applied sciences

MDPI

Article

A Multi-Stage Classification Approach for IoT Intrusion Detection Based on Clustering with Oversampling

Raneem Qaddoura [1], Ala' M. Al-Zoubi [2,3], Iman Almomani [3,4] and Hossam Faris [3,5,*]

[1] Information Technology, Philadelphia University, Amman 11118, Jordan; rqaddoura@philadelphia.edu.jo
[2] School of Science, Technology and Engineering, University of Granada, 18071 Granada, Spain; alaah14@gmail.com or alzoubi@correo.ugr.es
[3] King Abdullah II School for Information Technology, The University of Jordan, Amman 11942, Jordan; imomani@psu.edu.sa or i.momani@ju.edu.jo
[4] Security Engineering Lab, Computer Science Department, Prince Sultan University, Riyadh 11586, Saudi Arabia
[5] School of Computing and Informatics, Al Hussein Technical University, Amman 11118, Jordan
* Correspondence: hossam.faris@ju.edu.jo or hossam.faris@htu.edu.jo

Abstract: Intrusion detection of IoT-based data is a hot topic and has received a lot of interests from researchers and practitioners since the security of IoT networks is crucial. Both supervised and unsupervised learning methods are used for intrusion detection of IoT networks. This paper proposes an approach of three stages considering a clustering with reduction stage, an oversampling stage, and a classification by a Single Hidden Layer Feed-Forward Neural Network (SLFN) stage. The novelty of the paper resides in the technique of data reduction and data oversampling for generating useful and balanced training data and the hybrid consideration of the unsupervised and supervised methods for detecting the intrusion activities. The experiments were evaluated in terms of accuracy, precision, recall, and G-mean and divided into four steps: measuring the effect of the data reduction with clustering, the evaluation of the framework with basic classifiers, the effect of the oversampling technique, and a comparison with basic classifiers. The results show that SLFN classification technique and the choice of Support Vector Machine and Synthetic Minority Oversampling Technique (SVM-SMOTE) with a ratio of 0.9 and the k value of 3 for k-means++ clustering technique give better results than other values and other classification techniques.

Keywords: intrusion detection; IoT; internet of things; imbalanced; oversampling; IoTID20; clustering

Citation: Qaddoura, R.; Al-Zoubi, A.M.; Almomani, I.; Faris, H. A Multi-Stage Classification Approach for IoT Intrusion Detection Based on Clustering with Oversampling. *Appl. Sci.* **2021**, *11*, 3022. https://doi.org/10.3390/app11073022

Academic Editor: Eui-Nam Huh

Received: 6 March 2021
Accepted: 25 March 2021
Published: 28 March 2021

1. Introduction

The Internet of Things (IoT) can be defined as connected objects (electronic-based) linked through a network to communicate with each other [1]. The concept of IoT is not new; on the contrary, it comes from the late 1990s, when Kevin Ashton, the co-founder of Auto-ID Center at MIT, referred to it in order to describe computers connected to individual objects [2]. These objects mostly consist of computing devices, digital and mechanical machines, or microchips implanted in living creatures that own unique identifiers (UIDs) [1]. However, the actual implementation of IoT took place between 2008 and 2009 according to Cisco Systems [3]. Since then, IoT has been improved and applied in a different number of applications, including motion detection [4], smart home [5], farming [6,7], cities [8,9], connected cars [10], wearable health monitors [11], biometric cybersecurity scanners [12], and logistics tracking [13].

In view of the rapid speed of evolving and existence of IoT devices, it becomes challenging to prevent all security attacks [14]. This occurs due to the fact that IoT devices are created without considering the security and privacy factors [15]. Therefore, such measures cause many vulnerabilities and threats for these devices; for example, in 2019, a team from Cable News Network (CNN) managed to access various types of camera

feeds [16]. They did this using the IoT devices search engine (Shodan), in which several feeds were revealed. One of these feeds showed the daily routine of an Australian family, a man in Moscow getting ready for sleep, and a cat been feeding by her owner in Japan. Furthermore, all cameras did not have any security checks and could be accessed by anyone according to CNN. Another situation happened in 2017, where the US FDA stated that St. Jude Medical's implantable cardiac devices could possibly be hacked [17]. These devices are responsible for monitoring patients' heartbeat and checking for heart attacks. Consequently, hackers could control device functions such as drain the battery, incorrect pacing, and shocks.

The huge data (big data) generated from IoT devices made the detection of such attacks nearly impossible without proper mechanisms. One of these mechanisms is the Intrusion Detection System (IDS), which is a defense system that is responsible for monitoring the activities of the network in IoT devices [18].

There are several intrusion detection approaches, such as preemptive blocking, signature-based, and anomaly detection. In recent years, anomaly-based intrusion detection has shown superior performance over the other approaches, especially using machine learning techniques [19]. The main idea of machine learning-based detection is to train a trustworthy activity model and compare it against the new behavior. Machine learning is known to have better generalization properties than other conventional methods, given the potential to train hardware and application configurations.

Due to the big data generated from IoT devices, machine learning algorithms consider the optimal approach to deal with such data using their ability to deliver meaningful interpretations and predictions as well as deep analysis of the data patterns [20]. The authors of [21] stated that, to develop a computational approach that can detect different types of cyber-attacks, an intelligent data-driven intrusion detection system is required, by means of machine learning technique. Mishra et al. [22] reported that it can be simple to bypass the signature-based intrusion detection system if the the attack is modified slightly, whereas machine learning-based methods can detect these variations as a result of the learning characteristics of the activity of the traffic flow. Further, it can capture the complicated properties of such attacks through learning their behavior and improve the speed as well as the detection performance better than the traditional intrusion detection system.

