Mesopotamian journal of Cybersecurity
Vol.2023, **pp**. 560–571
DOI: https://doi.org/10.58496/MJCS/2022/019; ISSN: 2958-6542
https://mesopotamian.press/journals/index.php/cybersecurity

Research Article

Cyber-Attack Detection for Cloud-Based Intrusion Detection Systems

Muna Ismael Shihan Al-jumaili [1], Dr. Jad Bazzi [1,*]

[1] *American University of Culture and Education (AUCE), Lebanon.*

ARTICLE INFO

Article History
Received 20 August 2023
Accepted 27 Oct. 2023
Published 2 Nov. 2023

Cloud Computing
Machine Learning
Intrusion Detection System
Cyber-attack

ABSTRACT

In this research, we delve into an exhaustive ex- amination of the usage of machine learning (ML) models for the identification of cyber threats within Cloud-Based Intrusion Detection Systems (IDS). With the escalating dependence on cloud services across various industries, the urgency to develop effective and resilient IDS to mitigate burgeoning cyber risks is paramount. Our study makes a significant contribution to this imperative field by analyzing the efficacy of diverse ML models in detecting cyber-attacks within a cloud context. We scrutinized an array of ML models, namely Decision Trees (DT), Random Forest (RF), XGBoost, and Support Vector Machines (SVM), utilizing key performance parameters such as accuracy, recall, precision, F1-s, and confusion matrix to understand their practical application in real-world IDS scenarios. XGBoost stood out as the most proficient model, showcasing not only an impressive accuracy but also a balanced performance in terms of precision and recall. This highlights the considerable potential of ensemble and gradient boosting techniques in optimizing cloud- based IDS detection capabilities. Our findings underscore the significant role of machine learning in fostering more dependable, robust, and efficient IDS in the cloud, thus significantly aiding in securing our digital ecosystems.

1. INTRODUCTION

Cloud computing (CC) is a paradigm that enables the supply of computer resources such as storage, processing, and applications on-demand through the Internet. Scalability, agility, cost-efficiency, and productivity are just a few of the advantages of cloud computing for businesses. However, as cloud systems and data are exposed to different internal and external threats, such as malevolent insiders, hackers, and cybercriminals [1], [2], CC poses new problems and dangers for cybersecurity.

One of the most difficult aspects of protecting cloud systems is detecting and mitigating cyber-attacks that try to compromise the confidentiality, integrity, or availability of cloud services and data. Cyber-attacks can take several forms and have diverse objectives, as well as damage distinct levels of the cloud stack, such as infrastructure, platform, or software [3], [4]. DDoS assaults, data breaches, ransomware, malware injection, cross-site scripting (XSS), SQL injection, and man- in-the-middle (MITM) attacks are some examples of common cyber-attacks against cloud systems [5].

To identify known attack patterns or abnormalities in net- work traffic or system behavior, traditional CAD approaches depend on predetermined rules or signatures. However, when applied to cloud systems, these technologies have numerous disadvantages. For starters, they are incapable of dealing with the dynamic and diverse character of cloud systems, which necessitate regular upgrades and changes to configuration and rules. Second, they are unable of detecting unique or unknown assaults that do not conform to any current rules or signatures. Third, they produce a huge number of false positives or negatives, reducing detection accuracy and efficiency [6], [7]. To circumvent these restrictions, machine learning (ML) approaches have been offered as a possible option for CAD in cloud-based systems. ML is a subfield of artificial intelligence (AI) that allows computers to learn from data and improve their performance without the need for explicit programming. There are two types of ML techniques: supervised and unsupervised. Supervised ML approaches employ labeled data to train models that can categorize or predict new data based on previously learnt characteristics or patterns. Unsupervised ML approaches employ unlabeled data to uncover hidden patterns or clusters in data without previous information [7].

ML approaches have the potential to provide various ad- vantages for CAD in cloud-based systems. To begin, they can adapt to the dynamic and complicated nature of cloud systems by learning from new data and changing their models as needed. Second, by recognizing abnormalities or outliers that depart from regular behavior or patterns, they can detect unknown or zero-day threats. Third, they can reduce the amount of false positives and negatives by improving detection accuracy and precision [1], [8].

*Corresponding author. Email: j.k.bazzi@gmail.com

We propose in this thesis to compare several ML models for CAD in cloud-based IDS. IDS systems monitor network traffic or system activity and notify users or administrators when suspicious or harmful events occur. We concentrate on random forest (RF), decision tree (DT) [9], XGBoost [10] and support vector machine (SVM) [11] ML models. These models are evaluated using performance measures such as ac- curacy (ACC), recall (REC), precision (PREC), F1-score (F1- s), and confusion matrix (CM). We also compare and contrast their computational complexity, scalability, interpretability, resilience, and generalization [12].

2. RELATED WORKS

The author of [13] offers a cloud-based intrusion detection model that uses the random forest (RF) algorithm and feature engineering to improve detection system accuracy. The paper examines how intrusion detection systems (IDSs) may be used to monitor and detect aberrant network activity. The proposed model is analyzed and verified using two datasets (Bot-IoTand NSL-KDD), reaching high accuracy rates of 98.3% and 99.99%, respectively. The findings suggest that the proposed model surpasses current relevant efforts in terms of accuracy, precision, and recall.

The authors of [14] emphasize the shortcomings of existing IDSs in cloud-based systems, notably in identifying unknown or novel assaults, which frequently result in significant false alarm rates. They also point out that minimizing false alarms can result in higher computing difficulties, as shown in geneticalgorithm- and artificial neural network (ANN)-based IDSs. The authors offer a strong data-driven approach to cloud security to handle concerns such as zero-day threats. They highlight the need of linking occurrences as well as inferring contexts and evidence in order to improve monitoring and decision-making capacities. They present a novel data-driven framework in this research that uses ontology and a knowledgebase to better cyber-attack detection (CAD) via an IDS in the cloud.

