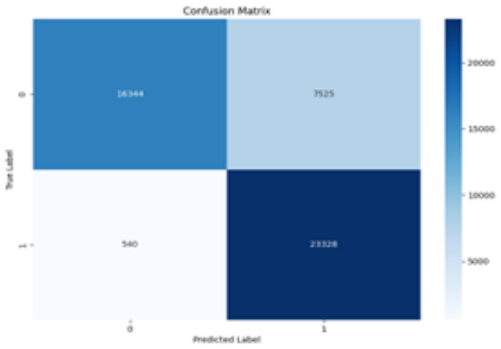

Figure 29. LR

Table[3]: Classifier Performance Evaluation using Accuracy, Precision, Recall, and F1-Score

Classifiers	Accuracy	Precision	Recall	F1-score
Random Forest	1.00	1.00	1.00	1.00
Decision Tree	1.00	1.00	1.00	1.00
XGBoost	0.98	0.98	0.98	0.98
Naive Bayes	0.77	0.83	0.77	0.76
Support Vector Machine	0.90	0.90	0.90	0.90
Logistic Regression	0.83	0.86	0.83	0.83
Gradient Boosting	0.91	0.91	0.91	0.91

5.3. Individual Classifiers by using Random Forest

```
'sttl', 'ct_state_ttl', 'dload', 'rate',
'dmean','dttl', 'tcprtt', 'log_duration',
'sload', 'dinpkt', 'attack'
```

The features picked by RF are noted below.This study compares the performance of individual classifiers using RF-selected features based on accuracy, precision, recall, and F1-score.

a.DT

The DT classifier had an accuracy of 0.93, with precision, recall, and F1-score values around 0.92. These data reveal that Decision Tree is a high performer, however its recall of 0.92 indicates that it may miss a few occurrences, as seen in Figure 30 and 31.

b. RL

Logistic Regression achieved 0.92 accuracy, 0.93 precision, 0.88 recall, and a 0.90 F1-score. While it scored well in accuracy, its lower recall shows that the model may have missed certain incursions, despite providing a balanced overall performance, as seen in Figure 32 and 33.

c. NB

Naive Bayes fared badly, with 0.89 accuracy, 0.89 precision, 0.86 recall, and a 0.87 F1 score. Its poor recall indicates that it misses many genuine threats, rendering it unsuitable for efficient intrusion detection, as seen in Figure 34 and 35.

d. RF

Random Forest had 0.94 accuracy, 0.94 precision, 0.92 recall, and a 0.93 F1-score. These findings demonstrate that it effectively identifies most infiltration attempts while balancing precision and recall, with high precision suggesting a good capacity to reduce false positives, as seen in Figure 36 and 37.

e. GB

Gradient Boosting achieved 0.9308 accuracy, 0.9524 precision, 0.8922 recall, and a 0.9150 F1-score. Despite having the best precision, the lower recall indicates that it may be more cautious in spotting threats. However, its overall performance remained outstanding, particularly in terms of accuracy, as seen in Figure 38 and 39.

f. SVM

The Support Vector Machine attained an accuracy of 0.87, with precision, recall, and F1-score of 0.86. While its accuracy was adequate, its lower recall rendered it more susceptible to false negatives, making it unsuitable for intrusion detection, where strong recall is critical, as seen in Figure 40 and 41.

g. XGBoost

XGBoost worked well, with 0.94 accuracy, 0.94 precision, 0.92 recall, and 0.93 F1-score. Its constant performance across all parameters demonstrates that it is very dependable for intrusion detection, comparable to Random Forest and Decision Tree in terms of accuracy and threat identification, as seen in Figure 42 and 43.

classificationDecisionTree_report

Metric	Value				
precision	recall	f1-score	support		
0	0.89	0.89	0.89	11200	
1	0.95	0.95	0.95	23869	
accuracy	0.93	35069			
macro	avg	0.92	0.92	0.92	35069
weighted	avg	0.93	0.93	0.93	35069