Well-known datasets have been presented for intrusion detection, namely CICIDS2017 [23], UNSWNB15 [24], and ISCX2012 [25]. However, none of these datasets are collected from an IoT environment. Several works from previous years have started to consider intrusion detection datasets in the IoT environment, such as BoT-IoT [26] and DS20S [27]. Nevertheless, with the growth of IoT devices and novel attack techniques in recent years, it has become a necessity to update and upgrade the datasets to reflect such techniques. Besides, available IoT intrusion datasets lack a large number of features. Thus, recent datasets are introduced such as LITNET-2020 [28] and IoTID20 [29]. The LITNET-2020 dataset was collected from the KTU LITNET network to present the normal and attack network traffic, while the data gathered for IoTID20 were generated from various sources such as smartphones, laptops, tablets, and smart home devices. The IoTID20 focuses more on the daily home usage devices, while the LITNET-2020 concentrates on the academic network traffic. Therefore, in this study, we considered the IoTID20 dataset to investigate the IoT intrusion detection for in-home environments.

Machine learning methods are generally split into two types; supervised and unsupervised learning. The first type is the common one, which is also referred to as classification, where the algorithm learns from the dataset labels; in other words, it has the answer keys of the attack types to evaluate the detection accuracy of the training data. Several works adopted the supervised learning method [30,31]. For example, Alharbi et al. [32] proposed a malware cyberattacks detection system in the IoT environment. Their approach consists of several components, including a traffic analysis unit using the supervised learning method. They applied the decision tree classification algorithm to detect suspicious traffic flow. The proposed approach shows effective detection of malicious attacks with little

bandwidth consumption and a low response time. Verma and Ranga [33] applied the supervised learning method to investigate the detection of DoS attacks in IoT devices. They employed three well-known datasets to evaluate the classification model: NSL-KDD, CIDDS-001, and UNSWNB15. The outcome of the approach showed better performance than the other methods. Supervised learning is useful when attacks are known; however, with the evolution of the attacking techniques alongside emerging new (unknown) ones, it becomes more difficult to detect them.

Another interesting work investigates the webshell detection in IoT environment using the ensemble methods [34]. The authors applied three types of ensemble techniques to enhance the machine learning model's performance: voting, extremely randomized trees (ET), and random forest (RF). Moreover, their outcome of the study is that RF and ET are better for lightweight IoT scenarios, while the voting method was better heavyweight scenarios.

Furthermore, previous works lack two main aspects: the novelty of the IoT intrusion detection datasets and the testing the combined approaches of the learning criteria, supervised and unsupervised. Thus, our approach fills the gap and studies the novelty of the dataset and the use of a combined learning approaches. Unsupervised learning, which is also referred to as clustering, can work alongside with the supervised techniques to handle these unknown novel attacks [22]. Unsupervised learning can be described as a method that extracts and finds hidden patterns of an unlabeled dataset [35]. Consequently, it identifies similar characteristics of current and new IoT attacks and divides them into groups using the clustering machine learning method.

The main contribution of this paper can be summarized as follows:

- We propose a multi-stage approach for classifying the intrusion and normal activities by applying clustering, reduction, oversampling, and classification techniques.
- A unique reduction with clustering technique is applied on the IoT training data to undersample the data while maintaining a representative dataset for training.
- Oversampling the dataset is used to solve the issue of imbalance distribution of the classes in the data.

The remainder of this paper is organized as follows. Section 2 shows recent studies on intrusion detection and recent datasets found in related works. Section 3 discusses the *k*-means clustering technique, the SLFN, and the oversampling techniques, which are used in the proposed approach. The IoTID20 dataset used in the experiments is presented in Section 4. Section 5 discusses in detail the three main stages of the approach, namely data reduction with clustering, oversampling, and classification with SLFN stages. The experiments and results are presented and discussed in Section 6. Finally, Section 7 concludes the work.

2. Related Works

In the past few years, IoT has been utilized in different and essential fields, including healthcare, industry, education, smart cities, agriculture, and retail. Numerous capabilities have led IoT to be applied in various applications [36]. Besides, IoT can help users select the best possible opportunity in any scenario, e.g., cloud resources, management, and decision making [37], thus gaining more attention in academia and industry. For instance, Iqbal et al. [38] proposed an IoT wearable sensor-based device to monitor human health. They used several objects: wearable sensors, activity recognition, and smartphones. Zielonka et al. [39] presented an IoT convection system for small houses. Their approach, which is a remote platform control system, is responsible for collecting sensor readings all over the house from users and optimizing its parameters using computational intelligence to enhance the IoT convection system to improve family comfort. Further, Kamble et al. adopted IoT to recognize the barriers in the retail supply chain for food in India [40]. Abdel-Basset et al. investigated the smart education environment through IoT [41]. Ahmed et al. [42] proposed network architecture for controlling the agricultural places in rural areas.