The paper [15] presents a cloud-based IDS for FinTech databased on an IoT federated learning architecture and smart contract analysis. The study's goal is to look at data sharing inthe 5G era and how it affects the performance of ML models. The authors provide a unique approach that makes use ofa cyber-threat federated graphical authentication system and cloud-based smart contracts. As their qualifications, participants in the examination built routes on a globe map. When tested on several FinTech cyber-attack datasets, the proposed approach achieves good accuracy (95%), precision (85%), re- call (68%), F-measure (83%), AUC (79%), trust value (65%), scalability (91%), and integrity (83%). The results show that the suggested technique is successful at identifying intrusions in FinTech data.

The research [16] describes an ensemble learning and fog- cloud architecture-driven methodology for identifying cyber-attacks in IoMT networks. Given the rising frequency and severity of cyber-attacks, it tackles the limits of existing healthcare systems and emphasizes the necessity for effective security measures in the IoMT setting. For CAD, the suggested framework incorporates the DT, Naive Bayes, RF, and XGBoost algorithms. It also proposes a deployment strategy geared to the dynamic and diverse nature of IoMT networks, leveraging Software as a Service (SaaS) on the fog side and Infrastructure as a Service (IaaS) on the cloud side. In contrast to prior assessments, which employed old datasets,the suggested model makes use of a realistic dataset named ToN-IoT, which was obtained from a large-scale IoT network.The results of the experiments show excellent detection rates (99.98%) and accuracy (96.35%), as well as a considerable reduction in false alarm rates (up to 5.59%).

The study [17] is concerned with the security of medical cyber-physical systems (MCPS), which include medical sensor devices with cyber components. Existing MCPS attack detection systems are regarded inefficient and time-consuming. To solve this, the research provides a unique technique for attack detection dubbed Fuzzy C-Means algorithm with Artificial Bee Colony Optimization (FCM-ABC). The innovative aspectis the use of the fuzzy c-means degree approach to assess whether data points belong to legitimate users or attackers, as well as the usage of ABC for self-organizing clusters with collective intelligence. In the MCPS model, the suggested FCM- ABC technique monitors health information utilizing sensor networks. The accuracy rates of several algorithms are com- pared: SVM obtains 76.32%, FCM achieves 81.34%, LSTM achieves 86.22%, and the suggested FCM-ABC achieves the greatest accuracy rate of 93.34%. The results show that the FCM-ABC technique is successful in identifying assaults in MCPS.

The study [18] goes through the security hazards of CCand how it might jeopardize the scalability and advantages of cloud-based applications. It focuses on the cyber issues that CC services face, with a focus on HTTP-based assaults likeas distributed denial of service (DDoS) attacks and zombie attacks. The use of ML algorithms to CAD in CC has yielded encouraging results. However, selecting characteristics andnormalizing cyber-attack data continue to be problems. To address this, the research presents an ensemble classifier-basedtraffic content optimization strategy. SVMs, RF, and DT are combined in the proposed ensemble classifier. The algorithm iswritten in MATLAB and tested with the CIDDS 001

datasets. The suggested technique outperforms current algorithms in terms of CAD ratio, with a performance improvement of 2-3%when compared to other ML approaches.

In a study focusing on cyber threats against cars [19], re- searchers propose using computational offloading to overcome the vehicle's limited processing capacity. They demonstrate the feasibility and benefits of outsourcing intrusion detection to external processing resources using a robotic land vehicleas a case study. Deep learning models are employed to analyze real-time data from both cyber and physical processes,achieving high accuracy in detecting various types of attacks. The authors also develop a mathematical model to assess the advantages of compute offloading based on network reliability and processing requirements, showing that the decrease indetection latency depends on these factors.

Paper [20] highlights the significance of security and IDSs in countering cyber attacks. It emphasizes the use of data mining and ML methods to achieve accurate detection rates and minimize false alarms. The study focuses on enhancing the CAD system through cloud-based ML techniques, as traditional ML approaches may struggle with large datasets. By utilizing a cloud-based ML platform, specifically Microsoft's Azure ML, and employing a Multiclass Decision Forest al-algorithm, attacks can be effectively classified. The proposed model's performance is evaluated using the NSL KDD Cup99 dataset and shows promising results compared to competition benchmarks. The paper concludes by providing suggestionsfor future research in this domain.

Table I presents a comprehensive summary of various related works in the domain of IDSs. Specifically, it outlines the methods used in each study, the datasets utilized for the research, and the resulting accuracy rates obtained from the proposed models. This tabular representation provides a clear comparison of the different approaches and their respective results, aiding in identifying effective techniques and areas for potential future research.

TABLE I RELATED WORKS

Ref	Method Used	Dataset	Results
[13]	Cloud-based intrusion de- tection model using the RF algorithm and feature engineering	Bot-IoT and NSL-KDD	Accuracy rates of 98.3% and 99.99% respectively
[14]	Data-driven framework utilizing ontology and a knowledge base	UNSW-NB15	Accuracy of 88.82%
[15]	IoT federated learning ar- chitecture and smart con- tract analysis for cloud- based intrusion detection	Various FinTech cyber-attack datasets	Accuracy 95%
[16]	Ensemble learning and fog-cloud architecture- driven framework (DT, Naive Bayes, Random Forest, and XGBoost algorithms)	ToN-IoT	Accuracy 96.35%
[17]	Fuzzy C-Means algorithm with Artificial Bee Colony Optimization (FCM-ABC) for attack detection	Medical information collected from various sensor devices	Accuracy 93.34%
[18]	Traffic content optimization approach based on an ensemble classifier (SVMs, RF, and DT)	CIDDS 001 and CICIDS- 2017	CIDDS 001 : 97.36% & CICIDS- 2017 : 98.78%
[19]	Computational offloading for intrusion detection based on DL	Data captured in real-time from a small four-wheel robotic land vehicle	Accuracy 86.9%
[20]	Cloud-based ML technique to classify attacks (Multiclass Decision Forest ML algorithm)	NSL KDD Cup99	Accuracy 96%

