applied sciences

MDPI

Article

Towards an Intelligent Intrusion Detection System to Detect Malicious Activities in Cloud Computing

Hanaa Attou [1], Mouaad Mohy-eddine [1], Azidine Guezzaz [1], Said Benkirane [1], Mourade Azrour [2,*], Abdulatif Alabdultif [3,*] and Naif Almusallam [4]

[1] Technology Higher School Essaouira, Cadi Ayyad University, Essaouira 44000, Morocco; attouhanaa39@gmail.com (H.A.); mouaadmohyeddine@gmail.com (M.M.-e.); a.guzzaz@gmail.com (A.G.); sabenkirane@gmail.com (S.B.)
[2] Informatique Décisionnelle et Modélisation des Systèmes (IDMS) Team, Faculty of Sciences and Techniques, Moulay Ismail University of Meknès, Errachidia 52000, Morocco
[3] Department of Computer Science, College of Computer, Qassim University, Buraydah 52571, Saudi Arabia
[4] Department of Management Information Systems (MIS), College of Business Administration, King Faisal University (KFU), Al-Ahsa 31982, Saudi Arabia; nalmuslem@kfu.edu.sa
* Correspondence: mo.azrour@umi.ac.ma (M.A.); ab.alabdulatif@qu.edu.sa (A.A.)

Abstract: Several sectors have embraced Cloud Computing (CC) due to its inherent characteristics, such as scalability and flexibility. However, despite these advantages, security concerns remain a significant challenge for cloud providers. CC introduces new vulnerabilities, including unauthorized access, data breaches, and insider threats. The shared infrastructure of cloud systems makes them attractive targets for attackers. The integration of robust security mechanisms becomes crucial to address these security challenges. One such mechanism is an Intrusion Detection System (IDS), which is fundamental in safeguarding networks and cloud environments. An IDS monitors network traffic and system activities. In recent years, researchers have explored the use of Machine Learning (ML) and Deep Learning (DL) approaches to enhance the performance of IDS. ML and DL algorithms have demonstrated their ability to analyze large volumes of data and make accurate predictions. By leveraging these techniques, IDSs can adapt to evolving threats, detect previous attacks, and reduce false positives. This article proposes a novel IDS model based on DL algorithms like the Radial Basis Function Neural Network (RBFNN) and Random Forest (RF). The RF classifier is used for feature selection, and the RBFNN algorithm is used to detect intrusion in CC environments. Moreover, the datasets Bot-IoT and NSL-KDD have been utilized to validate our suggested approach. To evaluate the impact of our approach on an imbalanced dataset, we relied on Matthew's Correlation Coefficient (MCC) as a normalized measure. Our method achieves accuracy (ACC) higher than 92% using the minimum features, and we managed to increase the MCC from 28% to 93%. The contributions of this study are twofold. Firstly, it presents a novel IDS model that leverages DL algorithms, demonstrating an improved ACC higher than 92% using minimal features and a substantial increase in MCC from 28% to 93%. Secondly, it addresses the security challenges specific to CC environments, offering a promising solution to enhance security in cloud systems. By integrating the proposed IDS model into cloud environments, cloud providers can benefit from enhanced security measures, effectively mitigating unauthorized access and potential data breaches. The utilization of DL algorithms, RBFNN, and RF has shown remarkable potential in detecting intrusions and strengthening the overall security posture of CC.

Keywords: cloud security; anomaly detection; features engineering; radial basis function neural network; random forest

Citation: Attou, H.; Mohy-eddine, M.; Guezzaz, A.; Benkirane, S.; Azrour, M.; Alabdultif, A.; Almusallam, N. Towards an Intelligent Intrusion Detection System to Detect Malicious Activities in Cloud Computing. *Appl. Sci.* **2023**, *13*, 9588. https://doi.org/10.3390/app13179588

Academic Editor: Ying Weng

Received: 16 July 2023
Revised: 12 August 2023
Accepted: 17 August 2023
Published: 24 August 2023

1. Introduction

CC is a network access model that provides resources such as networks, data centers, hardware, software, and utilities on demand [1,2]. Hence, CC is a promising technology

that offers several facilities like obtaining data remotely, storage, and accessibility [2]. It significantly reduces costs due to its different characteristics, such as availability, scalability, and self-services [2]. According to the National Institute of Standard Technologies, three cloud service models comprise this model. Therefore, the cloud services are Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS), which are employed in different deployments of CC, including hybrid, public, and private clouds [3–6]. The most common type of cloud is public because individual customers and organizations use it.

Nevertheless, this type has some limitations, like maintenance worries and lower security [7]. For this, companies choose the private cloud because this type is located on premises and provides higher protection [7,8]. In general, CC faces several security issues that prevent the cloud infrastructure from being adopted quickly [9,10], such as regulation, the destruction of data stored on the cloud, and privacy concerns [11,12]. These issues include the sensitivity of users and organizations [10]. Numerous strategies have been developed and applied to secure applications, data, and cloud environments against attacks like firewalls and anti-virus, but they still need to be improved [13]. Then, to enhance cloud security, intrusion detection is a set of advanced technologies that recognize unpleasant activities [14–17]. There are three types of IDSs: misuse detection, anomaly detection, and hybrid detection [17,18]. An anomaly IDS is installed to detect attacks based on previously recorded normal behavior [19,20]. This form of IDS is commonly utilized because it can detect new intrusions by comparing current real-time traffic with recorded regular real-time traffic. However, it registers good false-positive alarms, implying that many regular packets are mistaken for storm packets.

On the contrary, a misuse IDS is used to identify intrusion using a signatures database. It does not cause false alarms but can be passed by a new attack with a unique signature [19,20]. Moreover, IDS is affected by several constraints that reduce the effectiveness of intrusion detection, such as vast volumes of data, instantaneous detection, the integrity of data, and more [18]. Recently, ML and DL algorithms have been employed to handle security concerns and improve data management [21–26]. In this respect, we suggest a reliable model combining ML and DL techniques to enhance IDS for cloud security and help distinguish attacks and expected activities on the cloud. We aim in this article to develop an intrusion detection approach using RF and the RBFNN classifier. The RF selector is used for the feature selection process to reduce the number of used variables, perform the suggested system, and overtake the impact of the imbalanced datasets. As a result, using a minimum number of variables saves execution time, storage space, and computational costs. Then, we train and evaluate the RBFNN classifier on the NSL-KDD and the Bot-IoT dataset as an imbalanced dataset to study the influence of this imbalance on the performance of our model.

In summary, the motivation for this novel work stems from the limitations of existing studies, including their inability to detect complex intrusions, inefficient feature selection, and limited interpretability. By addressing Top of Form, the increasing popularity of CC in various sectors, our research aims to explore the security challenges cloud providers face due to the shared infrastructure of cloud systems. While CC offers scalability and flexibility, it poses significant security concerns, including unauthorized access, data breaches, and insider threats. These vulnerabilities make cloud systems attractive targets for attackers. Hence, we recognize the urgency to integrate robust security mechanisms like IDSs to safeguard networks and cloud environments effectively.

In this study, we propose a novel IDS model based on DL algorithms, namely the RBFNN and RF. Our primary contributions include the following:

- A novel IDS model leveraging DL algorithms: We demonstrate the effectiveness of utilizing the RBFNN and RF to enhance IDS performance in CC environments.
- Improved ACC and detection rates: By selecting the top-k most essential features using RF and by training the RBFNN classifier accordingly, we achieve an ACC higher than 92% using minimal features, which is a substantial increase from an initial MCC of 28% to 93%.

- Addressing security challenges in CC: Our approach targets explicitly the security challenges posed by CC environments, offering a promising solution to enhance overall security.
- Utilization of real-world datasets: To validate our proposed approach, we employ the Bot-IoT and NSL-KDD datasets, reflecting the relevance and practicality of our findings.

This paper makes significant contributions to securing data in cloud environments. Firstly, it introduces a novel approach that combines the RBFNN classifier and RF for feature selection. By leveraging the strengths of both techniques, the proposed method enhances the ACC, efficiency, and interpretability of intrusion detection in the cloud. Secondly, the RBFNN classifier effectively captures complex and non-linear relationships within the data, enabling the detection of intricate and evolving intrusions. Thirdly, incorporating RF for feature selection improves computational efficiency, reduces overfitting risks, and enhances overall system performance. Lastly, the proposed approach provides enhanced interpretability by offering insights into underlying patterns and decisions, aiding in understanding classification decisions and identifying the root causes of detected intrusions. Overall, this paper provides a valuable contribution by addressing limitations in existing studies and presenting a comprehensive and practical approach to securing data in cloud environments. The rest of this work is structured as follows: Section 2 defines the CC architecture and several related studies on IDSs that use ML and DL. Section 3 outlines the phases of our approach. The experimental setting is depicted in Section 4, and the obtained outcomes are detailed in Section 5. A conclusion is included at the end of the paper.

