Article

Not peer-reviewed version

Developing an Integrated Framework for Securing Internet of Things traffics in Smart Cities Using Machine Learning Techniques

Moody Alhanaya and Khalil Al-Shqeerat *

Posted Date: 13 July 2023

doi: 10.20944/preprints202307.0865.v1

Keywords: Internet of Things; Machine learning; Intrusion detection system; Ensemble classifier; Principal Component Analysis

 doi:10.20944/preprints202307.0865.v1

Article

Developing an Integrated Framework for Securing Internet of Things Traffics in Smart Cities Using Machine Learning Techniques

Moody Alhanaya and Khalil Al-Shqeerat *

Department of Computer Science, College of Computer, Qassim University, Saudi Arabia; m.alhanaya@qu.edu.sa
* Correspondence: kh.alshqeerat@qu.edu.sa

Abstract: The Internet of Things technology opens the horizon for a broader scope of intelligent applications in smart cities. However, the massive amount of traffic exchanged among devices may cause security risks, significantly when devices are compromised or vulnerable to cyber-attack. An intrusion detection system is the most powerful tool to detect unauthorized attempts to access smart systems. It identifies malicious and benign traffic by analyzing network traffic. In most cases, only a fraction of network traffic can be considered malicious. As a result, it is difficult for an intrusion detection system to detect attacks at high detection rates while maintaining a low false alarm rate. This work proposes an integrated framework to detect suspicious traffic to address secure data communication in smart cities. This paper presents an approach to developing an intrusion detection system to detect various attack types. It can be done by implementing a Principal Component Analysis method that eliminates redundancy and reduces system dimensionality. Furthermore, the proposed model shows how to improve intrusion detection system performance by implementing an ensemble model.

Keywords: internet of things; machine learning; intrusion detection system; ensemble classifier; principal component analysis

1. Introduction

Internet of Things (IoT) technology significantly increases the efficiency and productivity of smart cities. However, they must be protected against potential security threats and attacks. Whenever a cyber-attack escalates, security has become a major concern. In order to protect insightful information, security systems identify anomalies. Unfortunately, Security measures like encryption techniques and firewalls have been deployed to protect the system, but several attacks have bypassed them. Therefore, it is essential to recognize these attack patterns earliest to avoid losses to critical resources. Accordingly, appropriate actions can be taken to get rid of this intrusion.

A computer intrusion detection system (IDS) is one of the most effective tools for detecting attempts to access, manipulate, or shut down a computer system in an undesirable manner. It monitors the entire network's traffic from input and output, including resource activity, to protect the system from attacks [1]. IDS is an essential tool for designers of secure network systems, without which a considerable amount of data cannot be scanned in a second. One of the most promising approaches to improve the performance of IDSs is using machine learning (ML). This method can be used for both anomaly detection and misuse detection. In addition, an IDS can identify malicious and benign traffic by analyzing network traffic.

In most cases, only a fraction of the traffic that passes through a network can be considered malicious. This situation makes it hard for an IDS to identify attacks with high attack detection rates while keeping the false alarm rate low. One of the main challenges that IDSs face is the lack of ML models used to build an effective IDS. Researchers have now started to develop ensemble classifiers designed to combine multiple individual classifiers while improving the classification performance of an IDS [2]. For instance, if a single classifier is trained on a set of subsets of an IDS dataset, it could produce different results. However, by implementing an ensemble model, it can achieve better performance. However, due to the complexity of the network traffic attributes and the number of

 doi:10.20944/preprints202307.0865.v1

attack types that can be considered, ML models are also prone to experiencing time and computational issues. One of the most effective ways to improve the performance of an IDS is by implementing feature selection [3, 4]. This method can help the system identify highly relevant features and prevent useless ones from being detected.

This paper presents a novel approach to developing an IDS that can detect various types of attacks. It can be done through a Principal Component Analysis (PCA) method that eliminates redundancy and reduces the system's dimensionality. The proposed model shows how to improve the performance of an IDS by implementing an ensemble model. This method combines multiple decisions from multiple classifiers (Extra Trees (ET) [5, 6], K Nearest Neighbors (KNN) [7, 8], and Random Forts (RF)) into one model [9, 10]. The model is based on an Average-of-Probabilities (AOP) vote combination rule.

The combination of PCA and the ensemble model can improve the accuracy and stability of an IDS by reducing its time and computational issues. It can also generate an unbiased model to perform better analysis. Furthermore, the system performance must be analyzed before deploying it in the real world. So, an adequate dataset must be available to evaluate the performance of IDS. Therefore, a dataset is chosen based on the training and testing of the model. However, only limited datasets are publicly accessible, which remains a challenge today, and a few among them even lack comprehensiveness and completeness. Network Security Laboratory - Knowledge Discovery in Databases (NSL-KDD) is the commonly used dataset for IDS [11]. IDS faces many challenges, including misjudgment, false detection, and the absence of real-time responses.

The paper is organized into five sections. Section 2 introduces the literature review about IDSs solutions. The proposed methodology is demonstrated in Section 3. Section 4 shows, discusses, and analyzes the results of the experiments.