3. METHODOLOGY

In this paper, we introduce an innovative methodology targeting Cyber Attack Detection (CAD) within the realm of cloud-based IDSs as depicted in Figure 1. Acknowledging the distinct complexities inherent to the cloud milieu, our methodology capitalizes on the robust capabilities of ML, with a specific focus on sophisticated classification algorithms, in an effort to augment detection precision and efficacy. We implement a meticulous data preprocessing strategy that en-compasses label encoding, data scaling and data partitioning, aiming to curate a resilient dataset conducive to effective model training. Our proposed methodology is crafted to strike a delicate balance between false positives and negatives, thereby establishing a more trustworthy and comprehensive detection mechanism. Moreover, we plan to gauge the performance of our approach using an assortment of evaluation metrics, including accuracy, precision, recall, and the F1-

score, in order to provide an exhaustive appraisal of its efficacy. Ultimately, our research aims to offer a significant contribution to addressing the prevailing challenges encountered in cloud- based intrusion detection.

3.1 Dataset

In our work, we utilized the NSL-KDD dataset, which is an improvement of the previously used KDD Cup 99 dataset for network intrusion detection. This dataset comprises 137,823 entries and spans across 43 columns, demonstrating a broad spectrum of features associated with network traffic, network behaviors, and potential intrusion markers.

Columns include various integer, float, and object data types, encompassing attributes such as 'duration' of the connection, 'protocol type', 'service', and 'flag', along with 'src bytes' and 'dst bytes', representing the number of data bytes from source to destination, and vice versa. It also details several other behavioral characteristics, such as the number of 'wrong fragments', 'urgent' packets, 'hot' indicators, and failed login attempts, to name a few.

The dataset further includes numerous columns illustrat- ing rates such as 'serror rate', 'rerror rate', 'same srv rate', and 'diff srv rate', each shedding light on different as- pects of the connection profile. Features of the host are also encompassed, seen in the columns for 'dst host count', 'dst host srv count', and the various rate-related attributes for the destination host.

The 'class' column serves as the target variable, providing classification of network interactions as normal or anomalous. Hence, the NSL-KDD dataset forms a comprehensive data re- source for exploring and building models for network intrusion detection. Table II shows a summary of NSL-KDD dataset.

TABLE II SUMMARY OF THE NSL-KDD DATASET

Summary of the NSL-KDD dataset	
Number of Instances	137,823
Total Number of Features	43
Numerical Features	39
Categorical Features	4

3.2 Data preprocessing

Data preprocessing constitutes a pivotal stage in the data mining workflow. This stage entails converting unprocessed data into a format that is interpretable and appropriate for ML models. Untreated data is often characterized by noise, incompleteness, inconsistency, or it may be in a format that poses challenges for analysis. Hence, preprocessing is indispensable to sanitize, normalize, and standardize the data, rendering it suitable for subsequent examination [21]. In our research, we execute a biphasic data preprocessing protocol. The initial phase encompasses transmuting our categorical data through a technique referred to as label encoding. Subsequent to this transformation, we perform data scaling and then we segregate our data into two discrete subsets: a training set and a testing set.

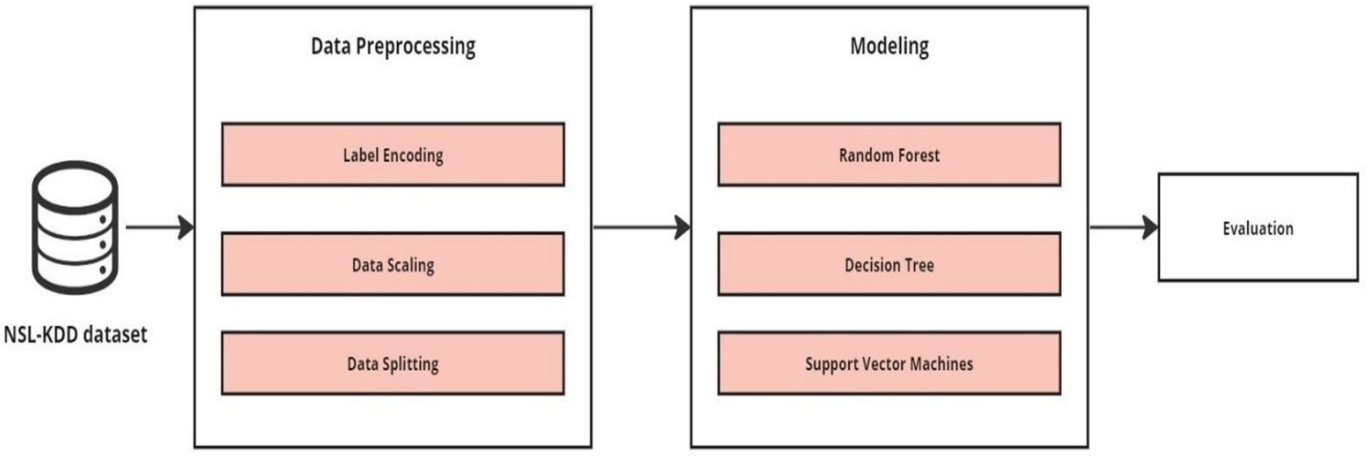

Fig. 1. Proposed Approach

Label Encoding: In our study, we employ label encoding, a pivotal process of converting categorical data into a machine-comprehensible format, to three specific columns in our dataset: Hash, Category, and Family. This procedure assigns unique numerical identifiers to each distinct category within these features. This transformation permits our ML algorithms to process these formerly non-numeric labels as numeric inputs, thereby enhancing the manageability and efficiency of our computational modeling. Nevertheless, it's important to note that while label encoding facilitates the algorithmic handling of categorical data, it may inadvertently imply a non-existent ordinal relationship among the categories. Hence, it's crucial to apply this technique judiciously, taking into consideration the characteristics of the dataset and the specifics of the research context.