2. Background and Related Works

This section outlines CC infrastructure and intrusion detection techniques and references current IDSs using ML and DL techniques to enhance intrusion detection.

CC has become accessible as a set of public and private cloud services, giving users an Internet-wide uniform platform [9]. The CC ecosystem comprises three main service models: IaaS, PaaS, and SaaS. These models form the fundamental components of CC and are deployed in various configurations, including community, private, hybrid, and public clouds [9]. Each service model offers unique functionalities catering to different user needs [9].

IaaS, the foundational layer, provides virtualization, servers, storage, and network resources, offering users a flexible and scalable infrastructure to build and manage their applications [5]. PaaS builds upon IaaS by offering technical layers and management software instances, enabling developers to focus on application development without worrying about the underlying infrastructure [5]. On the other hand, SaaS offers fully functional software applications accessed via the cloud, allowing users to run applications without needing local installations [5].

Despite their advantages, each service model also faces specific challenges. For IaaS, virtualization, although critical for infrastructure provisioning, has some limitations, and the usefulness of IaaS services may diminish over time [5,6]. PaaS faces challenges with interoperability, host sensitivity, confidentiality, authorization, reliability, and extensibility. However, SaaS grapples with security concerns around authorization, authentication, data protection, reliability, and network monitoring [5]. Cloud companies must address these security challenges [5,6].

As the threat landscape evolves, actors seeking to exploit weaknesses in cloud environments constantly change their tools and techniques [27]. Traditional IDSs often need help to detect variations in network traffic characteristics effectively. As a response, researchers emphasize the importance of using ML and DL techniques to enhance IDS capabilities [28]. ML and DL have gained prominence in various fields, including finance, government, scientific research, and security [29,30]. ML's data clustering and classification efficiency are critical in cybersecurity applications [31,32].

IDSs, designed to detect malicious files and activities, can be classified into two categories: misuse-based and anomaly-based [1,19,20,33–35]. An anomaly-based IDS analyzes

real-time traffic against previously recorded normal behavior to detect new intrusions. While this approach can identify novel attacks, it may also generate false-positive alarms, incorrectly flagging regular packets as malicious [19,20]. Conversely, a misuse-based IDS relies on a signature database to detect known attacks, reducing the false alarm rate, but it may miss new threats with unrecognized signatures [19,20].

ML- and DL-enhanced IDS development has become a key focus for various sectors as they strive to address security challenges in cloud environments and safeguard against emerging threats.

ML, DL, and ensemble learning methods have recently enhanced IDSs to identify attacks [36]. As a result, as shown in Table 1, several authors have examined their efforts to improve IDSs for the cloud environment. In 2023, Mohy-eddine et al. [37] suggested an IDS using K-NN to enhance the detection rate and ACC. They applied principal component analysis, univariate statistical tests, and genetic algorithms for feature selection. They evaluated their proposed model on the Bot-IoT dataset with 99.99% ACC. In 2016, the authors of [13] proposed a collaborative and hybrid detection approach in CC. In [14], M. Douiba et al. proposed an optimized IDS using Gradient Boosting and a Decision Tree (DT) for Internet of Things (IoT) security. The authors of [23] presented a Novel Anomaly Network IDS to secure the IoT. In [38], long short-term memory (LSTM) and recurrent neural networks (RNN) were identified as the most effective options for multichannel IDSs after the authors assessed the performance of the suggested approach. The model's performance was estimated at 99.23%, with an ACC of 98.94%. A. Alshammari et al. in [36] used an Artificial Neural Network (ANN), K-nearest neighbors (KNN), a DT, a Support Vector Machine (SVM), Naïve Bayes (NB), and RF to feed an IDS and identify an intrusion. In [39], the authors applied ML algorithms for data integrity, and they deduced that RF outperforms other techniques such as NB, SVM, and KNN. In 2020, the authors of [40] developed a model based on an SVM to identify attacks. The model efficiency is specified to be 96.23%. In 2021, the authors of [41] proposed a system based on ML approaches, including KNN, RF, and NB, to identify intrusion in CC. This model's ACC is 99.76%. The authors deduced that RF outperforms KNN and NB. The authors [42] describe a reliable network-based IDS that utilizes these classifiers: boosted tree, bagged tree, subspace discriminant, and RUS Booted. They used CICIDS 2017 and Cloud Sim datasets for tests and simulation. The system achieves a 97.24% ACC. In [43], the authors proposed an IDS for detecting DDoS attacks in CC, employing KNN, RF, and NB. The system achieves 99.76% ACC. In 2022, Mohy-eddine M. et al. [44] suggested an IDS model using ensemble learning to secure IIoT edge computing. The authors of [45] proposed an IDS applying a GA-Based Feature Selection technique, and RF. Verma et al. [43] recently compared various ML approaches to identify a classification algorithm to secure the IoT. They displayed an ensemble learning-based IDS with an ACC of 99.53%. A OneM2M IDS utilizing ML was recommended by Chaabouni et al. [46] to control the IoT. The model achieves 92.32% in terms of ACC. In [47], the authors proposed an IDS using DL algorithms for binary classification. H. Attou et al. [48] suggested an IDS to secure the cloud environment from intrusion. They use a combination of graphic visualization and RF classifier to enhance the detection of anomalies. They achieve 100% and 98.3% in terms of ACC using Bot-IoT and NSL-KDD datasets. In 2021, the authors of [49] used an LSTM classifier in a suggested SDN-based IDS to identify attacks on the IoT. They achieved a 99.05% ACC on the used datasets, as mentioned in Table 1.

Table 1. A comparison study of several IDSs.

Contribution	Year	Methods	Data	ACC (%)
[13]	2016	ANN	-	-
[14]	2022	Gradient Boosting DT	NSL-KDD Bot-IoT IoT 23	100 100 100
[36]	2021	ANN KNN DT SVM NB RF	ISOT-CID	92 100 100 81 60 100
[38]	2018	LSTM	NSL-KDD	98.94
[39]	2022	RF, NB, SVM, KNN	-	92
[40]	2020	SVM	-	96.23
[41]	2021	RF, KNN, NB	-	99.76
[42]	2021	Ensemble Learning	CICIDS 2017, CloudSim	97.24
[45]	2022	RF, GA	NSL-KDD UNSW-NB15	92 96
[43]	2019	RF, GBM, Adaboost	NSL-KDD	99.5
[46]	2020	DT, J48	OneM2Mdata	92
[47]	2021	CNN	Bot-IoT	-
[44]	2022	Ensemble learning	Bot-IoT wustl_IIoT_2021	99.99 99.12
[48]	2023	RF	NSL-KDD Bot-IoT	98.3 100
[49]	2021	LSTM	KDDCup'99 NSD-KDD DARPA KDD CSE-CIC-IDS2018	99.05

As a result of the mentioned research, it is noticeable that reliable IDS approaches are obtained using ML and DL algorithms. According to this, we propose a novel model combining RF and an RBFNN to detect intrusion in the cloud.

RF is an ensemble learning method based on DTs. It works by constructing multiple DTs during training and outputs the class, that is, the mode of the categories or the mean prediction of the individual trees [50].

RF is well suited for our IDS model because it handles high-dimensional datasets with many features [48]. It performs feature selection naturally by evaluating the importance of each feature based on how much they contribute to the overall ACC of the model [27,48]. Moreover, RF is robust against overfitting as each tree is trained on a random subset of the data and uses a random subset of the features for node splitting. It helps to reduce variance and enhance the generalization of the model. In intrusion detection, where the dataset may have many features and potential imbalances, RF's ability to handle these challenges makes it a suitable choice [27,50].

An RBFNN is a type of neural network that uses radial basis functions as activation functions in its hidden layer. The activation function transforms the input data into the hidden layer [27,50]. RBFNNs are particularly effective in handling non-linear problems and are well suited for pattern recognition tasks. They are suitable for intrusion detection scenarios where attacks can be complex and non-linear [27,50,51]. The architecture of

an RBFNN allows it to approximate complex decision boundaries efficiently, leading to improved performance in capturing the underlying patterns in the data. Additionally, an RBFNN's training process is relatively faster than that of traditional feedforward neural networks, making it computationally efficient for large-scale intrusion detection tasks [50].