2. Literature Review

Several studies have been conducted to identify and detect attacks in smart cities. ML-based systems have proven to be effective in quickly detecting intrusion and working efficiently with a large amount of data, considering the destruction of the working principle and purpose of the system design. Feature selection must be considered to increase model performance and reduce data dimensionality by removing redundant or irrelevant features during the construction of an IDS.

In [12], the pigeon-inspired optimizer is introduced to identify features with a DT classifier to detect attacks by selecting the most important features from the data set. The researchers tried to compare feature selection techniques and evaluate their performance on three data sets: NSL-KDD, KDD cup'99, and UNSW-NB15. The study showed that the model's performance using NSL-KDD with 14 characteristics was 86.9% in accuracy. In [13], the RF classifier removed irrelevant traits from the data set. Several ML models, such as KNN, Support-Vector Machine (SVM), Decision Tree (DT), and Logistic Regression (LR), were used to train and test the model. As a result, the model achieved an accuracy of 99.3% and 99.2% detection rate on selected significant features (=10) from an entire set (=41) in the NSL-KDD data set. Hosseini [14] introduced an ML algorithm consisting of three main components; LR, genetic algorithm (GA), and Artificial neural network (ANN). The first stage involved extracting the trait set from the data. The AI neural network (NN) is trained to detect intrusions in the second stage. The performance of the proposed model was 94.4% in terms of accuracy. Two data sets were used to analyze the proposed model. One is the NSL-KDD dataset, and the other is the KDD cup'99. Although the proposed model is only 88.90%, the training and testing time for the proposed method was shallow as it was 74 seconds.

The differential evolution technique was introduced in [15] to reduce the number of traits in the data. This method mainly affects the accuracy of the intrusion detection system. The accuracy of the proposed model was 87.3% for the binary classification and 80.15% for the multiple classifications. The proposed model was evaluated against two data sets: NSL-KDD and KDD cup'99. The proposed model performed lower in accuracy compared to existing studies in intrusion detection systems.

Iram et al. in [16], studied network data classification by implementing multiple ML technologies such as SVM, KNN, LR, Naive Bayes(NB), Multi-layer Perceptron(MLP), RF, DT, and ET. Study results were evaluated on four subsets of the NSL-KDD data set, and an accuracy of over 99% was achieved using random forest classifiers, Extra Trees tree, and decision tree in all four feature subsets. To reduce features from the data set and eliminate noise, De la Hoz et al. used PCA and Fisher

 doi:10.20944/preprints202307.0865.v1

Discriminant Ratio (FDR) [17]. The authors developed a probabilistic self-organizing map model to model feature space and identify normal patterns from anomalous ones. The accuracy, specificity, and sensitivity results were 90%, 93%, and 97%, respectively.

Moreover, the clustering method combines several basic models to reduce false positive rates and produce more accurate solutions. Several studies have been conducted using ensemble methods to utilize the advantages of more than individual classifiers in one model. The authors in [18] demonstrated how ensemble machine learning, NN, and kernel methods can detect abnormal behavior in an IoT IDS. In this study, ensemble methods outperformed NN and kernels in accuracy by 99.1%. The study was evaluated using the Kitsune and NSL-KDD data sets. In [19], cyberattacks using cluster methods for IoT-based smart cities were revealed. The authors evaluated their new model using the UNSW-NB15 and CICIDS2017 datasets, and in this study, the proposed method was found to be consistent with LR, SVM, DT, RF, ANN, and KNN with an accuracy of 99.91%. Zhou et al. in [20] proposed IDS considers the correlation between the attributes and uses a feature selection method, the CFS-BA, to reduce the data's dimensionality. The system was then implemented by combining C4.5 and RF algorithms through voting. The results showed that the system achieved 99.8% accuracy on the NSL-KDD dataset with ten selected features. Even though the results showed adequate accuracy, the system was tested using only ten validation, which does not guarantee itself if the entire test data set is used. Even though there have been a number of studies [21, 22] in the field of IDS, a number of problems still need to be solved. An enhanced Long Short-Term Memory (LSTM) network was proposed by Elsayed et al. in [23] to distinguish between benign and malicious traffic, identify the attack category, and define the type of sub-attacks. ToN-IoT and InSDN datasets were used to assess the proposed system. The authors of [24] proposed a Transformer-based IoT NIDS method to determine the characteristics of attacks and their effects based on the types of data generated within the heterogeneous IoT environments. Using self-attention mechanisms, this method learns contextual embedding for input network features. It is capable of handling continuous and categorical features. The method uses network traffic data and telemetry information from IoT sensors to detect intrusions.

To tackle the same, a more diverse model was formed in the current work, where several classifiers that were not used in the prior research were combined to optimize the model performance. As such, a more robust model was obtained that could be effectively used for intrusion detection.