Figure 30. DT

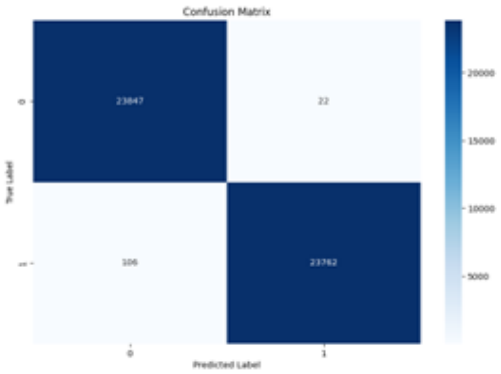

Figure 31. DT

classificationLogisticRegression_report

Metric	Value				
precision	recall	f1-score	support		
0	0.96	0.77	0.86	11200	
1	0.90	0.98	0.94	23869	
accuracy	0.92	35069			
macro	avg	0.93	0.88	0.90	35069
weighted	avg	0.92	0.92	0.91	35069

Figure 32. LR

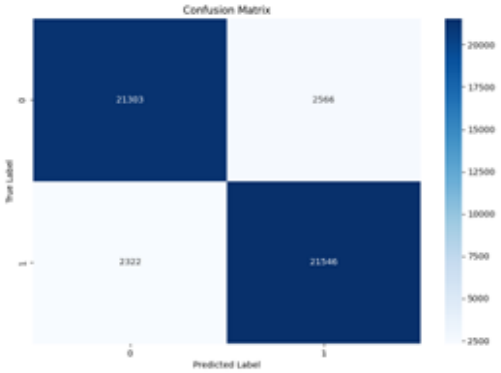

Figure 33. LR

classificationNaiveBayes_report

Metric	Value				
precision	recall	f1-score	support		
0	0.88	0.76	0.82	11169	
1	0.90	0.95	0.92	23900	
accuracy	0.89	35069			
macro	avg	0.89	0.86	0.87	35069
weighted	avg	0.89	0.89	0.89	35069

Figure 34. NB

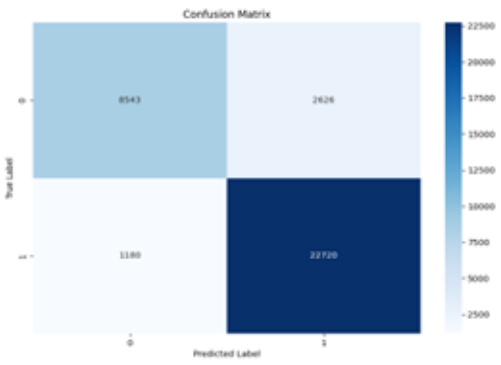

Figure 35. NB

Table 4 compares the performance of individual classifiers in terms of accuracy, precision, recall, and F1. This table compares the advantages and disadvantages of each intrusion detection model

The performance of several classifiers is compared in Table 4 according to F1-score, recall, accuracy, and precision. With precision, recall, and F1 values high and accuracy scores of 0.94, Random Forest and XGBoost demonstrated excellent performance. Despite having far worse outcomes, Decision Tree also did well. Across all categories, Naive Bayes performed the worst, while Support Vector Machine and Logistic Regression had average scores. Gradient Boosting received high scores on every evaluation criteria, performing comparable to Random Forest and XGBoost.

classificationRandomForest_report

Metric	Value				
precision	recall	f1-score	support		
0	0.94	0.87	0.90	11169	
1	0.94	0.97	0.96	23900	
accuracy	0.94	35069			
macro	avg	0.94	0.92	0.93	35069
weighted	avg	0.94	0.94	0.94	35069