While RF and RBFNNs have demonstrated their effectiveness in handling intrusion detection tasks, it is also essential to consider other algorithms' performances. Each algorithm has its strengths and limitations, and the choice of the most appropriate one depends on the dataset's specific characteristics and the problem's requirements.

Our research thoroughly compared various ML and DL algorithms, including DT, SVM, and feedforward neural networks. RF and RBFNNs emerged as top-performing algorithms based on multiple evaluation metrics, including ACC, precision, recall, and F1-score. These algorithms showed promising results in handling intrusion detection's complex and dynamic nature in CC environments.

In conclusion, the selection of RF and an RBFNN in our IDS model is rooted in their ability to handle high-dimensional datasets, adapt to non-linear patterns, and efficiently detect intrusions in cloud environments. We have considered their strengths and performance compared to other algorithms to ensure the effectiveness and robustness of our proposed approach.

3. Our Approach

This section presents the details of the RF-RBFNN-IDS model proposed in our study. We have outlined all the model construction techniques, including feature reduction, to enhance prediction and processing time. The proposed approach for cloud security involves two processes, preprocessing and intrusion detection, for which the RBFNN and RF are employed, as illustrated in Figure 1 and Algorithm 1.

Algorithm 1: Feature Reduction Algorithm

Input:

- Nsl: NSL-KDD dataset
- Bot: Bot-IoT dataset
- Rf: RF Model
- Model: RBFNN Model

Output:

- MeasuresTabNsl: ACC, Precision, Recall, and MCC
- MeasuresTabBot: ACC, Precision, Recall, and MCC

Variables:

- ScalerNsl: Standardized NSL-KDD dataset
- ScalerBot: Standardized Bot-IoT dataset
- Snsl: The NSL-KDD selected features.
- Sbot: The best Bot-IoT selected features.

Begin:

```
ScalerNsl = Normalize (Snsl)
ScalerBot = Normalize (Sbot)
Snsl = Rf (preprocess (ScalerNsl))
Sbot = Rf (preprocess (ScalerBot))
Model = Hyperparameter (Model (Snsl))
Model = Model.fit(SnslTrain)
MeasuresTabNsl = Calculation (Model.predict (SnslTest))
Model = Hyperparameter (Model (Sbot))
Model = Model.fit(SbotTrain)
MeasuresTabBot = Calculation (Model.predict (SbotTest))
Display(MeasuresTabNsl, MeasuresTabBot)
```

End.

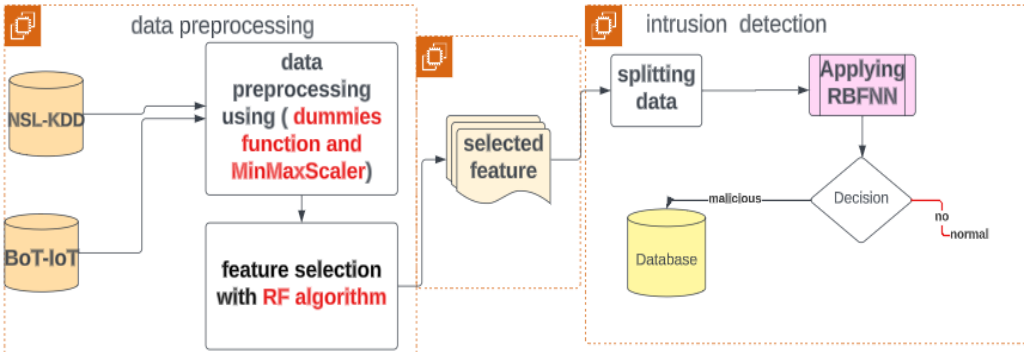

Figure 1. Scheme of the RF-RBFNN-IDS.

3.1. Our Proposed IDS

In our research, we have incorporated several key optimizations to improve the efficiency of our proposed method compared to previous studies using similar datasets and methodologies:

- Parallel processing: In our implementation, we leverage parallel processing techniques to use modern multi-core processors and accelerate the computation. By distributing the workload across multiple cores, we can significantly reduce the processing time, especially when dealing with large-scale datasets.
- Optimized data structures: We have employed efficient data structures to store and access the dataset, ensuring quick access and retrieval during the training and testing. This optimization minimizes memory usage and improves the overall computational efficiency.
- Data preprocessing and normalization: Proper data preprocessing, including converting categorical attributes into numerical values and the normalization of feature values. It ensures consistent scaling and faster convergence during training. These preprocessing steps improve efficiency by reducing the computational burden and minimizing convergence time.
- Feature reduction with RF: We can identify the most relevant features contributing significantly to intrusion detection by utilizing RF as a feature selection method. This step reduces the dimensionality of the data and focuses the model on the most informative attributes, resulting in faster processing and improved efficiency.
- Smart batching: When training the RBFNN classifier, we employ intelligent batching techniques to batch data efficiently, reducing memory consumption and speeding up the learning process.
- Optimized RBFNN hyperparameters: We carefully tuned the hyperparameters of the RBFNN classifier. This optimization process ensures that the RBFNN performs efficiently and effectively in detecting intrusions. By finding the right balance between complexity and performance, we avoid unnecessary computational overhead, leading to better efficiency.

By implementing these efficiency-enhancing techniques, we aim to demonstrate the improved performance of our proposed method compared to other studies using similar datasets and methodologies. Viarigorous experimentation and comparative analysis, we can provide concrete evidence of the efficiency gains achieved by our approach.

3.2. Data Preprocessing

The datasets comprise a mix of numerical and categorical features. To improve data quality, we converted categorical attributes into numerical values [52–54]. Additionally, we transformed character-based value systems into [0, 1] using the pandas get dummies function [55]. We then normalized the features to the [0, 1] range to ensure consistent scaling, providing advantages such as faster data collection, reduced bias, more straightforward analysis, and improved convergence and training time [52–54].

Additionally, feature reduction involves selecting relevant features and discarding insignificant ones to obtain a subset that accurately reflects the classification process. In this step, we utilized the Reduction algorithm based on the RF classifier to select the most informative and minimal features. Our contribution significantly enhances IDS performance and reduces processing time.

To identify the most practical features, we applied the RF algorithm. As a result, we obtained three features in the Bot-IoT dataset and four features in the NSL-KDD dataset. The RF selector played a crucial role in this feature reduction process. The algorithm returns the best-selected features based on ACC. The RF classifier, also known as random decision forests, is a group of learning methods used for classification or regression. It creates multiple DTs during development and derives the final category [27]. The RF classifier is particularly attentive to outlier data, which helps overcome prediction errors in the learning algorithm. It automatically generates accuracy and variable importance scores [51].

3.3. Intrusion Detection

After feature selection, we utilized the RBFNN classifier for network intrusion detection. An RBFNN is a powerful and efficient Deep Learning algorithm that optimizes functions. It consists of three layers, including input, hidden, and output, as shown in Figure 2, which effectively address classification problems [50].

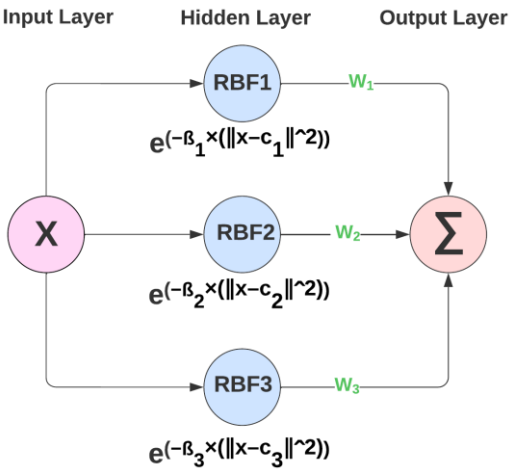

Figure 2. Scheme of RBFNN model.

The input layer collects and feeds inputs to the RBF network's unique hidden layer. The hidden layer uses non-linear functions primarily based on RBFs [56]. Each node in the hidden layer, represented by RBF1, is a vector of n nodes expressing the RBF of [x1, x2, . . ., xn], with C1 being the first clustering vector. The RBF1 vector calculates the distance between the first centroid and the data using Equation (1):

$$e\hat{} - \beta i \times ||x - ci||\hat{}2, \quad (1)$$

$$\beta i = \sqrt{(2 \times K)/Dmax}, \quad (2)$$

where K is the number of clusters, and Dmax is the maximum Euclidean distance between each of the two sets.