Data scaling: Data scaling is an essential procedure in machine learning research which normalizes the input features' range, usually to a 0-1 interval, ensuring optimal model performance. The method used in this case is Min- Max scaling, which addresses issues of data disparity, where some features disproportionately influence the model due to their larger range.

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (1)$$

where X is the original data, Xmin and Xmax are its minimum and maximum values, and Xscaled is the transformed data set after scaling.

By scaling data, each feature can more equally influence model parameter learning, thereby enhancing prediction accuracy. Importantly, Min-Max scaling retains the original data distribution, beneficial when the data follows a known distribution.

Training Test splitting: The process of data splitting is an essential step within the ML workflow, wherein the accumulated dataset is partitioned into two separate subsets: the training set and the test set. This bifurcation facilitates the model's learning from a portion of the data (the training set) and subsequent validation of learned patterns on a portion of data it has not seen before (the test set). In our research, we adhere to the conventional practice of allocating 80% of the aggregate data to the training set while reserving the remaining 20% for the test set. This 80-20 ratio is a widely accepted practice, ensuring a balance between the necessity of having an ample training set for effective model learning and the need for a suitably sized test set to assess the model's capability to generalize to unfamiliar data. Employing this strategy of data splitting assists in reducing the likelihood of overfitting, a scenario where the model excels on the training data but fares poorly on novel data, thus promoting a more dependable and robust model.

3.3 Modeling

In the data modeling phase of our research, we implement a broad array of seven ML algorithms to enhance the detection capabilities of cyber-attacks in cloud-based IDSs. This selection encompasses a diverse set of models including DT, XGBoost, and Support Vector Machine (SVM) with both polynomial and Radial Basis Function (RBF) kernels. These algorithms were chosen based on their unique strengths and capabilities in learning from complex datasets, as well as their

distinct capacities to handle different types of classification problems. By deploying multiple algorithms, we are able to compare their performance and ascertain the most effective approach for detecting potential threats within our specific cloud-based IDS context. This diverse modeling approach is instrumental in addressing the complexity and evolving nature of cyber threats, thereby contributing to the development of a robust and reliable IDS.

3.4 Models Evaluation

Evaluation metrics play a vital role in assessing the effectiveness of ML models by quantifying their performance in specific tasks, helping us make informed decisions on their accuracy and potential improvements. These metrics are indispensable in the iterative process of model development and validation.

Confusion Matrices: are crucial in ML, allowing practitioners to quantitatively evaluate binary classification mod- els.As illustrated in Table III, they consist of four elements: True Positives (TP), False Positives (FP), True Negatives (TN), and False Negatives (FN), providing insights into the model's performance. These matrices serve as the foundation for various evaluation metrics, helping us understand the strengths and weaknesses of our binary classification models.

TABLE III GENERAL STRUCTURE OF CONFUSION MATRIX

		Actual Class	
		0	1
Predicted Class	0	TN	FN
	1	FP	TP

Accuracy (ACC): The accuracy (ACC) metric is widely used for evaluating binary classification tasks, providing an overall measure of how well a model correctly identifies instances, regardless of class. It is calculated as the sum of True Positives (TP) and True Negatives (TN) divided by the total count of all classes (TP, TN, False Positives - FP, and False Negatives - FN). Accuracy quantifies the proportion of correct predictions, offering a straightforward way to assess model performance, but it should be used with caution in cases of imbalanced classes. The formula for accuracy is as follows:

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}} \tag{2}$$

Precision (PREC): is a performance metric commonly used in binary classification tasks. It measures how well a model can correctly predict positive instances. It is calculated by dividing the number of TP by the sum of TPs and FPs. Precision indicates the relevance of a model and provides insight into the reliability of its positive predictions. The formula for precision is:

$$\text{Precision} = \frac{\text{TP}}{\text{TP + FP}} \tag{3}$$

Recall (REC): is a performance metric in binary classification that measures the model's ability to identify all actual positive instances. It is calculated as the ratio of TPs to the sum of TPs and FNs, representing the proportion of correctly predicted positive observations out of all real positive cases (TP + FN).

$$\text{Recall} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Negatives (FN)}} \tag{4}$$

F1-score (F1-s): is a commonly used metric in binary classification that balances PREC and REC by taking their harmonic mean. It provides a single measure of performance, favoring models that excel in both PREC and REC . The calculation for the F1-Score is as follows:

$$\text{F}_1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision+Recall}} \tag{5}$$

4. EXPERIMENTAL RESULTS

This section describes the experimental findings of each model used in our study to identify cyber-attacks in a cloud setting. This section seeks to give a thorough examination and review of each model's performance, revealing insight on its efficacy and appropriateness for this specific purpose. We want to obtain insights into the strengths and limitations of each model through a rigorous experimental setup and painstaking evaluation, allowing for a full knowledge of their capabilities and performance in tackling the issues of CAD in cloud systems.

4.1 Random Forest Results

In our study, we used the Random Forest (RF) algorithm to differentiate between anomalous (1) and normal (0) network interactions. The algorithm showcased an exceptional performance, achieving an accuracy (ACC) rate of approximately 99.41% in the classification task. This high ACC denotes that our model successfully identified the majority of instances, proving its reliability for network intrusion detection tasks.

Further insights into the model's performance can be drawn from the confusion matrix analysis (Figure ??). The model precisely predicted 13,916 normal interactions and 13,478 anomalies. Nevertheless, it incorrectly classified 83 instances as normal when they were actually anomalies and mislabeled 79 actual normal instances as anomalies. The comparatively low number of false negatives and positives relative to true positives and negatives underscores the effectiveness of our RF model in this application.