Figure 36. RF

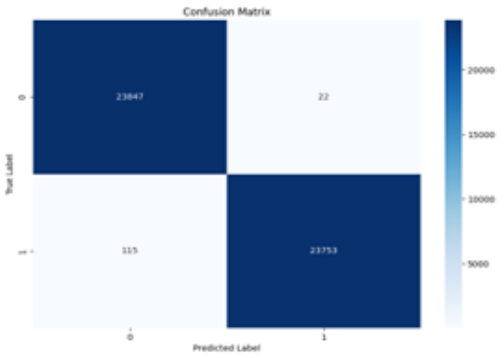

Figure 37. RF

classificationGradientBoosting_report

	precision	recall	f1-score	support	accuracy
0	0.995802609188883O	0.785925329035724O	0.878502802241793O	11169.0	
1	0.908928163327493O	0.998451882845188O	0.951589105554891	23900.0	
macro avg	0.952365386258188O	0.892188605940456	0.915045938983420	35069.0	
weighted avg	0.936596493927905O	0.930765063161196O	0.928312111009737O	35069.0	
0					0.930765063161196O

Figure 38. GB

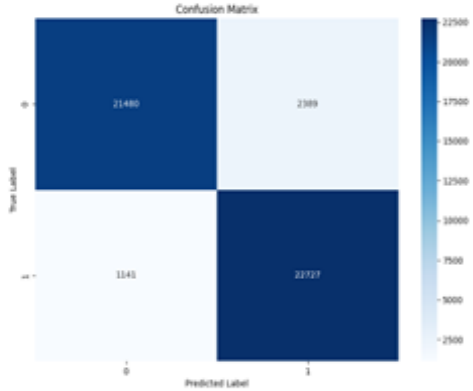

Figure 39. GB

classificationSVM_report

Metric	Value				
precision	recall	f1-score	support		
0	0.80	0.81	0.80	11200	
1	0.91	0.91	0.91	23869	
accuracy	0.87	35069			
macro	avg	0.86	0.86	0.86	35069
weighted	avg	0.87	0.87	0.87	35069

Figure 40. SVM

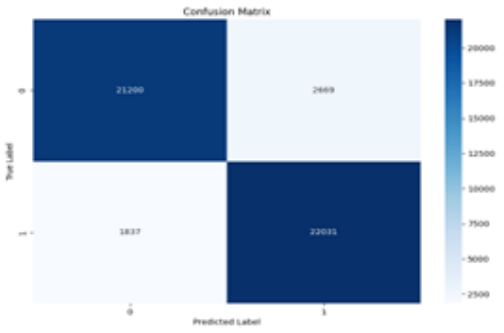

Figure 41. SVM

Table[4]: Classifier Performance Evaluation using Accuracy, Precision, Recall, and F1-Score

Classifiers	Accuracy	Precision	Recall	F1-score
Random Forest	0.94	0.94	0.92	0.93
Decision Tree	0.93	0.92	0.92	0.92
XGBoost	0.94	0.94	0.92	0.93
Naive Bayes	0.89	0.89	0.86	0.87
Support Vector Machine	0.87	0.86	0.86	0.86
Logistic Regression	0.92	0.93	0.88	0.90
Gradient Boosting	0.9308	0.9524	0.8922	0.9150

classificationXGBoost_report

Metric	Value				
precision	recall	f1-score	support		
0	0.94	0.86	0.90	11200	
1	0.94	0.97	0.96	23869	
accuracy	0.94	35069			
macro	avg	0.94	0.92	0.93	35069
weighted	avg	0.94	0.94	0.94	35069

Figure 42. XGBoost

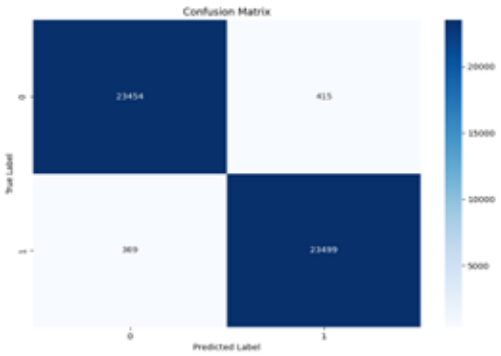

Figure 43. XGBoost

5.4. Comparison of Accuracy and Feature Selection Methods Across Different Studies and the Proposed Model on the - UNSW-NB15 dataset