This increase in IoT utilization and employment makes it hard to control the safety and privacy of connected devices. Thus, the intrusion detection in IoT becomes more crucial. Da Costa et al. [43] suggested that network security is essential due to the high

access to critical information that may cause substantial business losses. Thus, upgrading the field of intrusion detection in IoT is a real necessity [43].

Fu et al. [44] presented an intrusion detection technique in the IoT environment. The approach is used to detect and report different types of attacks, namely reply-attack, jam-attack, and false-attack. Further, they developed experiments to validate the proposed approach against the RADIUS application. Loulianou et al. [45] implemented an intrusion detection system for IoT network protection. The work adopted the signature-based intrusion detection method as well as distributed and centralized modules. Additionally, they designed a DoS scenario using the Cooja simulator and proved that these types of attacks might affect the availability of IoT devices.

On the other hand, applying machine learning for intrusion detection has shown, in many applications, better performance than other approaches. da Costa et al. [43] presented a review of machine learning intrusion detection based in the IoT environment. They surveyed more than 95 papers in the literature that applied machine learning to deal with the issue of IoT intrusion detection. Another recent review presents and analyzes different machine learning and deep learning-based methods in order to identify the intrusion activities of IoT applications [46]. Both reviews emphasize the advantage of machine learning techniques against other approaches in intrusion detection problems.

Most of the works in the literature that employs the machine learning techniques are divided into two groups, namely supervised and unsupervised learning. For example, Smys et al. introduced the supervised learning hybrid convolutional neural networks model for intrusion detection in IoT networks [47]. Their model achieved better results when compared against deep learning and conventional machine learning models. The proposed approach proved that it is sensitive to the detection of IoT attacks. Another supervised learning work developed an attack detection system using the Support Vector Machine (SVM) as a classification model to detect any injected data in the IoT network [48]. The classification model achieved satisfactory results in terms of accuracy. Additionally, Almomani and Alenezi [49] applied eight different data mining techniques to detect and classify different types of DoS (Denial of Service) attacks in the context of sensor-based IoT networks. The dataset used was created by Almomani et al. [50] and includes five types of DoS attacks including flooding, TDMA, grayhole, and blackholes attacks. Although the feature selection algorithm reduced 53% of the overall features, their intrusion detection system attained high accuracy that reached 98%.

On the other hand, the detection of unknown new attacks is better performed by unsupervised learning than supervised learning, because of its ability to group and sort new attacks with similar characteristics. Choudhary and Kesswani [51] presented a cluster-based intrusion detection algorithm for IoT. Their model consists of hybrid intrusion detection for detecting sinkhole and forwarding attacks. The model obtained 96.3% true positive rate and 6.1% false positive rate. Bostani and Sheikhan [52] proposed an un-supervised optimum-path forest algorithm for intrusion detection in IoT. The proposed approach contains two intrusion detection methods, which are specification-based and anomaly-based. The specification-based method analyzes the host nodes and transfers their results to the root node, whereas the anomaly-based method applies the clustering models using the transfer data. Further, through a voting mechanism, the hybrid proposed model identifies the suspicious behavior. The results show that the proposed method acquired 76.19% and 5.92% true positive and false positive rates, respectively.

Notwithstanding, most of the IoT datasets in intrusion detection suffer from the class-imbalance issue that causes poor performance of traditional machine learning approaches. Consequently, some researchers have tried to solve this problem. Telikani and Gandomi suggested a cost-sensitive stacked auto-encoder (CSSAE) approach to handle the imbalance problem in IoT intrusion detection systems [53]. CSSAE produced a cost for each class that depends on the distribution of the classes. Then, an auto-encoder with a two-layer stack was applied to learn the differences between majority and minority classes. Their approach can be employed in both the binary and multi-class data. CSSAE achieved

better results when compared with other intrusion detection systems against KDD CUP 99 and NSL-KDD datasets. Ullah and Mahmoud [54] investigated intrusion detection in IoT networks using the two-level hybrid model to identify the irregular activity. The model utilizes the Synthetic Minority Oversampling Technique (SMOTE) to apply the oversampling technique on CICIDS2017 and UNSW-15 datasets. The experiment results obtained by the model are competitive with 100% for CICIDS2017 and 99% for UNSW-15 in terms of precision, recall, and F-score. Shahriar et al. [55] also addressed the imbalanced issue in IoT intrusion detection systems. They used a generative adversarial network (GAN) as a model to solve the difficulties of imbalanced classes. They argued that their approach performs better in detecting attacks than the standalone intrusion detection systems. Moreover, IoTID20 dataset was tested by Maniriho et al. [56] by classifying three different subsets: normal traffic and DoS attack, normal traffic and MITM, and normal traffic and Scan attack. They did not test the dataset with the classification of normal activities and all the categories of intrusion activities (including Mirai, DoS, MITM, and scan attacks) at once, which is considered in our work.

Therefore, due to the scarcity of research on this matter, we combine both supervised and unsupervised learning methods on the recently published IoTID20 dataset. We solve the imbalanced data issue by the reduction and oversampling techniques, which ensures an expressive and balanced training data, by minifying the training data at one stage using a reduction technique and enlarging it at another stage using oversampling.

3. Preliminaries

This section discusses the introductory information needed for understanding the main components of the proposed approach. It includes a discussion of the *k*-means clustering algorithm, the SLFN algorithm, and the oversampling technique.