3. Methodology

To improve the detection capabilities of IDS, we propose an effective ML-based IDS using PCA. This method involves taking the data in a reduced form and keeping most of its original variance. The framework was built using voting, which is an ensemble of classifiers. The framework for developing an intelligent detection system (IDS) based on ML is shown in Figure 1.

 doi:10.20944/preprints202307.0865.v1

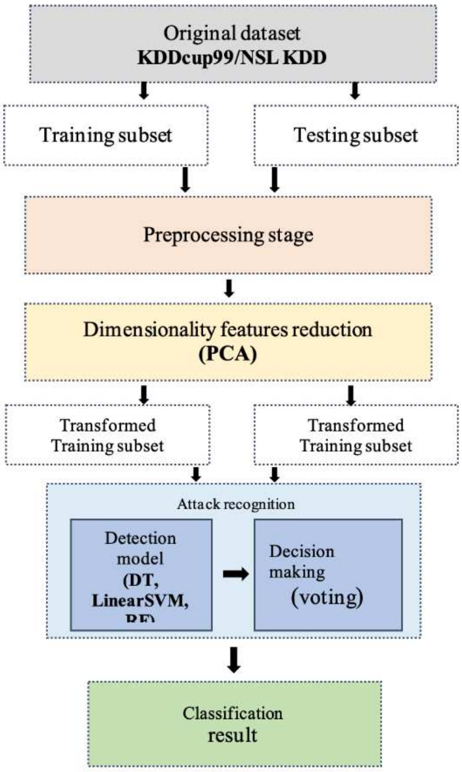

Figure 1. The framework of the PCA-Ensemble model.

In the dataset and preprocessing stage, the original dataset is processed to transform it into a suitable format for analysis. Then, the PCA method is applied in the dimensionality features reduction stage to scale down high-dimensional datasets by selecting the most appropriate features for each attack. Next, in the training classifier stage, three different classifiers were trained as base learners to improve the accuracy of the IDS using ET, KNN, and RF techniques. These classifiers are then used to create an ensemble classifier. After that, the attack recognition model is tested using a cross-validation approach and a voting technique to determine the probability of the base learners making the classification decision. Finally, benign traffic and various intrusive events can be classified and detected with high classification accuracy, according to the results of the ensemble classifier. Sections 3.1–3.3 provide detailed information about the stages of the proposed framework.

3.1. Dataset and Preprocessing

NSL-KDD dataset retained the original dataset's characteristics, such as its advantageous and challenging structure. The new version of the dataset addressed some drawbacks inherited from the previous version, reduced the number of instances, and maintained the diversity of selected samples. The NSL-KDD dataset was compiled to maximize its difficulty of prediction. In order to classify the records according to their difficulty level, several benchmark classifiers were used [25]. The number of selected records for each difficulty level group is inversely proportional to the number of records percentage from the original dataset. The KDDTrain+, KDDTest+, and KDDTest- 21 sets were used to classify the records in this study. The KDDTrain+ set comprises 125,973 instances composed of 67,343 instances of normal traffic and 58,630 instances of attack traffic. The KDDTest+ set contains over 22,000 in- stances, while the KDDTest- 21 set includes 11,850 instances.

The details of the dataset's instances are shown in Table 1. Each instance of the KDDTrain+ set has 42 attributes representing the different features of the connection. The values of these attributes are labeled as an attack or a normal.

Table 1. Statistics of the three sets of the NSL-KDD dataset.

Class	NSL-KDD

	KDDTrain+	KDDTest+	KDDTest-21
Normal	67343	9711	2152
DoS	45927	7458	4342
PRB	11656	2421	2402
R2L	995	2754	2754
U2R	52	200	200
Attack	58630	12833	9698
Total	125973	22544	11850

NSL-KDD consists of four main attack types: denial-of-service (DOS), Remote to user (R2L), User to Root attack (U2R), and Probing attack (PRB).

The most critical step in data mining is preprocessing the raw data. This process involves extracting the necessary details from the data. Unfortunately, the data coming from heterogeneous platforms is often noisy, incomplete, and inconsistent. This is why it is essential to transform it into a format that can be used for knowledge discovery. The preprocessing step of this research involves analyzing and transforming the data. Due to the varying requirements of the platforms, the data may contain redundant and anomalous instances. This redundancy can affect the accuracy of classification. Therefore, all the records containing redundant values should be removed from the dataset at the start of the experiments to prevent data duplication.

Regarding symbolic values, for instance, in the NSL-KDD datasets, the feature "protocol type" includes values such as TCP, UDP, and ICMP protocols. Therefore, the conversion process is considered vital in order to improve the accuracy of IDS. In this paper, we replaced each symbolic feature with integer values. In addition, due to the varying scales of features, the classification performance can be affected by size. For instance, features with large numerical values can overwhelm the model's performance compared to features with small numerical values. Accordingly, we took normalization into account in our experiment.