From a computational perspective, the RF model displayed an acceptable execution time of 13.92 seconds, indicating its capability to deliver high-accuracy predictions in a relatively short period. This adds to its applicability in real-time intrusion detection situations.

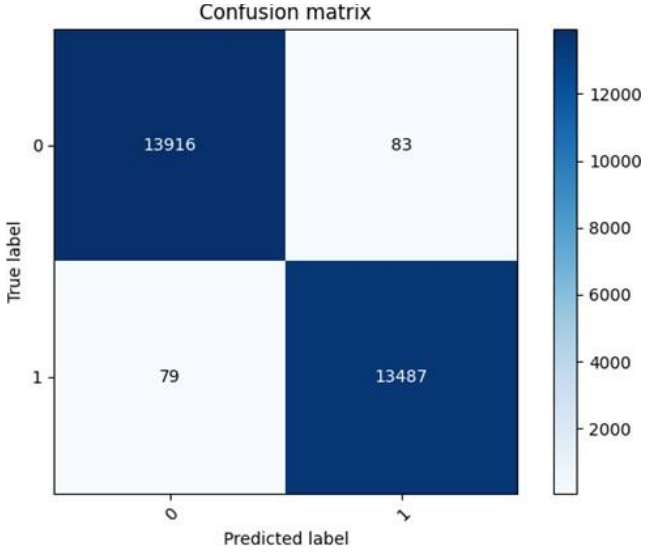

Fig. 2. Confusion Matrix of Random Forest

The outcomes from the application of the RF classification algorithm in our research delineate an exceptional performance on several fundamental evaluation metrics (Table IV). For both the anomalous (1) and normal (0) classes, the model delivered an impressive PREC and RE score of roughly 0.99. This indicates the model's proficiency in accurately pinpointing the majority of true instances for each class while maintaining a low count of false positives, hence the high PREC. The F1- s, a measure that encapsulates both PREC and RE, clocked approximately 0.99 for both classes. This denotes our model's success in striking an effective balance between these two critical metrics. The overall ACC of the model, considering all classes, also rounded to about 0.99. This confirms the model's superior performance in distinguishing between regular and abnormal network interactions. Additionally, both the macro and weighted averages, which are calculations of the evaluation metrics for each class without and with consideration of their proportions, respectively, scored roughly 0.99. These figures further emphasize the consistency and reliability of our RF model's performance across different metrics and classes.

TABLE IV CLASSIFICATION REPORT OF THE RF MODEL

	Precision	Recall	F1-Score	Accuracy
Class 0	0.99	0.99	0.99	0.99
Class 1	0.99	0.99	0.99	0.99
Macro Avg	0.99	0.99	0.99	0.99
Weighted Avg	0.99	0.99	0.99	0.99

In our exploration of the RF algorithm's capabilities for network intrusion detection, we have observed compelling results. Specifically, our model yielded a Sensitivity score of approximately 0.994, which denotes a high degree of effective- ness in accurately identifying positive instances, or anomalies. The paramount importance of Sensitivity in network security contexts cannot be overstated, as the implications of not detecting malicious activities can lead to substantial security compromises. In parallel, our model also displayed a Specificity score of roughly 0.994. This value points to the model's strong aptitude for correctly identifying negative instances, or normal network interactions in our context.

4.2 XGBoost Results

In our study, we additionally investigated the utility of the XGBoost algorithm for discerning between normal (0) and anomalous (1) network interactions. The performance exhibited by XGBoost was truly extraordinary, achieving an overall ACC of approximately 99.63%. This substantial degree of ACC suggests that the XGBoost model correctly recognized an overwhelming majority of instances, underscoring its competency for network intrusion detection tasks. A deeper analysis via the confusion matrix depicted in Figure 3 offers further perspectives on the XGBoost model's performance. The model accurately identified 13,954 normal interactions and 13,509 anomalies, equating to true negatives and true positives, respectively. However, it misclassified 45 actual anomalies as normal and erroneously flagged 57 actual normal instances as anomalies. Despite these slight inaccuracies, the quantities of false negatives and positives are minuscule compared to the number of instances correctly identified, underpinning the potency of our XGBoost model. Regarding computational efficiency, the XGBoost model necessitated an execution time of 26.02 seconds. Although slightly longer compared to some other models, the remarkable accuracy yielded by XGBoost substantiates this trade-off.

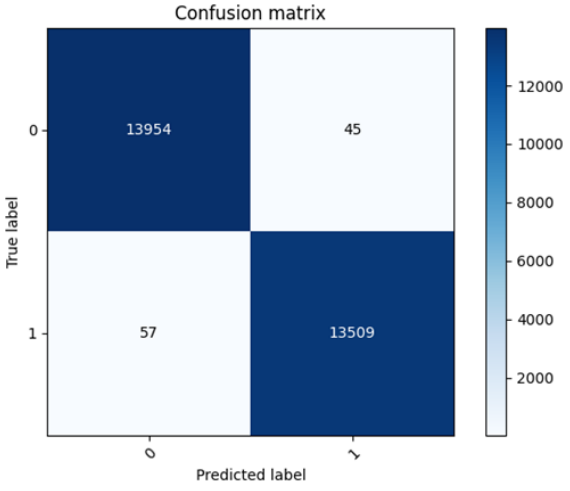

Fig. 3. Confusion Matrix of XGBoost

The performance of the XGBoost model, as outlined inTable 3, achieved an outstanding score of 1.00 in both PREC and RE across the two classes. This indicates the model's impeccable ability to identify true instances for each classand its excellent control over false positive predictions, hence exhibiting the maximum level of PREC. Further, the F1-s, a metric that harmonizes PREC and RE, also attained the top score of 1.00 for both classes. This achievement underscores our model's capacity to masterfully balance these critical metrics. In addition, the model's overall ACC registered 1.00, indicating its superior ability in differentiating normal from anomalous network interactions. Furthermore, both the macro average and weighted average, which represent the mean scores of the evaluation metrics for each class without and with consideration to their proportions, respectively, mirrored the excellent score of 1.00. These results collectively highlight the exceptional performance of our XGBoost model in this classification task.