3.1. K-Means++ Clustering

K-means++ algorithm, as any other clustering algorithm, finds the relationships between the instances and groups similar ones into the same groups [57]. The *k*-means++ algorithm is one of the popular variations of the *k*-means algorithm, which has a different process for initializing the cluster centers. For the first iteration, the *k*-means++ algorithm chooses the first center randomly but chooses the remaining centers using a weighted probability distribution of the closest centers. Then, the algorithm assigns every other point to the cluster with the closest center to the point. For later iterations, the center of each cluster is calculated and the points are reassigned to the cluster of the closest center. The algorithm stops when the centers are the same for two successive iterations or a predefined number of iterations is reached, which results in k clusters of points [58]. The aim is to minimize the sum of distances between every instance x_j of a cluster s_i and the center c_i for a total of k clusters, which can be represented by the following equation:

$$\underset{s}{argmin} \sum_{i=1}^{k} \sum_{x_j \in s_i} ||x_j - c_i||^2 \quad (1)$$

3.2. Oversampling Techniques

The problem of imbalanced datasets resides in the lack of minority class instances compared to those in the majority class. This problem results in wrong classification of the minority class instances to majority class instances having many false positives [59].

Oversampling techniques can solve this problem by increasing the number of instances for the minority class. Figure 1 shows that the minority class (Class-2) can be enlarged by the oversampling technique to produce a class with a number of instances similar to the other class (Class-1). Oversampling can be done in many ways: randomly copying the minority class instances, synthetically creating new instances of the minority class based

on the features of the instances, or creating new instances of the minority class from the instances that are harder to learn.

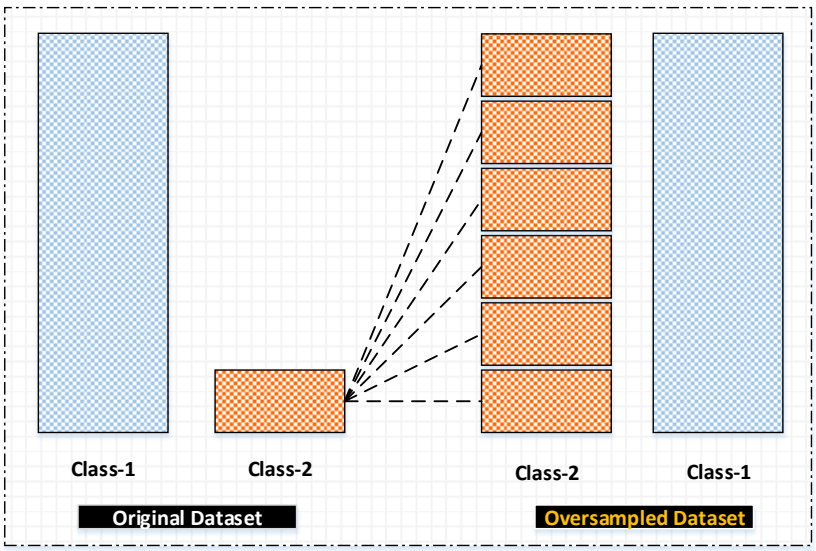

Figure 1. Oversampling technique of the two classes to enlarge the minority class.

3.3. Single Hidden Layer Feed-Forward Neural Network (SLFN)

Figure 2 illustrates the topology of SLFN. The network consists of an input layer which has the values of the features. A hidden layer has neurons with values calculated by an activation function based on the weights between the neurons and the input layer. Finally, the predicted outputs are calculated by the output layer. The aim is to generate a model with output values closer to the target values, which is done by adjusting the weights based on the error value between the output values and the target values.

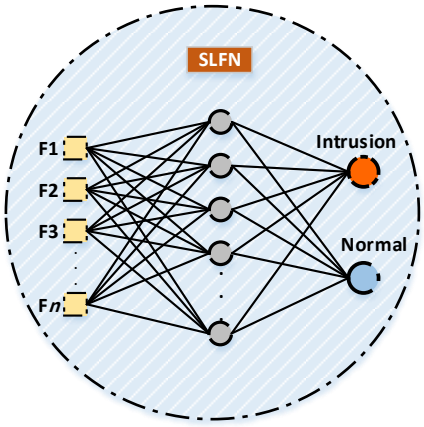

Figure 2. Single Hidden Layer Feed-Forward Neural Network (SLFN).

The task of training the neural network aims at giving high-quality classification results by finding the relationships between the inputs and the outputs because the actual relationship in most cases cannot be recognized by traditional techniques [60]. This is done by tuning the weights and biases of the neural network to give better mapping, and thus increasing the accuracy of the predicted labels, which is measured by a cost function.

For each neuron in the hidden layer, a weighted sum is calculated as the sum of the product between the input values and the corresponding values of the weights, which can be observed by the following equation:

$$WS_j = \sum_{i=1}^{N} w_{ij} F_i + b_J \quad (2)$$

where WS_j is the weighted sum for a neuron j, N is the number of input nodes, F_i is the value of an input node i, w_{ij} is the weight connecting input node i with hidden neuron j, and b_J is the bias of the hidden layer J.

The value of the neuron j is the value of activating the weighted sum WS_j. This value is then considered as input value for the output layer. The weighted sum is then applied on the output layer to generate the predicted value for each neuron in the output layer. The predicted value is then compared to the expected value to find the error generated by the network for each label.