3.2. Reduction of Dimensionality

A feature selection process aims to find a subset of attributes representing the data collected from an intrusion detection dataset [26]. These attributes ensure that the algorithm can interpret the data correctly. Unfortunately, many irrelevant and redundant attributes exist in modern intrusion detection datasets [27]. This study proposes a method that aims to reduce the dimensionality of the data and select the feature subset representative of the data collected from the dataset. It also aims to improve the accuracy of the classification process by implementing the PCA technique. The main idea of this method is to evaluate the relevance of the selected feature subset and the redundancy of the data in the given search space. A PCA is a technique that combines the results of multiple correlated variables into several uncorrelated ones. This method aims to transform these variables into several principal components. The number of principal components derived from the various correlated variables is usually less than or equal to the original number of variables. Therefore, PCA aims to reduce the number of initial variables with significant dimensionality while retaining as much variance as possible. Let us consider a set of connection vectors composed of v_1, v_2, v_3, …, v_M. The following steps are used to calculate the PCAs of a data set:

1. Assume to obtain the entire dataset.
2. For each dimension, calculate the mean vector.
3. For the entire dataset, calculate the covariance matrix.
4. Identify the eigenvectors (e_1, e_2, e_3, ... e_d) and eigenvalues (v_1, v_2, v_3, ... v_d).
5. Select the eigenvector with the highest eigenvalues and sort the eigenvalues in decreasing order.
6. By using this M form, a new sample space can be created.
7. A principal component is determined by the samples obtained.

3.3. Ensemble Classification

 doi:10.20944/preprints202307.0865.v1

The method combines multiple base classifiers for ensemble learning to improve accuracy. This method can solve the same problem and produce much higher prediction results in stability and accuracy. The main reason ensemble classifiers are commonly used is their ability to improve the accuracy and performance of a given project. Another reason ensemble learning is commonly used is insufficient training data. This can lead to a weak or erroneous hypothesis. In this case, the individual classifier will spend significant time developing a reasonable hypothesis.

Voting is the most popular method used in ensemble learning to improve classification performance. It is widely used to build various models. Ensemble methods, such as intrusion detection, can often improve classification accuracy in security applications. Voting is more suitable for heterogeneous learners' ensembles (ET, KNN, RF) with lower computational complexity and less time overhead. ET has been widely used in anomaly detection among decision tree algorithms due to its high efficiency and superficial characteristics. The main advantage of KNN is that it can be applied to various programming problems, such as quadratic programming. This allows the current optimal solution to be continuously renewed. Random forest, on the other hand, is the most representative algorithm used in ensemble learning techniques. It is typically more reliable and capable of achieving better results than individual decision trees. As a result, ET, KNN, and random forest are chosen to build the ensemble for multi-class intrusion detection.

3.3.1. Extra Trees Classifier

The ET classifier aims to provide a prediction and classification framework for analyzing and predicting trees. When growing a tree in a random forest, the features that are considered for splitting are only random. This method can make the trees more random by considering the random thresholds found in each feature. An extremely random forest called an Extra-Trees ensemble is a tree considered for classification and prediction. It also makes the training of Extra-Trees faster since finding the optimal threshold for each feature at each node is very time-consuming. The prediction aims to determine the number of trees in a forest, and the selected features are random. Each tree in the forest represents a different class of prediction. This algorithm performs the random feature selection process on a case-by-case basis [28].

3.3.2. *K* Nearest Neighbors Classifier

The K-NN algorithm is a highly regarded machine learning and data mining algorithm for classification. It is straightforward to implement and is suited to various tasks such as searching. The main reason it is considered one of the most influential classification methods is its ability to use various distance weighting measures. The K-NN algorithm is mainly used for classification as it considers the various elements of a record set. For instance, the distance measures generally use Euclidian distance and the value of K number of neighbors. The type of KNN algorithm that is used for classification. It considers the training data of the various k nearest neighbors and predicts the class value of an unknown record with the help of its nearest neighbors [29, 30]. The distance between the training data (point = x) and the testing data (point = y) is calculated using the Euclidean formula:

$$d(x,y) = \sqrt{\sum_{i=1}^{n} \int ((Xi\colon Yi) - (wi)^2)} \tag{1}$$

where,
x = training data
y = data testing
n = number of attributes
f = similarity function between point x and point y
w_i = weight is given to attribute i

3.3.3. Random Forest (RF) Classifier

Random Forest is a decision tree technique that constructs multiple decision trees that classify thousands of input variables based on their relevance. It can be viewed as an ensemble of classification trees that cast one vote for the class that appears most frequently in the data. Compared to other ML techniques, such as support vector machines or artificial neural networks, RF does not

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 July 2023 doi:10.20944/preprints202307.0865.v1

require many parameters to be specified. In RF, a collection of individual tree-structured classifiers can be defined as:

$$\{h(x, \theta_k), k = 1,2, \ldots i..\} \quad (2)$$

where, h is a RF classifier, $\{\theta k\}$ represents identical random vectors that are independently distributed. Each tree gets a vote for every well-known class at input x. The nature and dimensionality of θ depend on how it is used during the construction of a tree.

The main goal of RF is to create a decision tree representing the forest. This is done by training a subset of the training dataset, around two-thirds. Out Of Bag (OOB) samples are elements employed for inner cross-validation to evaluate the RF's classification accuracy.

Unlike other methods, RF does not require many computational resources to perform its task. It is also insensitive to outliers and parameters; therefore, it is unnecessary to prune the trees, which is a cumbersome task [31].