The UNSW-NB15 dataset has been used in a number of studies to examine the effectiveness of different ML models and feature selection techniques for intrusion detection.RF consistently performed exceptionally well, occasionally achieving 100% accuracy. An accuracy of up to 98.39% was attained by a research [54] that examined RF employing a range of feature sets, including entire features and selected features using Particle Swarm Optimization (PSO), Correlation-based Selection (CS), Information Gain (IG), and combinations of these approaches. In comparison to classifiers like Logistic Regression (LR), K-Nearest Neighbors (KNN), Decision Tree (DT), and Extra Trees using a K-best feature selection approach (K-FS), a research [55] found that Extra Trees obtained 97.53% accuracy.

Study [34] indicated that when ensemble techniques such as RF, Adaboost, Bagging, and LogitBoost were compared with PCA, RF outperformed Adaboost by 100%, but Adaboost's performance was much inferior (67.9%).

In a research [56] that compared RF to Naive Bayes (NB) utilizing full features, RF scored 87.08% while NB came in second at 46.16%.

The proposed model evaluated a number of classifiers, including RF, DT, XGBoost, NB, SVM, LR, and GB. Both RF and DT achieved perfect accuracy (1.00) using all criteria. When certain features were used (either through mutual information, information gain, or RF-based selection), there was a little drop in performance; NB remained the poorest overall, while RF continued to perform best with 0.94.

6. CONCLUSION AND FEATURES:

The CC offers scalable and affordable services, its complexity exposes it to a number of security threats. By identifying unauthorized access and harmful behavior, the IDS helps safeguard cloud infrastructures. However, standard IDS are useless due to the constantly shifting nature of cloud settings. This study tries to enhance intrusion detection in CC security by using multiple ML approaches to detect both known and novel cyber threats. We examine a variety of popular classifiers using the UNSW-NB15 dataset, such as RF, DT, XGBoost, NB, SVM, LR, and GB. The study's conclusions demonstrate the significant benefits of using ML methods for cloud intrusion detection. With nearly flawless accuracy, precision, recall, and F1-scores especially when using the whole feature set ,RF and DT were the classifiers that performed the best across all feature selection techniques. Although Random Forest-based selection maintained high accuracy, feature selection techniques had an impact on performance, with complete features often producing superior outcomes. These findings highlight how ML models outperform rule-based systems in identifying both conventional and new attack patterns. The security of cloud-based systems may be greatly improved by utilizing datasets such as UNSW-NB15 and putting strong ML techniques into practice. To further increase detection rates and flexibility, future research may employ deep learning models and real-time IDS deployment.

Declarations

Ethical Approval

This article does not contain any studies involving animals or human participants performed by any of the authors. We declare this manuscript is original, and is not currently considered for publication elsewhere. We further confirm that the order of authors listed in the manuscript has been approved by all of us.

Table[5]: Comparison of feature selection and ML techniques on UNSW-NB15 dataset

Research	Dataset	ML Technique	Accuracy	Feature selection
[54]	UNSW-NB15	RF	96.49 98.02 97.61 97.99 98.39	Full features 18 features by IG 32 features by CS 25 features by PSO 21 features by IG+CS+PSO
[55]	UNSW-NB-15	LR KNN DT Extra Tree	92.85 95.04 96.33 97.53	K-fS(K-best features)
[34]	UNSW-NB-15	Random Forest Adaboost Bagging Logitboost	100 67.9 93.4 88.7	PCA
[56]	UNSW-NB15	RF NB	87.08 46.16	Full features
Proposed model	UNSW-NB15	RF DT XGBoost NB SVM LR GB	1.00 1.00 0.99 0.86 0.94 0.94 0.9669	Full features
		RF DT XGBoost NB SVM LR GB	1.00 1.00 0.98 0.77 0.90 0.83 0.91	Selected features used mutual information and gain information
		RF DT XGBoost NB SVM LR GB	0.94 0.93 0.94 0.89 0.87 0.92 0.9308	Selected features by Random forest

Competing interests

The authors have no financial or proprietary interests in any material discussed in this article.