4. IoT Imbalanced Dataset

The IoTID20 dataset [29] contains intrusion and normal activities generated from laptops, tablets, and smartphone devices in a smart home IoT network with a Wi-Fi router connected to SKT NGU device and EZVIZ camera. The dataset includes 80 features and 625,783 instances. The dataset has three labels, the intrusion identification label, the category label, and the sub category label.

The detailed distribution of dataset records among the normal and intrusion activities is shown in Table 1. IoT systems have enlarged the attacks surface by introducing more destructive threats. The main malicious behaviors that were injected and monitored to generate the dataset include Denial of service (DoS), Distributed DoS (DDoS), Man-in-the-Middle (MITM), and active scanning.

Table 1. Dataset records distribution per category/subcategory.

Number of Records per Category and Sub-Category							
Attacks Surface							Normal
DoS		Mirai			MITM	Scan	
Synflooding	Ackflooding	Hostbruteforceg	HTTP Flooding	UDP Flooding	ARP Spoofing	Hostport	OS Port	40,073
59,391	55,124	599,925	55,818	183,554	35,377	22,192	53,073	
Tot: 59,391		*Tot*: 415,677			*Tot*: 35,377	*Tot*: 75,265		

The type of DoS attack considered is the one that usually targets the TCP-based connections (Transmission Control Protocol-based connections) by flooding synchronized (SYN) packets. SYN packets are usually used to build TCP connections between the communicating parties by reserving resources, mainly ports and buffers at both sides. It can be utilized to attack the availability of the server and/or the victim machines. Moreover, DDoS attacks in the context of IoT Mirai were implemented through flooding acknowledgment, HyperText Transfer Protocol (HTTP), and User Datagram Protocol (UDP) packets. In addition, brute force attack was executed to break the encrypted data and expose its secrecy.

MITM was also performed to poison the Address Resolution Protocol (ARP) table and map the Internet Protocol (IP) address of the router with the Media Access Control (MAC) address of the attacker. This allows the attacker to impersonate the network router and interfere with the communications among the network entities. The purpose of this attack is mainly sniffing or manipulating the transmitted data.

In general, before conducting any attack, the scanning phase should take place. This is part of the active reconnaissance that discovers the running services on the victims' machines and the type of operating systems they are using. This can be conducted through port scanning and Time to Live (TTL) analysis. Knowing this information helps the attacker identify the vulnerabilities of these services to attack them successfully and cause severe damage to the IoT system and its resources.

In this study, we considered the intrusion identification label for the IoTID20 dataset indicating identification of the intrusion activities from normal ones. The number of intrusion activities is approximately 15 times the normal ones, having the value of 585,710

for the intrusion label compared to the value of 40,073 for the normal label. Thus, it is an imbalanced dataset.

5. Proposed Approach

In this section, the components of the proposed approach are discussed. Figure 3 illustrates the steps proposed for detecting the intrusion attacks. As observed in the figure, the IoT dataset is split into 2/3 and 1/3 divisions indicating the training and testing portions, which is one of the most common split strategies for classification [61]. The training portion is then considered for the following three main stages of the proposed approach:

- Data reduction with clustering
- Oversampling
- Classification with SLFN

These stages are further discussed in the following sections. Finally, the generated model is evaluated using the testing data portion.

The proposed approach is represented by Algorithm 1. The algorithm accepts the dataset and several other values including the number of clusters (k), the oversampling ratio, and the reduction percentage. The training and testing split is presented by Line 1. The three stages reduction with clustering, oversampling, and classification are presented on Lines 2–4, 5, and 6, respectively. Then, the testing portion is predicted on Line 7 by the model generated from the classification stage. Finally, the evaluation process is performed on Line 8 generating the Accuracy (ACC), Precision (PREC), Recall (REC), and G-mean (GM).

Algorithm 1: SLFN-SVM-SMOTE

Input: dataset, k, ratio, reduction%
Output: ACC, PREC, REC, GM
1 train, test = split(dataset)
2 clusters = k-means++(train, k)
3 updated-clusters = reduce(clusters, reduction%)
4 reduced-dataset = aggregate(updated-clusters)
5 oversampled-dataset = SVM-SMOTE(reduced-dataset, ratio)
6 model = SLFN(oversampled-dataset)
7 predicted-labels = predict(model, test)
8 ACC, PREC, REC, GM = evaluate(predicted-labels)