Based on the RBFs, the output layer performs the prediction task, such as classifications. The challenge lies in determining [w1, w2, w3], which most significantly represents the linear association. Here, the principal advantage of the RBFNN lies in using the Least-Squares Linear Regression equation, enabling the rapid attainment of the global optimum of the minimization problem [27,51].

The RBFNN's unique architecture and efficient optimization make it an ideal choice for intrusion detection tasks in cloud computing environments.

4. Experimental Setting

4.1. Experiment Environment and Datasets

This study's investigation was conducted and assessed in a controlled environment using a Windows 10 Professional 64-bit PC powered by a 1.8 GHz Cortex TM-i5 8250U CPU. The suggested framework is implemented using Python 3. We assess and contrast our approach using the confusion matrix. This study makes use of two datasets. The NSL-KDD dataset is an updated version of the KDD, which was created to address many significant problems with the KDD 1999 dataset [57]. It offers the following benefits: In comparison to the KDD dataset, it exempts redundant records. The number of records is adequate, and the selected files are organized as a percentage of the total records: eKDDTrain+ 20Percent.ARFF. The NSL-KDD [57] contains 41 features from the KDD'99 dataset.

In addition, the fact that the Bot-IoT [58] dataset collection includes IoT apps makes it more sophisticated [59,60]. This dataset provides information on various types of IoT traffic, including malware, the IoT, and regular traffic [61]. The NSL-KDD and Bot-IoT datasets contain many variables, including 41 features and 46 features, respectively. Both datasets are detailed in Table 2. This work aims to use a minimum number of variables in each dataset by improving the quality of the intrusion detection model. Table 3 depicts the features chosen randomly from each dataset.

Table 2. Dataset descriptions.

Dataset	Number of Features	Class	Total
NSL-KDD	41	Normal, DoS, Probe, Remote to Local (R2L), User to Root (U2R).	125,192
Bot-IoT	46	Normal, DoS, DDoS, Information Gathering, Information Theft.	73,370,443

Table 3. The used features.

Dataset	Number of Features	Features
NSL-KDD	10	"dst_bytes", "src_bytes", "flag", "logged_in", "same_srv_rate", "protocol_type", "dst_host_srv_count", "dst_host_same_srv_rate", "count", "dst_host_same_src_port_rate", "class".
NSL-KDD	4	"flag", "logged_in", "same_srv_rate", "protocol_type", "class".
Bot-IoT	10	"daddr", "TnP_PerProto", "TnP_PSrcIP", "saddr", "TnP_PDstIP", "TnBPSrcIP", "bytes", "stime", "TnP_Per_Dport", "TnBPDstIP", "attack".
Bot-IoT	3	"daddr", "TnP_PerProto", "TnP_PSrcIP", "attack".

4.2. Evaluation Metrics

The efficiency indicators that verified the suggested approach are briefly described in this subsection. Then, each efficiency metric's response to the proposed model is detailed in the following subsection. A confusion matrix was produced to assess the effectiveness of the algorithm, as shown in Table 4, and these metrics, including ACC, precision, recall, and MCC, are calculated.

Table 4. The confusion matrix.

	Actually Positive	**Actually Negative**
Predict positive	True positive (TP)	False positive (FP)
Predict negative	False negative (FN)	True negative (TN)

It should be noted that the entries within a confusion matrix (TP, FP, FN, and TN) are defined as follows:

- TP: The model shows the attack as true, which it is.
- TN: The model shows normal as false, but it is true.
- FP: The model shows an attack, yet it does not occur.
- FN: The model shows normal but is incorrect.
- MCC: Examine the impact of our model on the dataset's imbalance. We used the MCC to assess the dependability of our classifier. The MCC's strength is that it takes into account the confusion matrix's four categories.

In addition, the used metrics are described as follows:

$$\text{ACC} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}; \tag{3}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}; \tag{4}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{TN}}; \tag{5}$$

$$\text{MCC} = \frac{\text{TP} \times \text{TN} - \text{FP} \times \text{FN}}{\sqrt{(\text{TP} + \text{TN})(\text{TP} + \text{FP})(\text{TN} + \text{FP})(\text{TN} + \text{FN})}}; \tag{6}$$

Intrusion detection scenarios often encounter imbalanced datasets, where the occurrences of standard instances significantly outweigh those of rare intrusions. We have implemented several comprehensive countermeasures to address this challenge and ensure the credibility and ACC of our proposed method. Firstly, we employed resampling techniques to balance the class distribution in the training dataset. By over-sampling the minority class or undersampling the majority class, we ensure that the model learns from a representative set of positive and negative instances. Additionally, to augment the minority class and further enhance its representation, we utilized synthetic data generation techniques, such as the Synthetic Minority Over-sampling Technique (SMOTE). This approach generates synthetic samples of the minority class, effectively increasing its presence in the dataset.

Moreover, we incorporated cost-sensitive learning, assigning different misclassification costs to each class to prioritize the correct prediction of the minority class. It encourages the model to focus on accurately detecting intrusions, even if it increases false positives for the majority class. Additionally, we harnessed ensemble methods, like boosting and bagging, to combine multiple classifiers and improve overall performance, particularly for the underrepresented category. Lastly, we carefully selected evaluation metrics, including the F1-score and MCC, as shown in Equation (6). Usingthese comprehensive countermeasures, we ensure that our proposed method effectively handles the challenges posed by imbalanced datasets in intrusion detection, leading to more accurate and robust results.

5. Results and Discussions

This section discusses our model results on the NSL-KDD and the Bot-IoT datasets.

5.1. NSL-KDD Dataset

Table 5 and Figure 3 present different measures to evaluate our model on the NSL-KDD dataset. The full dataset scored a 90.49% ACC, 91.69% precision, 48.05% recall, and 81% MCC. Even with these high scores, our feature selection model eliminates many features and maintains the model's incredible performance. As proof, the ten and the four selected elements scored, respectively, 92.12% and 94.16% ACC; 91.12% and 90.83% precision; 46.9% and 45.74% recall; and 84.19% and 88.39% MCC. The four selected features scored the higher MCC by distinguishing between regular instances and attacks. We have used ten features from the NSL-KDD to discuss the results. The main goal of this study is to find the minimum features we can use from this subset to enhance our model. We have tested all the possibilities (one feature, two, and three), but the best performance is represented using four selected features.

Table 5. Performance metrics on the NSL-KDD dataset.

Features	ACC (%)	Precision (%)	Recall (%)	MCC (%)
Full Dataset	90.49	91.69	48.05	81.00
10 Features	92.12	91.12	46.90	84.19
4 Features	94.16	90.83	45.74	88.39

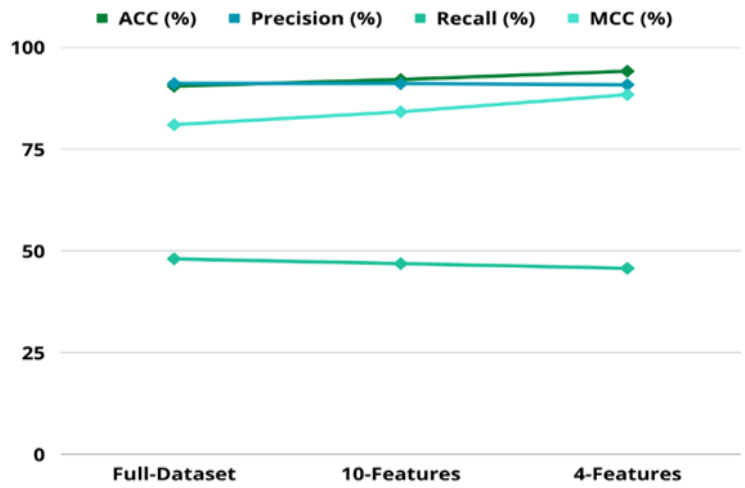

Figure 3. Different metrics measure the model performance on the NSL-KDD dataset.

Figure 4 illustrates a comparison histogram of TP, TN, FP, and FN scored by the NSL-KDD dataset. By observing the figure, we can deduce that our feature selection method helps the model to maintain superior performance in detecting positive instances and to boost the detection of negative cases. The full feature of the NSL-KDD dataset scored 89% on TN, the ten selected features scored 93%, and the four chosen elements scored 97.2%.