Authors' contributions

All authors contributed to the study conception and design. All authors performed simulations, data collection and analysis and commented the present version of the manuscript. All authors read and approved the final manuscript.

Funding: No funding is received from any organization for this work.

Availability of data and materials

No datasets is used in the present study.

Consent for publication

The authors confirm that there is informed consent to the publication of the data contained in the article.

Consent to participate

Informed consent was obtained from all authors.

REFERENCES

[1] R. Bingu and S. Jothilakshmi, "Design of intrusion detection system using ensemble learning technique in cloud computing environment," *Int. J. Adv. Comput. Sci. Appl.*, 2024, [Online]. Available: www.ijacsa.thesai.org.

[2] M. Marwan, A. Kartit, and H. Ouahmane, "Security enhancement in healthcare cloud using machine learning," *Procedia Comput. Sci.*, vol. 00, 2018. DOI: 10.1016/j.procs.2018.01.136.

[3] N. A. Banu and S. K. B. Sangeetha, "Intrumer: A multi module distributed explainable ids/ips for securing cloud environment," *Comput. Mater. Continua*, vol. 82, no. 1, pp. 579–607, 2025. DOI: 10.32604/cmc.2024.059805.

[4] M. Al-Sharif and A. Bushnag, "Enhancing cloud security: A study on ensemble learning-based intrusion detection systems," *IET Commun.*, pp. 1–16, May 2024. DOI: 10.1049/cmu2.12801.

[5] A. Adhikari and B. K. Bal, "Machine learning technique for intrusion detection in the field of the intrusion detection system," 2023, Unpublished manuscript, July 2023.

[6] A. B. Nassif, M. A. Talib, Q. Nasir, H. Albadani, and F. M. Dakalbab, "Machine learning for cloud security: A systematic review," *IEEE Access*, 2021. DOI: 10.1109/ACCESS.2021.3054129.

[7] M. H. Behiry and M. Aly, "Cyberattack detection in wireless sensor networks using a hybrid feature reduction technique with ai and machine learning methods," *J. Big Data*, vol. 11, no. 1, Dec. 2024. DOI: 10.1186/s40537-023-00870-w.

[8] G. Logeswari, S. Bose, and T. Anitha, "An intrusion detection system for sdn using machine learning," *Intell. Autom. Soft Comput.*, vol. 35, no. 1, pp. 867–880, 2023. DOI: 10.32604/iasc.2023.026769.

[9] A. Alotaibi and M. A. Rassam, "Adversarial machine learning attacks against intrusion detection systems: A survey on strategies and defense," *Future Internet*, Feb. 2023. DOI: 10.3390/fi15020062.

[10] M. A. Umar, Z. Chen, K. Shuaib, and Y. Liu, "Effects of feature selection and normalization on network intrusion detection," *Data Sci. Manag.*, vol. 8, no. 1, pp. 23–39, 2024. DOI: 10.1016/j.dsm.2024.08.001.

[11] T. Saranya, S. Sridevi, C. Deisy, T. D. Chung, and M. K. A. A. Khan, "Performance analysis of machine learning algorithms in intrusion detection system: A review," *Procedia Comput. Sci.*, vol. 171, pp. 1251–1260, 2020. DOI: 10.1016/j.procs.2020.04.133.