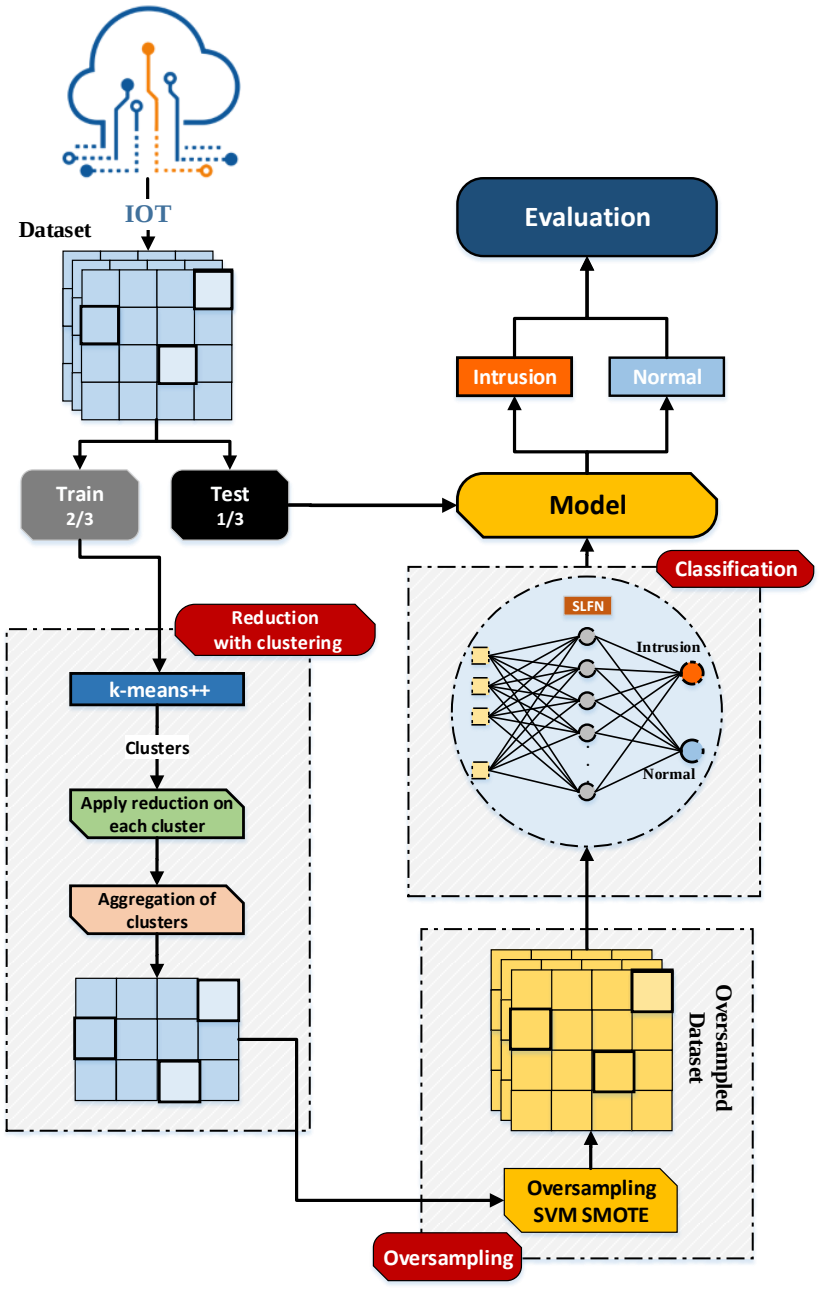

Figure 3. The proposed SLFN-SVM-SMOTE technique including the three stages reduction with clustering, oversampling, and classification.

5.1. Data Reduction with Clustering

Data of similar characteristics can be grouped using a clustering technique to handle each group of instances in a similar way. The aim is to provide a mechanism to maintain a useful dataset but with reduced number of instances.

Figure 4 shows the steps of applying the data reduction with clustering stage. First, the 2/3 split of data forming the training data is clustered using the *k*-means++ clustering algorithm to produce a set of clusters. Second, each cluster is reduced by 10% to form an updated set of clusters, which are then aggregated to form a reduced but comprehensive dataset in the last step.

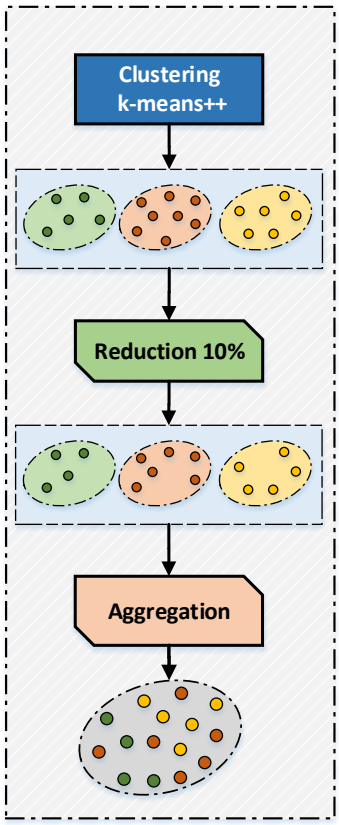

Figure 4. Data reduction with clustering stage using *k*-means++, a reduction by 10%, and the aggregation step.

The updated dataset is then passed to the oversampling stage discussed in the following section. This provides another benefit of the reduction process as it is later enlarged at the oversampling stage, which results in minimizing the processing volume of the computation while maintaining a useful and comprehensive dataset.

5.2. Oversampling

Unbalanced datasets suffer from poor predicting performance by a classification algorithm. The classification algorithm generates either an over-fitted model or a model which is bias toward the majority class. Oversampling techniques are used to solve this shortcoming, as discussed in Section 3.2. The SVM-SMOTE oversampling technique is chosen to oversample the reduced dataset into an enlarged one, as observed in Figure 5. The enlarged dataset is then passed to the classification stage discussed in the following section.

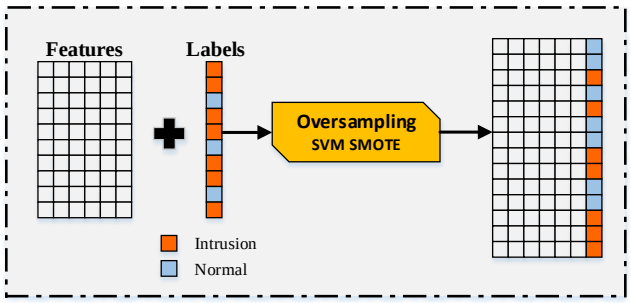

Figure 5. The SVM-SMOTE oversampling technique to increase the number of instances of the normal class.