Figures 5–7 demonstrate the four measures of the confusion matrix—TP, TN, FP, and FN—of the full NSL-KDD dataset, ten selected features, and four selected features.

Figure 5 displays the confusion matrix of the full NSL-KDD dataset. The model scored 89% TN, 11% FN, 8.3% FP, and 92.7% TP.

Figure 6 shows the confusion matrix of the ten selected features from the NSL-KDD dataset. The model depicted significant development in detecting the negative instances with 93% TN and 7% FN and maintained the excellent performance of the whole dataset in distinguishing the positive samples with 8.9% FP and 91.1% TP.

Figure 7 describes the confusion matrix of the four selected features from the NSL-KDD dataset. Our model showed superior results in detecting the negative instances with 97.2% TN and 2.8% FN and maintained the outstanding performance of the entire

dataset and ten selected features when distinguishing the positive samples with 9.2% FP and 91.8% TP.

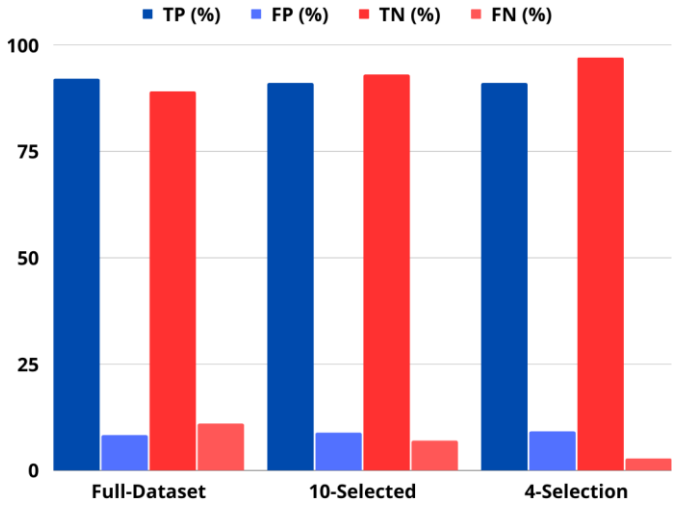

Figure 4. The NSL-KDD dataset confusion matrix comparison.

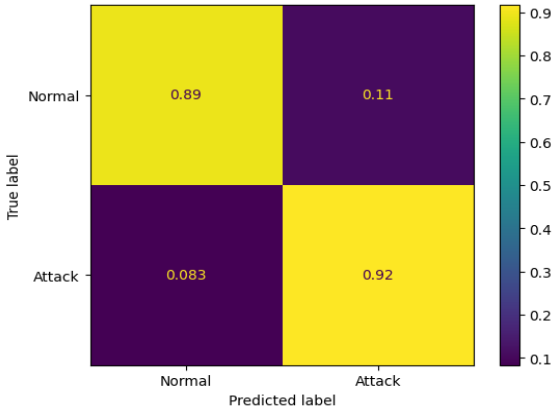

Figure 5. The full NSL-KDD dataset confusion matrix.

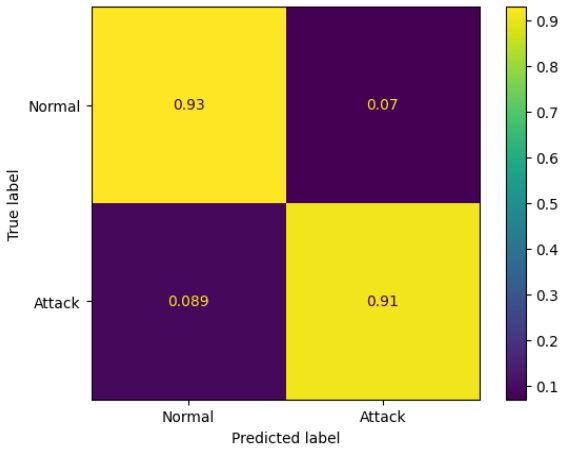

Figure 6. The ten selected features from the NSL-KDD dataset confusion matrix.

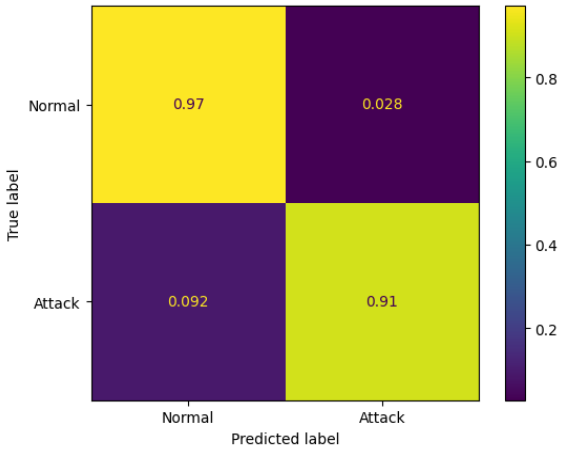

Figure 7. The four selected features from the NSL-KDD dataset confusion matrix.

5.2. Bot-IoT Dataset

Table 6 and Figure 8 present different measures to evaluate our model on the Bot-IoT dataset. The full dataset scored 99.98% ACC, 100% precision, and 99.99% recall. Our feature selection model eliminates many features and maintains the same performance on the ACC, precision, and recall. We integrated the MCC measure to prove our model worked well with an imbalanced dataset like the Bot-IoT. As proof, the entire dataset scored 28.47% on MCC, seeing that the model could not adequately recognize the typical instances due to its few numbers in the dataset.

Table 6. Performance metrics on the Bot-IoT dataset.

Features	ACC (%)	Precision (%)	Recall (%)	MCC (%)
Full Dataset	99.98	100	99.99	28.47
10Features	99.99	100	99.99	83.83
3Features	99.99	100	99.98	93.00

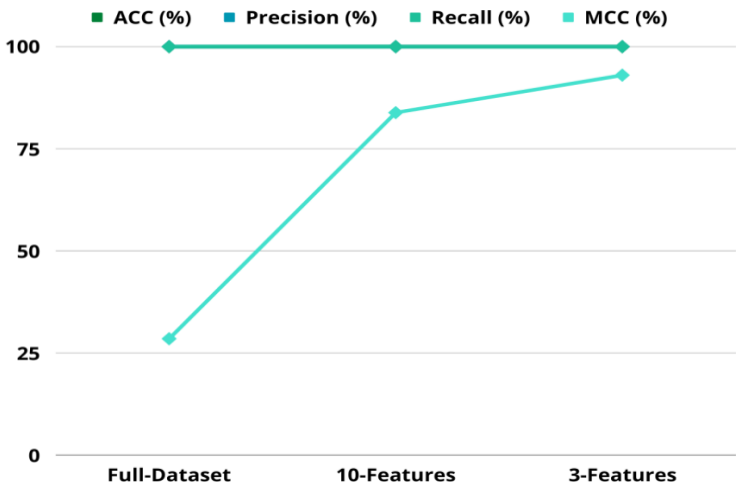

Figure 8. Different metrics measure the model performance on the Bot-IoT dataset.

On the other hand, our model helped reduce this issue's impact on the results. The ten and then the three selected features scored 83.83% and 93% MCC, respectively, which are way better than the 28.47% scored for the entire dataset. Consequently, the three chosen features achieved the higher MCC by distinguishing between regular instances and attacks well.

Figure 9 illustrates a comparison histogram of the confusion matrix measures TP, TN, FP, and FN scored by the Bot-IoT dataset. Based on the figure, our feature selection method helps the model to perform well in maintaining superior performance when detecting positive instances. It helps boost the detection of regular cases. Where the full feature of the NSL-KDD dataset scored 8.1% on TN, the ten selected features scored 70%, and the three chosen elements scored 86%.