TABLE V CLASSIFICATION REPORT OF THE XGBOOST

	Precision	Recall	F1-Score	Accuracy
Class 0	1.00	1.00	1.00	1.00
Class 1	1.00	1.00	1.00	1.00
Macro Avg	1.00	1.00	1.00	1.00
Weighted Avg	1.00	1.00	1.00	1.00

Our model exhibited an impressive Sensitivity score of around 0.996, signifying its exceptional proficiency in cor- rectly identifying positive instances - anomalies, in our study context. This is crucial in network security contexts, as any failure to detect malicious activities can lead to serious security breaches. Equally commendable is the model's Specificity score of roughly 0.997, indicative of its outstanding ability to correctly classify negative instances or, in our case, normal interactions. This aspect is of significant importance as it minimizes the chances of false alerts, thus preventing unnecessary resource utilization or creating undue alarm. The high scores in both Sensitivity and Specificity underline the XGBoost model's robust proficiency in accurately detecting both the existence and non-existence of network anomalies.

4.3 Decision Tree Results

Our investigation also encompassed the deployment of the DT algorithm for the classification of standard (0) and anomalous (1) network interactions. The DT model displayed laudable effectiveness, realizing an ACC of approximately 99.09%. This high ACC score underscores the model's adept- ness in correctly assigning the vast majority of instances to their respective classes. A deeper dive into the model's performance is facilitated by the confusion matrix, as depicted in Figure 4. The model succeeded in predicting 13,861 normal interactions and 13,454 anomalies accurately. Nevertheless, the model made a few errors, classifying 138 genuine anomalies as normal and misidentifying 112 normal instances as anomalies. Despite these errors, they represent only a small fraction compared to the correctly classified instances, hence validating the overall efficiency of the DT model. An important highlight is the model's computational efficiency with an execution time of just 1.34 seconds.

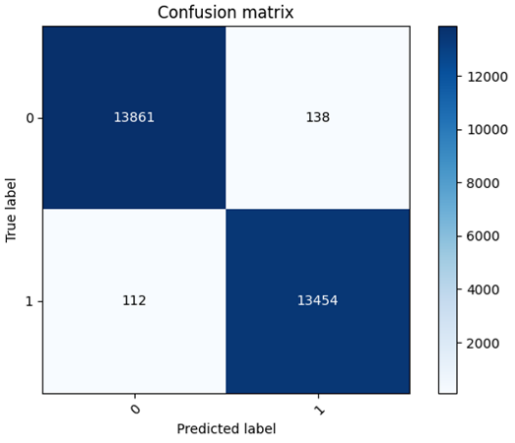

Fig. 4. Confusion Matrix of DT

The classification report of our DT model, detailed in Table VI, underscores its impressive predictive capabilities. The DT model exhibited high performance, achieving a PREC, RE, and F1-s of around 0.99 for both normal (0) and anomalous (1) classes. The PREC demonstrates the model's proficiency in making precise positive predictions, as it represents the pro- portion of true positive cases among those predicted positive by the model. Similarly, the RE underscores the model's ability to effectively identify all relevant instances in the dataset, highlighting its strength in accurately detecting anomalies. Furthermore, the F1-s, the harmonic mean of PREC and RE, also achieved an approximate value of 0.99 for both classes, reflecting the model's capacity to maintain a balanced performance between PREC and RE. This indicates that the DT model does not unduly prioritize one measure over the other, but rather preserves a noteworthy balance. Addition- ally, the model's overall ACC, along with the macro and weighted averages, also attained approximately 0.99. These figures further attest to the consistency and robustness of the DT model's performance across both classes, reinforcing the model's reliability in our dataset.

TABLE VI CLASSIFICATION REPORT OF THE DT

	Precision	Recall	F1-Score	Accuracy
Class 0	0.99	0.99	0.99	0.99
Class 1	0.99	0.99	0.99	
Macro Avg	0.99	0.99	0.99	
Weighted Avg	0.99	0.99	0.99	

The efficacy of the DT model is illuminated through its sensitivity and specificity scores, both registering around 0.99. With a sensitivity score of 0.9917, the DT model manifests an excellent ability in accurately detecting anomalous network behavior, which is crucial in preventing potential network intrusions. Concurrently, a specificity score of 0.9901 signifies the model's superior skill in recognizing normal network activities, thus minimizing the likelihood of falsely marking normal traffic as anomalous.

4.4 Polynomial Support Vector Machines Results

As part of our investigation into different algorithms for anomaly detection, we implemented a Poly SVM and scrutinized its performance indicators. The Poly SVM model produced an ACC of 0.9715, suggesting a notable precision in its predictions. To be more specific, the confusion matrix, as shown in Figure 5, reveals that the model correctly identified 13,710 instances as normal (class 0), albeit with a slight misclassification of 289 instances. In relation to anomalies (class 1), the model proficiently detected 13,069 instances as anomalous, while inaccurately classifying 497 instances. Notably, the Poly SVM's execution time stood at about 86.93 seconds, which underlines the model's computational efficiency, given its commendable ACC value.