3.3.4. Voting Algorithm

A voting algorithm is a meta-model that performs the decision process by implementing several classifiers. It considers the factors influencing the decision and applies a combination rule to perform the final step. For instance, the algorithm combination rules are product of probabilities, maximum, minimum and average of probabilities.

Due to the number of classes in a classification, majority voting cannot be performed due to the complexity of the task. This paper introduces the average classifier of probabilities approaches to perform the decision. The average of the predicted probabilities can determine the class label.

Suppose we have l classifiers C, and c classes $\Omega = \{\omega_1, \ldots \ldots, \omega_c\}$ For instance, due to the classifiers considered in our experiment, l can be set to 3, and the value of c depends on the attack types. A classifier C : $R^n \rightarrow$ [0, 1] accepts an object $x \in R^n$ and outputs a vector:

$$\left[p_{c_i}(\omega_1|x), \ldots . p_c(\omega_c|x)\right], \quad (3)$$

where $p_c(\omega_j|x)$ is the probability set by the classifier to determine which object x belongs to a class ω_j. For each class ω_j, let m_i represents the mean of the probabilities assigned by the l classifiers, which can be calculated as:

$$m_i = \frac{1}{l} \sum_{i=1}^{l} p_{c_i}(\omega_j|x) \quad (4)$$

4. Results

The performance of the IDS is evaluated based on its ability to classify network traffic into a specific type. The paper presents the results of the testing process of the proposed algorithm, which was performed by the ensemble. We compare its performance by various metrics, such as Accuracy, Precision, Recall, F-Measure, Performance time, and Error rate. The first step in the PCA process is to identify the PCAs. The proposed PCA method can reduce the dimensionality of the dataset significantly. It also eliminates irrelevant features. An ensemble classifier is also employed to increase the performance of IDS. This method combines three classifiers in a voting algorithm RF, KNN, and ET.

As a result, four separate classifiers were built using the training and testing datasets for classification. Table 2 shows the best classification performance with and without the dimensionality reduction method regarding the main metrics used.

Table 2. (a) The performance outcomes according to original features (41).

Classifier	#PCA	Accuracy	Precision	DR	F-Measure	Budling	Testing	Error rate
Ensemble	*	0.996	0.9996	0.9996	0.9996	36.22	3.9	.004

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 July 2023 doi:10.20944/preprints202307.0865.v1

RF	*	0.997	0.99979	0.9997	0.99979	5.4932	0.61308	.003
KNN	*	0.991	0.99509	0.9950	0.99509	2157.8	231.024	.003
ET	*	0.997	0.998	0.9980	0.99801	0.1132	0.014409	.003

The number of PCAs in Table 2 (b) represents the principal components for each classifier.

We repeatedly ran the experiment to evaluate the performance of the classifiers. Gradually the number of PCAs/features is increased for each classifier. In each iteration, we increased the number of PCAs until adding a new one did not improve the model's performance.

Table 2. (b)The performance outcomes according to the chosen features using PCA.

Classifier	#PCA	Accuracy	Precision	DR	F-Measure	Budling	Testing	Error rate
Ensemble	30	0.998	0.998	.9980	0.9980	19.256	2.238	0.002
RF	29	0.997	0.997	0.9978	0.9978	4.4909	0.518	0.003
KNN	34	0.996	0.996	0.99691	0.99691	14.2809	1.600	0.004
ET	20	0.994	0.994	0.99468	0.9946	0.0617	0.0073	0.006

As shown in Figure 2, the performance of the Ensemble models is not improved after 30 PCAs and 29, 34, and 20 PCAs for RF, KNN, and ET, respectively. The performance of the ensemble method obtains the maximum accuracy rate of 99.89% with 30 PCAs and exceeds all other individual classifiers. In contrast, the best accuracy of the RF, KNN, and ET classifiers were 99.85%, 99.75%, and 99.65% with 29, 34, and 20 PCAs, respectively.

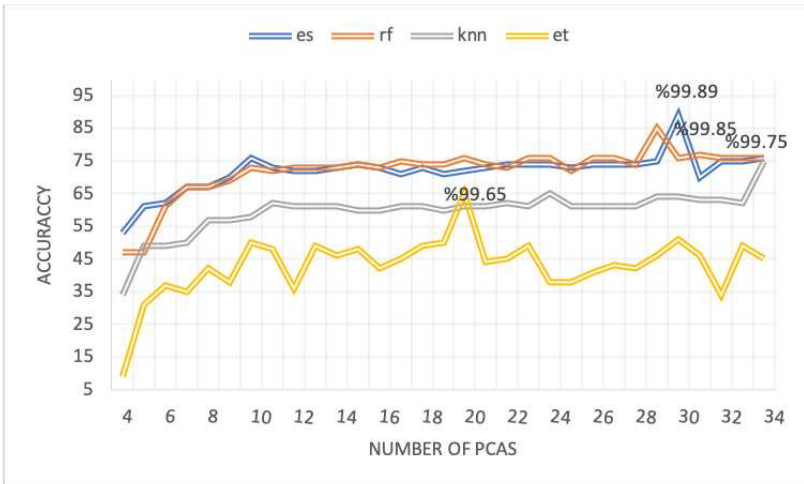

Figure 2. Performance of the ensemble models.