5.3. Classification with SLFN

The enlarged dataset is then classified using the SLFN to generate the classification model. The testing portion of the data is considered as an input for the model, which produces classified instances into intrusion and normal activities. Then, it is evaluated by the evaluation techniques to assess the quality of the classification model.

6. Experiments and Results

This section presents the environmental settings, the description of the evaluation measures, and the evaluation of the framework. The evaluation of the framework was done by the following stages:

- Select the best classification algorithm for the proposed framework with the best value of k for the k-means++ clustering technique.
- Select the best oversampling technique for the proposed framework with the best oversampling ratio.
- Compare the best selection of the classifier, oversampling technique, k value, and oversampling ratio with the other basic classifiers, including SVM, Stochastic Gradient Descent (SGD), Logistic Regression (LR), and SLFN.

6.1. Environmental Settings

A personal computer with Intel core i7-1065G7 CPU and 1.30GHz/16 GB RAM was used for running the experiments. The imbalanced-learn [62] and Scikit Learn [63] Python libraries with Python 3.8 were used to run the k-means++, SLFN, SVM, SGD, LR, SMOTE, Adaptive Synthetic (ADASYN) sampling approach, SVM-SMOTE, Borderline1-SMOTE, and Borderline2-SMOTE techniques.

The k values of 2, 3, and 4 for the k-means++ clustering algorithm and the oversampling ratio values of 0.1, 0,2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, and 1 for the oversampling technique were considered. The SLFN, SVM, SGD, LR and NB classifiers were used for comparison with the proposed framework. Finally, the intrusion identification label of the dataset was considered for the proposed framework.

6.2. Evaluation Measures

The framework was evaluated using the accuracy, precision, recall, and G-mean measures. The Accuracy (ACC), Precision of the intrusion class ($PREC_I$), Precision of the normal class ($PREC_N$), Recall of the intrusion class (REC_I), Recall of the normal class (REC_N), and G-mean (GM) measures are described by Equations (3)–(8), respectively:

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

$$PREC_I = \frac{TP}{TP + FP} \tag{4}$$

$$PREC_N = \frac{TN}{TN + FN} \tag{5}$$

$$REC_I = \frac{TP}{TP + FN} \tag{6}$$

$$REC_N = \frac{TN}{TN + FP} \tag{7}$$

$$GM = \sqrt{REC_I * REC_N} \tag{8}$$

where TP is the true positive indicating the number of intrusion instances which are predicted as intrusion, TN is the true negative indicating the number of normal instances which are predicted as normal, FP is the false positive indicating the number of normal instances which are predicted as intrusion, and FN is the false negative indicating the number of intrusion instances which are predicted as normal.

6.3. Effect of the Data Reduction with Clustering

The distribution of the intrusion and normal activity instances for each cluster is presented by Figure 6. As observed in the figure, the training dataset is clustered into three clusters. The first cluster contains a majority of the instances as intrusion which dominates the normal ones. The same observation is concluded for the other clusters having a domination of the intrusion instances for the second cluster and an absence of the normal instances for the third cluster.

In contrast, the distribution of the intrusion and normal activities for each cluster after the reduction process is presented in Figure 7. As intended, to provide representative dataset, a similar distribution can be observed in Figure 7 compared to Figure 6. The only difference is the number of instances, which are reduced by 10% for each cluster, as discussed in Section 5.1. Note that the reduction is not performed equally for each label of the same cluster but rather performed at the cluster level, producing a reduced number of cluster instances.

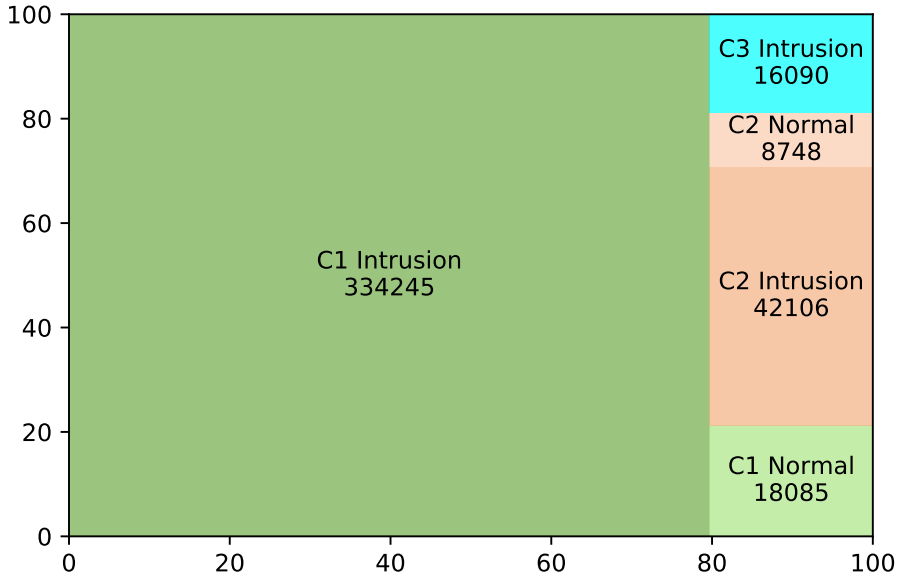

Figure 6. Distribution of instances before reduction.