[12] S. M. Othman, A. Y. Al-mutawkkil, and A. M. Alnashi, "Survey of intrusion detection techniques in cloud computing," *Sana'a Univ. J. Appl. Sci. Technol.*, vol. 2, no. 4, pp. 363–374, 2024. DOI: 10.59628/jast.v2i4.970.

[13] D. E. Useni, O. C. Emmanuel, G. K. Job, and A. Ahmad, "A review of machine learning-based algorithms for intrusion detection system," vol. 12, no. 1, pp. 251–256, 2023.

[14] I. Hidayat, M. Z. Ali, and A. Arshad, "Machine learning-based intrusion detection system: An experimental comparison," *J. Comput. Cogn. Eng.*, vol. 2, no. 2, pp. 88–97, 2023. DOI: 10.47852/bonviewJCCE2202270.

[15] Z. Ahmad, A. S. Khan, C. W. Shiang, J. Abdullah, and F. Ahmad, "Network intrusion detection system: A systematic study of machine learning and deep learning approaches," *Trans. on Emerg. Telecommun. Technol.*, vol. 32, no. 1, Jan. 2021. DOI: 10.1002/ett.4150.

[16] S. Wang, J. F. Balarezo, S. Kandeepan, A. Al-Hourani, K. G. Chavez, and B. Rubinstein, "Machine learning in network anomaly detection: A survey," *IEEE Access*, vol. 9, pp. 152 379–152 396, 2021. DOI: 10.1109/ACCESS.2021.3126834.

[17] S. Jadhav, V. Bhalerao, V. Yadav, S. Kamble, and B. Shinde, "Network intrusion detection system using machine learning," in *Conference Proceedings*, vol. 3307, 2023, pp. 74–81.

[18] M. M. Belal and D. M. Sundaram, "Comprehensive review on intelligent security defences in cloud: Taxonomy, security issues, ml/dl techniques, challenges and future trends characteristic features based neural network," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 34, no. 10, pp. 9102–9131, 2022. DOI: 10.1016/j.jksuci.2022.08.035.

[19] M. Saran, R. K. Yadav, and U. N. Tripathi, "Machine learning based security for cloud computing: A survey," vol. 17, no. 4, pp. 332–337, 2022.

[20] S. Ahmadi, "Systematic literature review on cloud computing security: Threats and mitigation strategies," *J. Inf. Secur.*, vol. 15, no. 2, pp. 148–167, 2024. DOI: 10.4236/jis.2024.152010.

[21] A. R. Sonule, M. Kalla, A. Jain, and D. S. Chouhan, "Unsw-nb15 dataset and machine learning based intrusion detection systems," *Int. J. Eng. Adv. Technol.*, vol. 9, no. 3, pp. 2638–2648, 2020. DOI: 10.35940/ijeat.c5809.029320.

[22] Y. I. Alzoubi, A. Mishra, and A. E. Topcu, "Research trends in deep learning and machine learning for cloud computing security," *Artif. Intell. Rev.*, vol. 57, no. 5, 2024. DOI: 10.1007/s10462-024-10776-5.

[23] P. R. Kumar, P. H. Raj, and P. Jelciana, "Exploring data security issues and solutions in cloud computing," *Procedia Comput. Sci.*, vol. 125, pp. 691–697, 2018. DOI: 10.1016/j.procs.2017.12.089.

[24] V. Chang, L. Golightly, P. Modesti, Q. A. Xu, L. Minh, and T. Doan, "A survey on intrusion detection systems for fog and cloud computing," 2022, Unpublished manuscript or preprint.

[25] M. K. Sinchana and R. M. Savithramma, "Survey on cloud computing security," in *Lecture Notes in Networks and Systems*, 4, vol. 103, 2020, pp. 1–6. DOI: 10.1007/978-981-15-2043-3_1.

[26] B. E. Ricks, "Intrusion detection systems," in *Physical Security and Safety: A Field Guide for the Practitioner*, CRC Press, 2014, pp. 101–108. DOI: 10.4018/jdm.338276.