Moreover, the proposed model has the highest scores in DR, precision, and f-measure and the lowest error rate compared to other combined models, as shown in Table 2.

When applied to the ensemble model, the dimensional reduction algorithm significantly reduces the computational cost. Table 3 compares the training and testing times according to the features used. The ensemble model with PCA has reduced the training and testing times compared to the same model using all features. The ensemble model significantly mitigates the training and testing times from 36.22 and 3.58 to 19.25 and 2.238, respectively. It is also noticed that the performance of most classifications is qualified. At the same time, several attacks cannot be classified very well, as seen in Figure 3. The numbers of 'U2R' and 'Heartbleed' are less than others, significantly affecting attack classification. In particular, there are only 52 'U2R' instances in the KDDTrain+ collection, making it difficult for the IDS to be classified correctly.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 July 2023 doi:10.20944/preprints202307.0865.v1

Table 3. The computational time of classifiers

	Without #PCA		**With #PCA**	
	Budling (s)	**Testing(s)**	**Budling (s)**	**Testing(s)**
Ensemble	36.22	3.9	19.256	2.238
RF	5.4932	0.61308	4.4909	0.518
KNN	2157.8	231.024	14.2809	1.600
ET	0.1132	0.014409	0.0617	0.0073

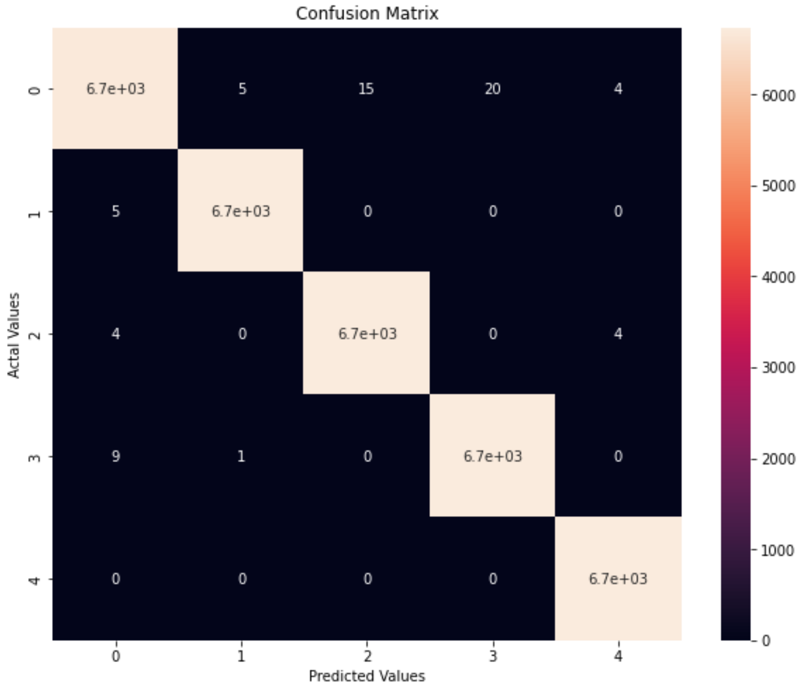

Figure 3. Confusion Matrix of proposed IDS with 30 PCAs.

The proposed ensemble model chooses pertinent features for all classes and does not focus on a specific class. It does not guarantee the effectiveness of all attacks, particularly those with a small number of instances in the datasets. However, the developed model can be employed for detecting intrusions as the classification findings are rather consistent across all datasets. The performance of the proposed IDS is evaluated by comparing it with the proposed PCA method and without feature selection, as shown in Figure 4. The results of the study show that the proposed IDS with PCA outperforms when it comes to distinguishing benign instances from attacks. The average values of various metrics, such as accuracy, precision, and DR have increased significantly.

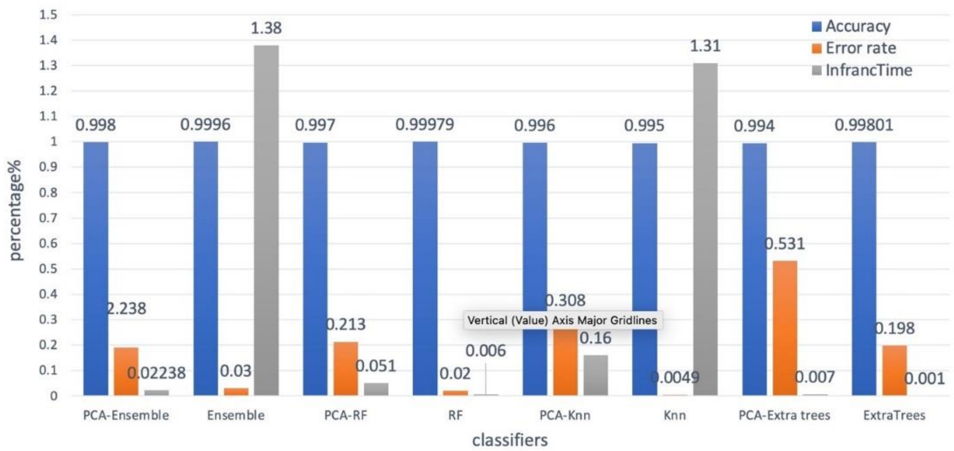

Figure 4. Comparison of performance of classifiers.