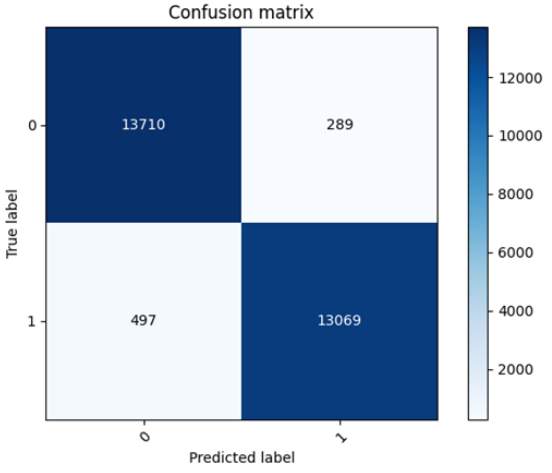

Fig. 5. Confusion Matrix of Poly-SVM

The classification report of the Polynomial SVM model offers a thorough assessment of its performance characteristics, as documented in Table VII. This report reveals that the model maintains high PREC and RE values for both classes. To be more precise, the model displays a PREC of 0.97 and a RE of 0.98 for the normal class (0). Concurrently, it garners a notable PREC of 0.98 and a RE of 0.96 for the anomaly class (1). Such a high PREC reveals the model's prowess in keeping the false positive rate low, while the elevated RE indicates a minimized false negative rate, which is critically important in scenarios of anomaly detection. The model's F1-s stands at 0.97 for both classes, suggesting a high and balanced performance. Moreover, the model's overall accuracy is recorded at 0.97, demonstrating its potent ability to distinguish between normal and anomalous instances.

TABLE VII CLASSIFICATION REPORT OF THE POLY-SVM

	Precision	Recall	F1-Score	Accuracy
Class 0	0.97	0.98	0.97	0.97
Class 1	0.98	0.96	0.97	
Macro Avg	0.97	0.97	0.97	
Weighted Avg	0.97	0.97	0.97	

Our application of the Poly SVM model to the Binary Classification (BC) problem demonstrates its robust proficiency in the accurate detection of both class labels. With a sensitivity metric of 0.963, the model correctly predicted around 96.3% of the positive instances (anomalies), indicating a relatively low FN rate and thus ensuring the model's dependability in detecting anomalies. Moreover, the model exhibited a specificity score of 0.979, demonstrating that it was able to accurately identify approximately 97.9% of actual negative instances, further underlining its efficacy.

4.5 RBF Support Vector Machines Results

In our anomaly detection task, we assessed the performance of the SVM-RBF in predicting normal (0) and anomalous(1) network traffic. The SVM-RBF displayed a high ACC score of approximately 0.971, suggesting that it accurately predicted the class labels for nearly 97.1% of all instances. Although the execution time was somewhat lengthier at 135.23 seconds, the SVM-RBF provided a satisfactory ACC level. An analysis of the SVM-RBF's confusion matrix, illustrated in Figure 6, provides more detailed performance metrics. The model managed to correctly classify 13,669 instances of the normal class (0), while misclassifying 330 instances. Similarly, it accurately identified 13,095 instances of the anomalous class (1), albeit misclassifying 471 instances. This suggests that while the model exhibited a strong performance in identifying normal network interactions, it faced slightly more challenges with anomalous traffic. Nevertheless, the SVM-RBF model's overall performance is admirable, making it a worthy consideration for tasks involving anomaly detection.

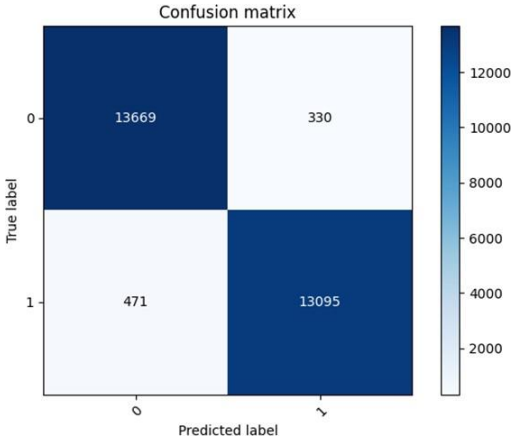

Fig. 6. Confusion Matrix of RBF-SVM

The SVM-RBF model demonstrated its aptitude in discerning between regular and anomalous network traffic, as highlighted by the classification report outlined in Table VIII. The model showcased high values of PREC and RE for both normal and anomalous classes, both around the 0.97 mark. This concurrent high level of PREC and RE suggests an effective equilibrium between correctly identifying true positives and reducing false negatives. Moreover, this balance was approximately equivalent for both classes, emphasizing the model's ability to manage the binary classification task without overly favoring one class. The SVM-RBF had an ACC of around 0.97, indicating that about 97% of total predictions were accurate. These performance metrics reflect the SVM- RBF's strong performance in this application.

TABLE VIII CLASSIFICATION REPORT OF THE DT

	Precision	**Recall**	**F1-Score**	**Accuracy**
Class 0	0.97	0.98	0.97	0.97
Class 1	0.98	0.96	0.97	
Macro Avg	0.97	0.97	0.97	
Weighted Avg	0.97	0.97	0.97	

5. DISCUSSION

When evaluating the best model for Cloud-Based IDS to de- tect cyber-attacks, we need to consider both the model's ACC, essential for the reliable detection of threats, and training time, affecting the system's overall performance and speed. Our comparative analysis, delineated in Table IX, illuminates that the XGBoost model outperforms other models with the highest ACC of roughly 0.9962. This remarkable ACC underscores XGBoost's capacity to detect subtle cyber-attacks, thereby minimizing false positives and missed threats. However, it is essential to note that this level of performance comes with an extended training time of 26.02 seconds. On the other hand, the DT model, with a slightly lower ACC of 0.9909,

stands out due to its short training time of 1.34 seconds. This rapid training time could be beneficial in real-time intrusion detection scenarios, where swift responses are vital, even at the cost of a minor decrease in ACC. The RF model presents a suitable compromise, securing an ACC of 0.9941 and a reasonable training time of 13.92 seconds, thus ensuring a blend of high ACC and satisfactory training speed, rendering it a flexible choice for various IDS settings. In comparison, both the SVM models—Polynomial and RBF—demonstrate the lengthiest training durations (86.93 and 135.23 seconds, respectively) among all models studied, even though their accuracies hover around 0.97.