[27] T. A. Devi and A. Jain, "Enhancing cloud security with deep learning-based intrusion detection in cloud computing environments," in *Proceedings of the 2nd International Conference on Advancement in Computation & Computer Technologies (InCACCT)*, 2024, pp. 541–546. DOI: 10.1109/InCACCT61598.2024.10551040.

[28] D. P. R. Sanagana and C. K. Tummalachervu, "Securing cloud computing environment via optimal deep learning-based intrusion detection systems," in *Proceedings of the 2nd International Conference on Data Science and Information System (ICDSIS)*, 2024, pp. 1–6. DOI: 10.1109/ICDSIS61070.2024.10594404.

[29] J. K. Samriya, S. Kumar, M. Kumar, H. Wu, and S. S. Gill, "Machine learning-based network intrusion detection optimization for cloud computing environments," *IEEE Trans. on Consumer Electron.*, vol. 70, no. 4, pp. 7449–7460, 2024. DOI: 10.1109/TCE.2024.3458810.

[30] P. Rana et al., "Intrusion detection systems in cloud computing paradigm: Analysis and overview," *Hindawi J.*, 2022. DOI: 10.1155/2022/3999039.

[31] A. John, I. F. B. Isnin, S. H. H. Madni, and F. B. Muchtar, "Enhanced intrusion detection model based on principal component analysis and variable ensemble machine learning algorithm," *Intell. Syst. with Appl.*, vol. 24, p. 200 442, Feb. 2024. DOI: 10.1016/j.iswa.2024.200442.

[32] A. D. Vibhute, M. Khan, C. H. Patil, S. V. Gaikwad, A. V. Mane, and K. K. Patel, "Network anomaly detection and performance evaluation of convolutional neural networks on unsw-nb15 dataset," *Procedia Comput. Sci.*, vol. 235, pp. 2227–2236, 2024. DOI: 10.1016/j.procs.2024.04.211.

[33] M. S. Al-Daweri, K. A. Z. Ariffin, S. Abdullah, and M. F. E. M. Senan, "An analysis of the kdd99 and unsw-nb15 datasets for the intrusion detection system," *Symmetry (Basel)*, vol. 12, no. 10, pp. 1–32, 2020. DOI: 10.3390/sym12101666.

[34] S. S. Tripathy and B. Behera, "Performance evaluation of machine learning," pp. 621–640, Apr. 2023. DOI: 10.17605/OSF.IO/WX6CS.

[35] S. Liu and M. Motani, *Improving mutual information based feature selection by boosting unique relevance*, [Online]. Available: http://arxiv.org/abs/2212.06143, 2022.

[36] M. Alalhareth and S. Hong, *An improved mutual information feature selection technique*, 2023.

[37] Z. Azam, M. Islam, and M. N. Huda, "Comparative analysis of intrusion detection systems and machine learning-based model analysis through decision tree," vol. 11, Jul. 2023.

[38] J. L. Solorio-Ramírez, R. Jiménez-Cruz, Y. Villuendas-Rey, and C. Yáñez-Márquez, "Random forest algorithm for the classification of spectral data of astronomical objects," *Algorithms*, vol. 16, no. 6, 2023. DOI: 10.3390/a16060293.

[39] M. A. Khan and Y. Kim, "Deep learning-based hybrid intelligent intrusion detection system," *Comput. Mater. Continua*, vol. 68, no. 1, pp. 671–687, Mar. 2021. DOI: 10.32604/cmc.2021.015647.

[40] A. Meryem, *Hybrid intrusion detection system using machine learning*, [Online]. Available: www.idg.com/tools-for-, 2020.