5. Conclusions

This paper aims to present an approach to developing an intrusion detection system that can detect multiple types of attacks, using machine learning ensembles to overcome individual classifier weaknesses. We evaluated the system's performance using KDD99 in various network environments. Ensemble intrusion detection uses learning models to detect attacks. The proposed model reduces system dimensionality and eliminates redundancy by employing Principal Component Analysis. Based on the results, it has been found that the stacked ensemble-based model maximizes performance by combining several classifiers that perform better according to the task they are assigned. Overall performance is significantly improved with the proposed model. Due to reduced false positives and increased accuracy, ensemble classifiers are suitable for classifying data in intrusion detection systems with highly imbalanced datasets. In general, the model outperforms expectations, but there are some areas where further improvements can be made, including detecting attacks such as U2R and R2L in KDD'99. A network intrusion detection system with a high-class imbalance is investigated in this paper using an ensemble method. As well as improving class imbalance problems with synthetic oversampling, cost-sensitive learning models can enhance class detection in a few instances.

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

Funding: The authors gratefully acknowledge Qassim University, represented by the Deanship of Scientific Research, on the financial support for this research under the number (COC-2022-1-1-J-25084) during the academic year 1444 AH/2022 AD.

Acknowledgments: The researchers would like to thank the Deanship of Scientific Research, Qassim University for funding the publication of this project.

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

References

1. Du,M.; Wang,K.; Chen,Y.; Wang,X.; Sun,Y. Big Data Privacy Preserving in Multi-Access Edge Computing for Heterogeneous Internet of Things. *IEEE Communications Magazine* **2018**, 56, 62–67. DOI:10.1109/MCOM.2018.1701148
2. Feng, X.; Xiao, Z.; Zhong, B.; Qiu, J.; Dong, Y. Dynamic ensemble classification for credit scoring using soft probability. *Applied Soft Computing* **2018**, 65, 139–151. DOI:10.1016/j.asoc.2018.01.021.
3. Hota, H.; Shrivas, A.K. Decision tree techniques applied on NSL-KDD data and its comparison with various feature selection techniques. *Advanced Computing, Networking, and Informatics* **2014**, 1, 205–211. DOI:10.1007/978-3-319-07353-8_24.