TABLE IX CLASSIFICATION REPORT OF THE DT

	ACC	Training Time (sec)
XGBoost	0.9962	26.02
Decision Tree	0.9909	1.34
Random Forest	0.9941	13.92
SVM (Poly)	0.9714	86.93

6. CONCLUSION

This thesis has extensively explored CAD within Cloud- Based IDS, underscoring the critical need for such robust systems due to the increasing reliance on cloud services. A variety of ML models, including DT, XGBoost, and SVM, were evaluated based on performance metrics like accuracy, precision, recall, and F1-score. The XGBoost model out- performed the rest, showcasing high accuracy and balanced performance, thus highlighting the promise of ensemble tech- niques in bolstering cloud-based IDS. Future research could focus on exploring other ML and DL models, ensemble methods, and the integration of emerging technologies like edge computing and federated learning. Additionally, the in- vestigation of multi-class classification or anomaly detection techniques for identifying specific cyber-attacks may enhance targeted responses and improve cloud system security.

Conflicts Of Interest

The authors declare no conflicts of interest.

Funding

None.

Acknowledgment

The authors would like to thank American University of Culture and Education for their moral support.

REFERENCES

[1] M. Arunkumar and K. Ashok Kumar, "Malicious attack detection approach in cloud computing using machine learning techniques," Soft Computing, vol. 26, no. 23, pp. 13097–13107, 2022.

[2] "What is Cloud Security?" Microsoft Security, June 2023. [Online; accessed 1. Jun. 2023].

[3] Y. Li and Q. Liu, "A comprehensive review study of cyber-attacks and cyber security; emerging trends and recent developments," Energy Reports, vol. 7, pp. 8176–8186, 2021.

[4] P. Chowdhury, S. Paul, R. Rudra, and R. Ghosh, "Cyber-attack in ICT cloud computing system," in Information and Communication Technology for Competitive Strategies (ICTCS 2021) ICT: Applications and Social Interfaces, Springer, 2022, pp. 115–121.

[5] N. Amara, H. Zhiqui, and A. Ali, "Cloud computing security threats and attacks with their mitigation techniques," in 2017 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), IEEE, 2017, pp. 244–251.

[6] K. Xing, A. Li, R. Jiang, and Y. Jia, "Detection and defense methods of cyber attacks," in MDATA: A New Knowledge Representation Model: Theory, Methods and Applications, 2021, pp. 185–198.

[7] S. Reddy and G. K. Shyam, "A machine learning based attack detection and mitigation using a secure SaaS framework," Journal of King Saud University-Computer and Information Sciences, vol. 34, no. 7, pp. 4047–4061, 2022.

[8] I. H. Sarker, A. S. M. Kayes, S. Badsha, H. Alqahtani, P. Watters, and A. Ng, "Cybersecurity data science: an overview from machine learning perspective," Journal of Big Data, vol. 7, pp. 1–29, 2020.

[9] B. De Ville, "Decision trees," Wiley Interdisciplinary Reviews: Computational Statistics, vol. 5, no. 6, pp. 448–455, 2013.

[10] T. Chen, T. He, M. Benesty, V. Khotilovich, Y. Tang, H. Cho, K. Chen, R. Mitchell, I. Cano, T. Zhou, et al., "Xgboost: extreme gradient boosting," R package version 0.4-2, vol. 1, no. 4, pp. 1–4, 2015.

[11] D. Meyer and F. T. Wien, "Support vector machines," The Interface to libsvm in package e1071, vol. 28, p. 20, 2015.

[12] F. Pasqualetti, F. Do¨rfler, and F. Bullo, "Attack detection and identification in cyber-physical systems–part I: models and fundamental limitations," arXiv preprint arXiv:1202.6144, 2012.

[13] H. Attou, A. Guezzaz, S. Benkirane, M. Azrour, and Y. Farhaoui, "Cloud-based intrusion detection approach using machine learning techniques," Big Data Mining and Analytics, vol. 6, no. 3, pp. 311–320, 2023.

[14] S. Badde, V. Kumar, K. Chatterjee, and D. Sinha, "Cyber attack detection framework for cloud computing," in Intelligent Data Engineering and Analytics: Frontiers in Intelligent Computing: Theory and Applications (FICTA 2020), Volume 2, Springer, 2021, pp. 243–254.

[15] V. N. Kollu, V. Janarthanan, M. Karupusamy, and M. Ramachandran, "Cloud-based smart contract analysis in fintech using IoT-integrated federated learning in intrusion detection," Data, vol. 8, no. 5, p. 83, 2023.

[16] P. Kumar, G. P. Gupta, and R. Tripathi, "An ensemble learning and fog-cloud architecture-driven cyber-attack detection framework for IOMT networks," Computer Communications, vol. 166, pp. 110–124, 2021.

[17] F. Alrowais, H. G. Mohamed, F. N. Al-Wesabi, M. Al Duhayyim, A. M. Hilal, and A. Motwakel, "Cyber attack detection in healthcare data using cyber-physical system with optimized algorithm," Computers and Electrical Engineering, vol. 108, p. 108636, 2023.

[18] J. K. Ghai and A. Jain, "Detection of cyber-attacks in cloud computing using optimization of traffic content and ensemble classifier," Stochastic Modeling.

[19] G. Loukas, T. Vuong, R. Heartfield, G. Sakellari, Y. Yoon, and D. Gan, "Cloud-based cyber-physical intrusion detection for vehicles using deep learning," IEEE Access, vol. 6, pp. 3491–3508, 2017.

[20] R. Chourasiya, V. Patel, and A. Shrivastava, "Classification of cyber attack using machine learning technique at Microsoft Azure cloud," Int. Res. J. Eng. Appl. Sci, 2018.

[21] C. Fan, M. Chen, X. Wang, J. Wang, and B. Huang, "A review on data preprocessing techniques toward efficient and reliable knowledge discovery from building operational data," Frontiers in Energy Research, vol. 9, p. 652801, 2021.