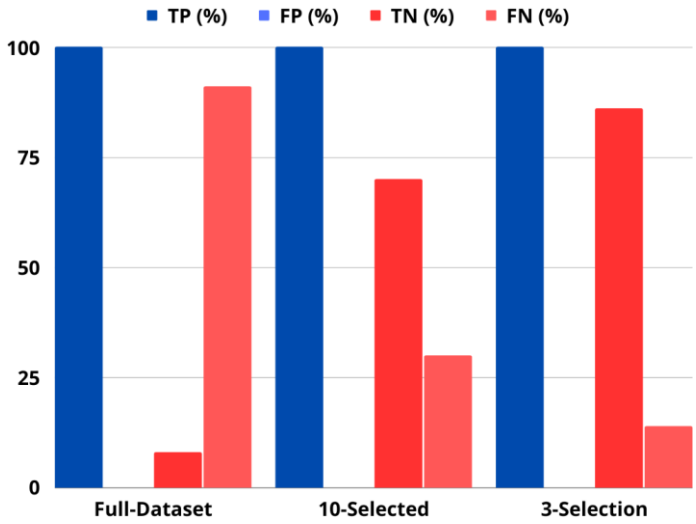

Figure 9. The Bot-IoT dataset confusion matrix comparison.

Figures 10–12 depict the four measures, including TP, TN, FP, and FN, of the full Bot-IoT dataset's confusion matrix, ten selected features, and three selected features.

Figure 10 shows the confusion matrix of the full Bot-IoT dataset. The model scored 8.1% TN, 92.9% FN, 0% FP, and 100% TP. As the results show, our model performed poorly in detecting the negative instances, which could lead to blocking many friendly data and packets. These unsatisfactory results were due to the imbalance between the standard and attack instances in the Bot-IoT dataset. To reduce the impact of this imbalance on the results, we applied the feature selection as mentioned above.

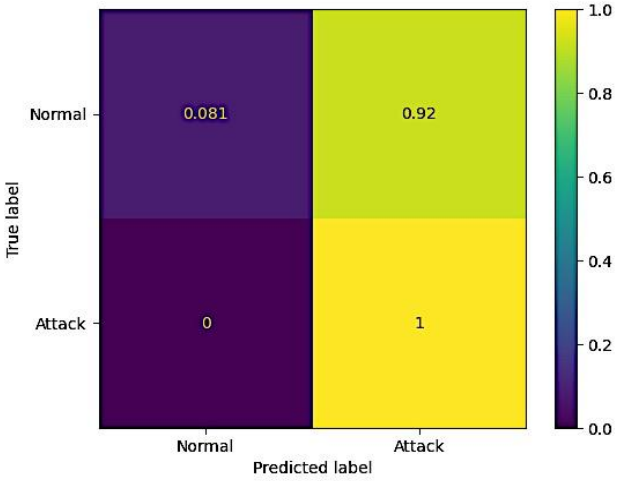

Figure 10. The full Bot-IoT dataset confusion matrix.

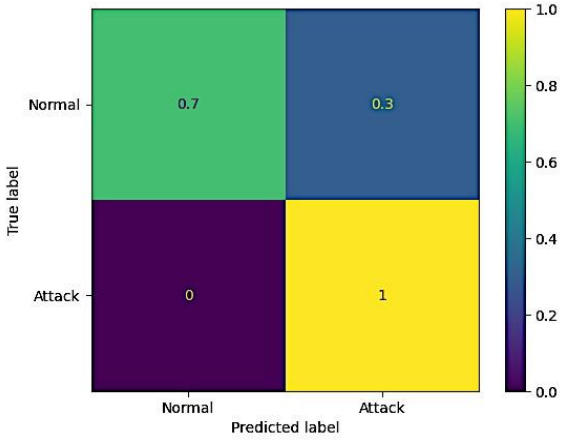

Figure 11. The ten selected features from the Bot-IoT dataset confusion matrix.

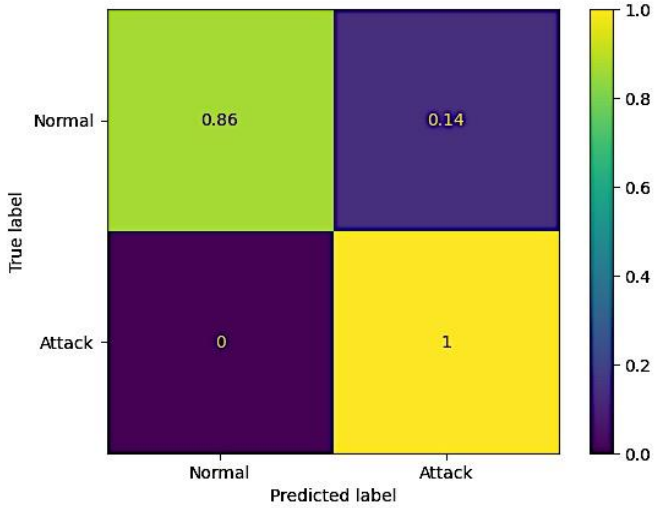

Figure 12. The three selected features from the Bot-IoT dataset confusion matrix.

We obtained the outcomes depicted in Figure 11 after executing the feature selection algorithm and choosing the ten best-performing features. The confusion matrix for the selected ten features from the Bot-IoT dataset is shown in this figure. The model showed significant improvement in detecting the negative instances with 70% TN and 30% FN and maintained the excellent performance of the entire dataset in distinguishing the positive samples with 0% FP and 100% TP. We again applied the feature selection to see if we could reduce the features and gain or maintain the same results. Thus, we obtained the results shown in Figure 12.

Figure 12 describes the confusion matrix of the three selected features from the Bot-IoT dataset as the last point before we started to lose our good results, even with testing every two components together. So, our model showed exciting results in detecting the negative instances with 86% TN and only 14% FN and maintained the outstanding performance of the entire dataset and ten selected features when distinguishing the positive examples with 0% FP and 100% TP. Our model showed efficiency in overtaking the impact of the Bot-IoT imbalance, as shown in Figure 12. To test the performance on a more balanced dataset, we evaluated our model on the NSL-KDD dataset, and the results remain very courageous.

As shown in Table 7, several methods have been explored to achieve high ACC. Three notable studies [14,37,44], employed different techniques to tackle this challenge using the Bot-IoT and NSL-KDD datasets. The ensemble learning approach was adopted in [37], resulting in an impressive ACC of 99.99%. Similarly, [44] implemented the KNN algorithm and achieved the same remarkable ACC. In [14], the authors utilized gradient

boosting with DT and achieved a perfect accuracy of 100% based on the NSL-KDD dataset. Our study proposed a novel RF-RBFNN model and evaluated its performance using the Bot-IoT and NSL-KDD datasets. Remarkably, our model achieved an accuracy of 99.99% on the Bot-IoT dataset and 94.16% on the NSL-KDD dataset. Notably, despite achieving comparable ACC rates, our proposed model utilized a minimum of features confronted by the previous works. It shows the effectiveness and efficiency of our model in achieving high performances while reducing the dimensionality of the feature space. Overall, these findings highlight the promising outcomes of different methods employed for intrusion detection in the IoT and cloud environments. Despite utilizing fewer features, each method demonstrated exceptional ACC, and our proposed RF-RBFNN model showcased competitive performance. These advancements contribute to enhancing the security of different systems.

Table 7. Model's performance comparison.

Article	Methods	Dataset	ACC (%)
[37]	Ensemble learning	Bot-IoT	99.99
[44]	KNN	Bot-IoT	99.99
[14]	Gradient boosting DT	Bot-IoT NSL-KDD	100 100
Our proposed model	RF-RBFNN	Bot-IoT NSL-KDD	99.99 94.16

6. Conclusions

Intrusion detection has significantly benefited from advancements in cyber security, particularly with the incorporation of ML and DL algorithms. This paper presented a novel technique for detecting intrusions in a cloud environment by combining ML and DL algorithms, explicitly utilizing a reduction algorithm based on the RF classifier for feature selection and the RBFNN for intrusion detection. The results obtained from our approach demonstrate its effectiveness in detecting intrusions, achieving an ACC rate higher than 94% and an FNR lower than 0.0831%. This showcases the capability of our model to identify and classify intrusions in the cloud environment accurately. Additionally, the utilization of feature selection methods has proved to be instrumental in enhancing the overall performance of the IDS. One notable strength of our model is its ability to achieve high ACC rates and reduce prediction time by utilizing a limited number of variables. Our model improves the ACC rate and enhances operational efficiency by leveraging carefully selected features.

Moreover, our model successfully addressed the challenges posed by imbalanced datasets, such as the Bot-IoT dataset, by effectively balancing the classification of high-dimensional data. The feature selection approach helped increase the TN from 8.1% when using all features to 86% with only three selected features.

In summary, our study presents a promising technique for intrusion detection in a cloud environment by combining ML and DL algorithms. The results validate the effectiveness of our approach, showcasing its potential for enhancing cyber security in cloud-based systems. With future advancements in feature engineering and dimensionality reduction, we anticipate even more significant improvements in the performance and efficiency of our model.