4. Khammassi, C.; Krichen, S. A GA-LR wrapper approach for feature selection in network intrusion detection. *Computers & Security* **2017**, 70, 255–277. DOI:10.1016/j.cose.2017.06.005.
5. Geurts, P.; Ernst, D.; Wehenkel, L. Extremely randomized trees. *Machine learning* **2006**, 63, 3-42. DOI:10.1007/s10994-006-6226-1
6. Cheng, W. X.; Katuwal, R.; Suganthan, P. N.; Qiu, X. A heterogeneous ensemble of trees. In Proceedings of the IEEE Symposium Series on Computational Intelligence (SSCI), Honolulu, HI, USA, 2017, pp. 1-6, DOI: 10.1109/SSCI.2017.8285445.
7. Peterson, L.E. K-nearest neighbor. Scholarpedia 2009, 4, 2, 1883. doi:10.4249/scholarpedia.1883
8. Al-Janabi, M.; Ismail, M.A.; ALI, A.H. Intrusion Detection Systems, Issues, Challenges, and Needs. *Int. J. Comput. Intell. Syst.* **2021**, 14, 1, 560-571. DOI: 10.2991/ijcis.d.210105.001
9. Wang, W.; Li, Y.; Wang, X.; Liu, J.; Zhang, X. Detecting android malicious apps and categorizing benign apps with ensemble of classifiers. *Future Generation Computing System* **2018**, 78, 12, 987–994. DOI: 10.1016/j.future.2017.01.019
10. Ambikavathi, C.; Srivatsa, S.K. Predictor selection and attack classification using random forest for intrusion detection. *Journal of scientific and industrial research* **2020**, 79, 365-68.
11. Tavallaee, M.; Bagheri, E.; Lu, W.; Ghorbani, A. A. A detailed analysis of the KDD CUP 99 data set. In Proceedings of the IEEE Symposium on Computational Intelligence for Security and Defense Applications, Ottawa, ON, Canada, 2009. DOI:10.1109/CISDA.2009.5356528.
12. Alazzam, H.; Sharieh, A.; Sabri, K. A feature selection algorithm for intrusion detection system based on Pigeon Inspired Optimizer. *Expert Systems with Applications* **2020**, 148, 1-14. DOI: 10.1016/j.eswa.2020.113249
13. Kunhare, N.; Tiwari, R.; Dhar, J. Particle swarm optimization and feature selection for intrusion detection system. *Sādhanā* **2020**, 45, 109, 1-14. DOI: 10.1007/s12046-020-1308-5
14. Hosseini, S. "A new machine learning method consisting of GA-LR and ANN for attack detection," *Wirel. Networks* **2020**, 26, 6, 4149–4162. DOI: 10.1007/s11276-020-02321-3.
15. Almasoudy, F. H.; Al-yaseen, W. L.; Idrees, A. K. Differential Evolution Wrapper Feature Selection for Intrusion Detection System. *Procedia Comput. Sci* **2020**. 167, 2019, 1230–1239. DOI: 10.1016/j.procs.2020.03.438
16. Abrar, I.; Ayub, Z.; Masoodi, F.; Bamhdi, A. M. A Machine Learning Approach for Intrusion Detection System on NSL-KDD Dataset. In Proceedings of the International Conference on Smart Electronics and Communication (ICOSEC), Trichy, India, 2020. DOI: 10.1109/ICOSEC49089.2020.9215232.
17. De la Hoz, E.; De La Hoz, E.; Ortiz, A.; Ortega, J.; Prieto, B. PCA filtering and probabilistic SOM for network intrusion detection, *Neurocomputing* **2015**,164, 71–81. DOI: 10.1016/j.neucom.2014.09.083
18. Abu Al-Haija, Q.; Al-Badawi, A. Attack-Aware IoT Network Traffic Routing Leveraging Ensemble Learning. *Sensors* **2022**, 22, 241. DOI: 10.3390/s22010241
19. Rashid, M.M.; Kamruzzaman, J.; Hassan, M.M.; Imam, T.; Gordon, S. Cyberattacks Detection in IoT-Based Smart City Applications Using Machine Learning Techniques. *Int. J. Environ. Res. Public Health* **2020**, 17, 9347. DOI:10.3390/ijerph17249347
20. Zhou, Y.; Cheng, G.; Jiang, S.; Dai, M. Building an efficient intrusion detection system based on feature selection and ensemble classifier. *Computer Networks* **2020**, 174, 107247. DOI: 10.1016/j.comnet.2020.107247.
21. Khare, N.; Devan, P.; Chowdhary, CL; Bhattacharya, S.; Singh, G.; Singh, S.; Yoon, B. SMO-DNN: Spider monkey optimization and deep neural network hybrid classifier model for intrusion detection. *Electronics* **2020**, 9, 692. DOI: 10.3390/electronics9040692.
22. Kasliwal, B.; Bhatia, S.; Saini, S.; Thaseen, I.S.; Kumar, C.A. A hybrid anomaly detection model using G-LDA. In Proceedings of the 2014 IEEE International Advance Computing Conference (IACC), Gurgaon, India, 21–22 February 2014.
23. Elsayed, R.A.; Hamada, R.A.; Abdalla, M.I.; Elsaid S.A. Securing IoT and SDN systems using deep-learning based automatic intrusion detection. *Ain Shams Engineering Journal* **2023**, 14, 102211. DOI: 10.1016/j.asej.2023.102211.
24. Wang, M.; Yang, N.; Weng, N. Securing a Smart Home with a Transformer-Based IoT Intrusion Detection System. *Electronics* **2023**, 12, 2100. DOI: 10.3390/electronics12092100.
25. Bala, R.; Nagpal, R. A review on kdd cup99 and nsl nsl-kdd dataset. *International Journal of Advanced Research in Computer Science* **2019**, 10, 2, 64-67. DOI: 10.26483/ijarcs.v10i2.6395
26. Acharya, N.; Singh, S. An IWD-based feature selection method for intrusion detection system. *Soft Comput* **2018**, 22, 4407–4416. DOI: 10.1007/s00500-017-2635-2.
27. Salo, F.; Nassif, A.B.; Essex, A. Dimensionality reduction with IG-PCA and ensemble classifier for network intrusion detection. *Computer Networks* **2019** 148, 164–175. DOI:10.1016/j.comnet.2018.11.010
28. Obeidat, I.; Hamadneh, N.; Alkasassbeh, M.; Almseidin, M.; AlZubi, M. Intensive Preprocessing of KDD Cup 99 for Network Intrusion Classification Using Machine Learning Techniques. *International Journal of Interactive Mobile Technologies* **2019**, 13, 1, 70-83. DOI: 10.3991/ijim.v13i01.9679
29. Nursalim, S.; Himawan, H. Classification of graduate occupations using the k-nearest neighbor algorithm. *Journal of Information Technology* **2014**, 10, 1, 31–43.

30. Riyadh, M.; Ali, B. J.; Alshibani, D. R. IDS-MIU: An Intrusion Detection System Based on Machine Learning Techniques for Mixed type, Incomplete, and Uncertain Data Set. *International Journal of Intelligent Engineering and Systems* **2021**, 14,3, 493-502 DOI:10.22266/IJIES2021.0630.41
31. Feng, Q.; Liu, J.; Gong, J. UAV remote sensing for urban vegetation mapping using random forest and texture analysis. *Remote sensing* **2015** 7, 1074–1094. DOI:10.3390/rs70101074.