[41] R. Satya and S. Dittakavi, "Dimensionality reduction based intrusion detection system in cloud computing environment using machine learning," *Int. J. Inf. Cybersecur.*,

[42] S. Latif, F. F. Dola, M. Afsar, I. J. Esha, and D. Nandi, "Investigation of machine learning algorithms for network intrusion detection," *Int. J. Inf. Eng. Electron. Bus.*, vol. 14, no. 2, pp. 1–22, Apr. 2022. DOI: 10.5815/ijieeb.2022.02.01.

[43] M. Mehmood, R. Amin, M. Magboul, A. L. I. Muslam, J. Xie, and H. Aldabbas, "Privilege escalation attack detection and mitigation in cloud using machine learning," *IEEE Access*, vol. 11, pp. 46 561–46 576, Apr. 2023. DOI: 10.1109/ACCESS.2023.3273895.

[44] V. Pai, Devidas, and N. D. Adesh, "Comparative analysis of machine learning algorithms for intrusion detection," in *IOP Conference Series: Materials Science and Engineering*, vol. 1013, 2021. DOI: 10.1088/1757-899X/1013/1/012038.

[45] A. Aldallal and F. Alisa, "Effective intrusion detection system to secure data in cloud using machine learning," *Symmetry (Basel)*, vol. 13, no. 12, 2021. DOI: 10.3390/sym13122306.

[46] M. Labonne, *Anomaly-based network intrusion detection using machine learning*, [Online]. Available: https://theses.hal.science/tel-02988296, 2020.

[47] Z. Somogyi, "Performance evaluation of machine learning models," in *The Application of Artificial Intelligence*, Apr. 2021, pp. 87–112. DOI: 10.1007/978-3-030-60032-7_3.

[48] D. Boldini, F. Grisoni, D. Kuhn, L. Friedrich, and S. A. Sieber, "Practical guidelines for the use of gradient boosting for molecular property prediction," *J. Cheminformatics*, vol. 15, no. 1, pp. 1–13, 2023. DOI: 10.1186/s13321-023-00743-7.

[49] H. Attou et al., "Towards an intelligent intrusion detection system to detect malicious activities in cloud computing," *Appl. Sci. (Switzerland)*, vol. 13, no. 17, Sep. 2023. DOI: 10.3390/app13179588.

[50] H. Attou, A. Guezzaz, S. Benkirane, M. Azrour, and Y. Farhaoui, "Cloud-based intrusion detection approach using machine learning techniques," *Big Data Min. Anal.*, vol. 6, no. 3, pp. 311–320, Sep. 2023. DOI: 10.26599/BDMA.2022.9020038.

[51] M. Labonne, *Anomaly-based network intrusion detection using machine learning*, [Online]. Available: https://theses.hal.science/tel-02988296, 2023.

[52] M. Bakro et al., "An improved design for a cloud intrusion detection system using hybrid features selection approach with ml classifier," *IEEE Access*, vol. 11, pp. 64 228–64 247, Jun. 2023. DOI: 10.1109/ACCESS.2023.3289405.

[53] P. Parameswarappa, T. Shah, and G. R. Lanke, "A machine learning-based approach for anomaly detection for secure cloud computing environments," in *Proceedings of the International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT 2023)*, 2023, pp. 931–940. DOI: 10.1109/IDCIoT56793.2023.10053518.

[54] A. Das, S. A. Ajila, and C. H. Lung, "A comprehensive analysis of accuracies of machine learning algorithms for network intrusion detection," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 12081, 2020, pp. 40–57. DOI: 10.1007/978-3-030-45778-5_4.

[55] P. Parameswarappa, T. Shah, and G. R. Lanke, "A machine learning-based approach for anomaly detection for secure cloud computing environments," in *Proceedings of the International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT 2023)*, 2023, pp. 931–940. DOI: 10.1109/IDCIoT56793.2023.10053518.

[56] A. Das, S. A. Ajila, and C. H. Lung, "A comprehensive analysis of accuracies of machine learning algorithms for network intrusion detection," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 12081, 2020, pp. 40–57. DOI: 10.1007/978-3-030-45778-5_4.