As part of our future work, we aim to advance our feature engineering techniques by incorporating dimensionality reduction methods. This enhancement will enable our model to perform even more efficiently by reducing the complexity of the input data. By exploring dimensionality reduction methods, we anticipate further improvements in the ACC and computational efficiency of our IDS.

Author Contributions: Conceptualization, H.A. and M.M.-e.; methodology, A.G.; software, M.M.-e.; validation, A.G., S.B. and M.A.; formal analysis, A.A.; investigation, N.A.; resources, A.G.; data curation, M.A.; writing—original draft preparation, H.A.; writing—review and editing, M.M.-e.; visualization, S.B.; supervision, A.G.; project administration, A.A.; funding acquisition, N.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: Assessments and Experimental results, obtained using Anaconda 3 IDE, are available andwill be shared with authors at https://sites-Google.com/umi.ac.ma/azrour, accessed on 22 August 2023.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Liu, I.-H.; Lo, C.-H.; Liu, T.-C.; Li, J.-S.; Liu, C.-G.; Li, C.-F. IDS Malicious Flow Classification. *J. Robot. Netw. Artif. Life* **2020**, *7*, 103. [CrossRef]
2. Tahirkheli, A.I.; Shiraz, M.; Hayat, B.; Idrees, M.; Sajid, A.; Ullah, R.; Ayub, N.; Kim, K.-I. A Survey on Modern Cloud Computing Security over Smart City Networks: Threats, Vulnerabilities, Consequences, Countermeasures, and Challenges. *Electronics* **2021**, *10*, 1811. [CrossRef]
3. Patel, H.B.; Kansara, N. Cloud Computing Deployment Models: A Comparative Study. *Int. J. Innov. Res. Comput. Sci. Technol.* **2021**, *9*, 45–50. [CrossRef]
4. Palumbo; Aceto, F.; Botta, G.; Ciuonzo, A.; Persico, D.; Pescapé, V. A Characterizing Cloud-to-user Latency as perceived by AWS and Azure Users spread over the Globe. In Proceedings of the 2019 IEEE Global Communications Conference, Big Island, HI, USA, 9–13 December 2019.
5. Hourani, H.; Abdallah, M. Cloud Computing: Legal and Security Issues. In Proceedings of the International Conference on Computer Science and Information Technology, Amman, Jordan, 11–12 July2018.
6. Hussein; Khalid, N.H. A survey of Cloud Computing Security challenges and solutions. *Int. J. Comput. Sci. Inf. Secur.* **2017**, *14*, 52.
7. Belal, M.M.; Sundaram, D.M. Comprehensive review on intelligent security defenses in the cloud: Taxonomy, security issues, ML/DL techniques, challenges and future trends. *J. King Saud Univ. Comput. Inf. Sci.* **2022**, *34*, 9102–9131.
8. El-Zoghbi, A.M.; Azer, M.A. Cloud Computing Privacy Issues, Challenges and Solutions. In Proceedings of the 2017 12th International Conference on Computer Engineering and Systems (ICCES), Cairo, Egypt, 19–20 December 2017.
9. Butt, U.A.; Mehmood, M.; Shah, S.B.H.; Amin, R.; Shaukat, M.W.; Raza, S.M.; Suh, D.Y.; Piran, J. A Review of Machine Learning Algorithms for Cloud Computing Security. *Electronics* **2020**, *9*, 1379. [CrossRef]
10. Al-Jaser, N.M.A. A Survey on Cloud Computing Security Challenges and Trust Issues. *Int. J. Comput. Sci. Inf. Secur.* **2020**, *18*, 7–12.
11. Namasudra, S.; Roy, P.; Balusamy, B.; Vijayakumar, P. Data accessing based on the popularity value for cloud computing. In Proceedings of the International Conference on Innovations in Information: Embedded and Communication Systems, Coimbatore, India, 17–18 March 2017.
12. Namasudra, S.; Roy, P.A. New Table Based Protocol for Data Accessing in Cloud Computing. *J. Inf. Sci. Eng.* **2017**, *33*, 585–609.
13. Chiba, Z.; Abghour, N.; Moussaid, K.; El omri, A.; Rida, M. A cooperative and hybrid network intrusion detection framework in cloud computing-based SNORT and optimized back propagation neural network. *Procedia Comput. Sci.* **2016**, *83*, 1200–1206. [CrossRef]
14. Douiba, M.; Benkirane, S.; Guezzaz, A.; Azrour, M. Anomaly detection model based on gradient boosting and decision tree for IoT environments security. *J. Reliab. Intell. Environ.* **2022**, 1–12. [CrossRef]
15. Padhy, S.; Dash, S.; Routray, S.; Ahmad, S.; Nazeer, J.; Alam, A. IoT-based hybrid ensemble machine learning model for efficient diabetes mellitus prediction. *Comput. Intell. Neurosci.* **2022**, *2022*, 2389636. [CrossRef]
16. Noor, M.; Hassan, H. Current research on the Internet of Things (IoT) security: A survey. *Comput. Netw.* **2018**, *148*, 283–294. [CrossRef]
17. Sethi, P.; Sarangi, S.R. Internet of Things: Architectures, Protocols, and Applications. *J. Electr. Comput. Eng.* **2017**, *2017*, 9324035. [CrossRef]
18. Khraisat, A.; Gondal, I.; Vamplew, P.; Kamruzzaman, J. Survey of intrusion detection systems: Techniques, datasets and challenges. *Cybersecurity* **2019**, *2*, 20. [CrossRef]
19. Almseidin, M.; Alzubi, M.; Kovacs, S.; Alkasassbeh, M. Evaluation of machine learning algorithms for intrusion detection system. In Proceedings of the 2017 IEEE 15th International Symposium on Intelligent Systems and Informatics (SISY) 2017, Avadi, India, 6–8 May 2017.

20. Buczak, A.L.; Guven, E. A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Commun. Surv. Tutor.* **2016**, *18*, 1153–1176. [CrossRef]
21. Li, K.; Gibson, C.; Ho, D.; Zhou, Q.; Kim, J.; Buhisi, O.; Brown, D.E.; Gerber, M. Assessment of machine learning algorithms in cloud computing frameworks. In Proceedings of the 2013 IEEE Systems and Information Engineering Design Symposium, Charlottesville, VA, USA, 26 April 2013; pp. 98–103. [CrossRef]
22. Dritsas, E.; Trigka, M. Efficient Data-Driven Machine Learning Models for Water Quality Prediction. *Computation* **2023**, *11*, 16. [CrossRef]
23. Guezzaz, A.; Benkirane, S.; Azrour, M. A Novel Anomaly Network Intrusion Detection System for Internet of Things Security. In *IoT and Smart Devices for Sustainable Environment*; Springer International Publishing: Cham, Switzerland, 2022. [CrossRef]
24. Guezzaz, A.; Asimi, A.; Asimi, Y.; Tbatou, Z.; Sadqi, Y. A Lightweight Neural Classifier for Intrusion Detection. *Gen. Lett. Math.* **2017**, *2*, 57–66. [CrossRef]
25. Ahmad, S.; Jha, S.; Alam, A.; Alharbi, M.; Nazeer, J. Analysis of intrusion detection approaches for network traffic anomalies with comparative analysis on botnets (2008–2020). *Secur. Commun. Netw.* **2022**, *2022*, 9199703. [CrossRef]
26. Mahadik, S.; Pawar, P.M.; Muthalagu, R. Efficient Intelligent Intrusion Detection System for Heterogeneous Internet of Things (HetIoT). *J. Netw. Syst. Manag.* **2023**, *31*, 2. [CrossRef]
27. Torres, J.M.; Comesaña, C.I.; García-Nieto, P.J. Review: Machine learning techniques applied to cybersecurity. *Int. J. Mach. Learn. Cybern.* **2019**, *10*, 2823–2836. [CrossRef]
28. Fouda, M.; Ksantini, R.; Elmedany, W. A Novel Intrusion Detection System for Internet of Healthcare Things Based on Deep Subclasses Dispersion Information. *IEEE Internet Things J.* **2022**, *10*, 8395–8407. [CrossRef]
29. Elghaish, F.; Matarneh, S.T.; Alhusban, M. The application of "deep learning" in construction site management: Scientometric, thematic and critical analysis. *Constr. Innov.* **2022**, *22*, 580–603. [CrossRef]
30. Halbouni, A.; Gunawan, T.S.; Habaebi, M.H.; Halbouni, M.; Kartiwi, M.; Ahmad, R. Machine Learning and Deep Learning Approaches for CyberSecurity: A Review. *IEEE Access* **2022**, *10*, 19572–19585. [CrossRef]
31. Chaabouni, N.; Mosbah, M.; Zemmari, A.; Sauvignac, C.; Faruki, P. Network Intrusion Detection for IoT Security Based on Learning Techniques. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 2671–2701. [CrossRef]
32. Hady, A.A.; Ghubaish, A.; Salman, T.; Unal, D.; Jain, R. Intrusion Detection System for Healthcare Systems Using Medical and Network Data: A Comparison Study. *IEEE Access* **2020**, *8*, 106576–106584. [CrossRef]
33. Guezzaz, A.; Azrour, M.; Benkirane, S.; Mohy-Eddine, M.; Attou, H.; Douiba, M. A Lightweight Hybrid Intrusion Detection Framework using Machine Learning for Edge-Based IIoT Security. *Int. Arab. J. Inf. Technol.* **2022**, *19*, 822–830. [CrossRef]
34. Hazman, C.; Guezzaz, A.; Benkirane, S.; Azrour, M. IDS-SIoEL: Intrusion Detection Framework for IoT-based Smart Environments Security using Ensemble Learning. *Clust. Comput.* **2022**, 1–15. [CrossRef]
35. Douiba, M.; Benkirane, S.; Guezzaz, A.; Azrour, M. An improved anomaly detection model for IoT security using decision tree and gradient boosting. *J. Supercomput.* **2022**, *79*, 3392–3411. [CrossRef]
36. Alshammari, A.; Aldribi, A. Apply machine learning techniques to detect malicious network traffic in cloud computing. *J. Big Data* **2021**, *8*, 90. [CrossRef]
37. Mohy-Eddine, M.; Guezzaz, A.; Benkirane, S.; Azrour, M. An efficient network intrusion detection model for IoT security usingK-NN classifier and feature selection. *Multimed. Tools Appl.* **2023**, *82*, 23615–23633. [CrossRef]
38. Jiang, F.; Fu, Y.; Gupta, B.B.; Liang, Y.; Rho, S.; Lou, F.; Meng, F.; Tian, Z. Deep Learning Based Multi-Channel Intelligent Attack Detection for Data Security. *IEEE Trans. Sustain. Comput.* **2018**, *5*, 204–212. [CrossRef]
39. Burhan, F.; Mustafa, G.; Nawaz, A.; Kiani, A.; Ali, T. Securing Cloud Data: A Machine Learning based Data Categorization Approach for Cloud Computing. *Res. Sq.* **2022**. [CrossRef]
40. Mubarakali, A.; Srinivasan, K.; Mukhalid, R.; Jaganathan, S.C.B.; Marina, N. Security challenges in internet of things: Distributed denial of service attack detection using support vector machine-based expert systems. *Comput. Intell.* **2020**, *36*, 1580–1592. [CrossRef]
41. Mishra, A.; Gupta, B.B.; Perakovic, D.; Penalvo, F.J.G.; Hsu, C.-H. Classification Based Machine Learning for Detection of DDoS attack in Cloud Computing. In Proceedings of the International Conference on Consumer Electronics, Las Vegas, NV, USA, 10–12 January 2021. [CrossRef]
42. Singh, P.; Ranga, V. Attack and intrusion detection in cloud computing using an ensemble learning approach. *Int. J. Inf. Technol.* **2021**, *13*, 565–571. [CrossRef]
43. Verma, A.; Ranga, V. Machine Learning Based Intrusion Detection Systems for IoT Applications. *Wirel. Pers. Commun.* **2020**, *111*, 2287–2310. [CrossRef]
44. Mohy-Eddine, M.; Guezzaz, A.; Benkirane, S.; Azrour, M. An effective intrusion detection approach based on ensemble learning for IIoT edge computing. *J. Comput. Virol. Hacking Tech.* **2022**, 1–13. [CrossRef]
45. Liu, Z.; Shi, Y. A Hybrid IDS Using GA-Based Feature Selection Method and Random Forest. *Int. J. Mach. Learn. Comput.* **2022**, *12*, 43–50.
46. Chaabouni, N.; Mosbah, M.; Zemmari, A.; Sauvignac, C. A OneM2M Intrusion Detection and Prevention System based on Edge Machine Learning. In Proceedings of the IEEE/IFIP Network Operations and Management Symposium, Budapest, Hungary, 20–24 April 2020. [CrossRef]

47. Ullah, I.; Mahmoud, Q.H. Design and Development of a Deep Learning-Based Model for Anomaly Detection in IoT Networks. *IEEE Access* **2021**, *9*, 103906–103926. [CrossRef]
48. Attou, H.; Guezzaz, A.; Benkirane, S.; Azrour, M.; Farhaoui, Y. Cloud-Based Intrusion Detection Approach Using Machine Learning Techniques. *Big Data Min. Anal.* **2023**, *6*, 311–320. [CrossRef]
49. Wani, A.; Revathi; Khali, R. SDN-basedintrusiondetectionsystemforIoTusingdeeplearningclassifier(IDSIoT-SDL). *CAAI Trans. Intell. Technol.* **2021**, *6*, 281–290. [CrossRef]
50. Albahar, M.; Alharbi, A.; Alsuwat, M.; Aljuaid, H. A Hybrid Model based on Radial basis Function Neural Network for Intrusion Detection. *Int. J. Adv. Comput. Sci. Appl.* **2020**, *11*, 781–791. [CrossRef]
51. Reis, I.; Baron, D.; Shahaf, S. Probabilistic Random Forest: A Machine Learning Algorithm for Noisy Data Sets. *Astron. J.* **2018**, *157*, 16. [CrossRef]
52. Kumar, P.; Gupta, G.P.; Tripathi, R. An ensemble learning and fog-cloud architecture-driven cyber-attack detection framework for IoMT networks. *Comput. Commun.* **2020**, *166*, 110–124. [CrossRef]
53. Alrashdi, I.; Alqazzaz, A.; Alharthi, R.; Aloufi, E.; Zohdy, M.A.; Ming, H. FBAD: Fog-based Attack Detection for IoT Healthcare in Smart Cities. In Proceedings of the 10th Annual Ubiquitous Computing, Electronics Mobile Communication Conference, New York, NY, USA, 10–12 October2019; pp. 515–522. [CrossRef]
54. Thamilarasu, G.; Odesile, A.; Hoang, A. An Intrusion Detection System for Internet of Medical Things. *IEEE Access* **2020**, *8*, 181560–181576. [CrossRef]
55. Palimote, J.; Atu, L.; Osuigbo, E. A Model to Detect Network Intrusion using Machine Learning. *J. Emerg. Technol. Innov. Res.* **2021**, *8*, 521–527.
56. Zhang, Q.; Wilson, F. RBNN application and simulation in big data set classification. *J. Intell. Fuzzy Syst.* **2019**, *37*, 4467–4475. [CrossRef]
57. Devarakonda, A.; Sharma, N.; Saha, P.; Ramya, S. Network intrusion detection: A comparative study of four classifiers using the NSL-KDD and KDD'99 datasets. *J. Phys. Conf. Ser.* **2022**, *2161*, 012043. [CrossRef]
58. Koroniotis, N.; Moustafa, N.; Sitnikova, E.; Turnbull, B. Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-iot dataset. *Future Gener. Comput. Syst.* **2019**, *100*, 779–796. [CrossRef]
59. Ferrag, M.A.; Maglaras, L.; Moschoyiannis, S.; Janicke, H. Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study. *J. Inf. Secur. Appl.* **2019**, *50*, 102419. [CrossRef]
60. Zeeshan, M.; Riaz, Q.; Bilal, M.; Shahzad, M.K.; Haider, H.J.; Rahim, A. Protocol-Based Deep Intrusion Detection for DoS and DDoS Attacks Using UNSW-NB15 and Bot-IoT Data-Sets. *IEEE Access* **2021**, *10*, 2269–2283. [CrossRef]
61. Shafiq, M.; Tian, Z.; Sun, Y.; Du, X.; Guizani, M. Selection of effective machine learning algorithm and Bot-IoT attacks traffic identification for the internet of things in smart city. *Future Gener. Comput. Syst.* **2020**, *107*, 433–442. [CrossRef]