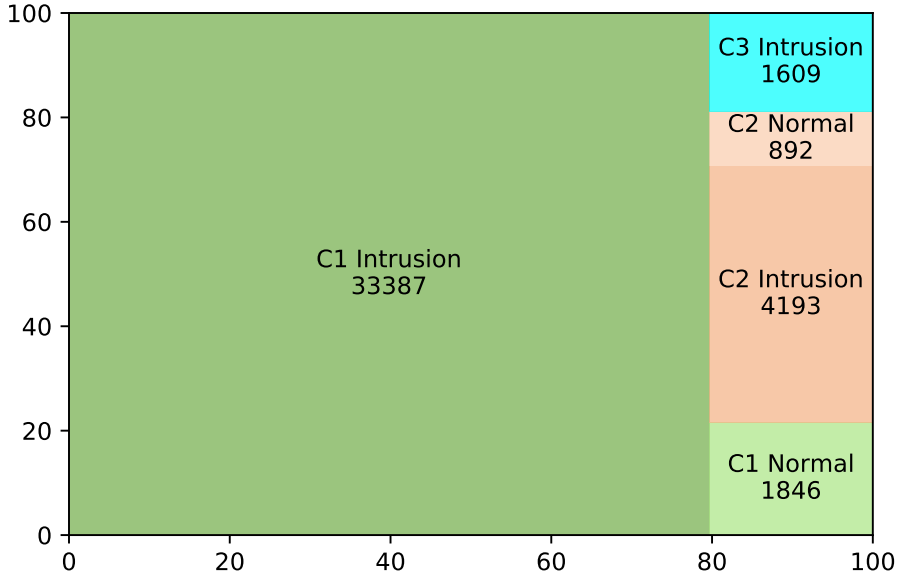

Figure 7. Distribution of instances after reduction.

6.4. Evaluation of the Framework with Basic Classifiers

Different classifiers were tested with the *k*-means having different values of *k*. The values of 2, 3, and 4 were considered for *k* and the SVM, SGD, LR, and SLFN were considered as a classifier. Table 2 shows the ACC, PREC, REC, and GM values for different classifiers and different values of *k*. We ignore the results achieved by the NB classifier because the high values of the $PREC_N$ and REC_I do not reflect high-quality prediction as the majority of the normal activity instances are classified as intrusion having very low values of REC_N and GM. Thus, by ignoring these values, the maximum values from the other techniques are determined in bold. Considering this, SLFN classifier outperformed the other classifiers for the IoTID20 dataset and thus was considered for the next stages of the evaluation of the framework. In addition, the $PREC_N$, REC_I, and GM values for SLFN for the *k* value of 3 outperformed the other values of *k*. With imbalanced datasets, GM measure ensures that the lack of instances for the minority class does not affect the quality of the results generated by the classifier. In addition, predicting the largest amount of intrusion from the instances that are labeled as intrusion, which is reflected by the REC_I measure, is critical. Thus, the SLFN classifier with the *k* value of 3 was considered for the framework for the IoTID20 dataset.

Table 2. Evaluation of the framework with basic classifiers including SVM, SGD, LR, and SLFN with the values of 2, 3, and 4 for *k*.

Technique	ACC	$PREC_I$	$PREC_N$	REC_I	REC_N	GM
SVM-*k*-means++$_{k=2}$	0.9709	0.9220	0.9731	0.5974	0.9965	0.77156
SVM-*k*-means++$_{k=3}$	0.9708	0.9177	0.9731	0.5976	0.9963	0.77161
SVM-*k*-means++$_{k=4}$	0.9700	0.9004	0.9732	0.5991	0.9955	0.7723
SGD-*k*-means++$_{k=2}$	0.9589	0.7692	0.9674	0.5133	0.9894	0.7127
SGD-*k*-means++$_{k=3}$	0.9657	0.8758	0.9694	0.5422	0.9947	0.7344
SGD-*k*-means++$_{k=4}$	0.9686	0.9332	0.9700	0.5498	0.9973	0.7405
LR-*k*-means++$_{k=2}$	0.9587	0.8457	0.9626	0.4359	0.9946	0.6584
LR-*k*-means++$_{k=3}$	0.9574	0.8083	0.9628	0.4403	0.9928	0.6612
LR-*k*-means++$_{k=4}$	0.9570	0.7905	0.9633	0.4491	0.9918	0.6674
SLFN-*k*-means++$_{k=2}$	0.9703	0.8435	0.9770	0.6591	0.9916	0.8084
SLFN-*k*-means++$_{k=3}$	0.9776	0.8216	**0.9885**	**0.8318**	0.9876	**0.9064**
SLFN-*k*-means++$_{k=4}$	**0.9844**	**0.9647**	0.9855	0.7854	**0.9980**	0.8854
NB-*k*-means++$_{k=2}$	0.3199	0.0862	1.0000	1.0000	0.2734	0.5228
NB-*k*-means++$_{k=3}$	0.3198	0.0861	1.0000	0.9999	0.2732	0.5227
NB-*k*-means++$_{k=4}$	0.3184	0.0860	1.0000	0.9999	0.2717	0.5212

6.5. Effect of the Oversampling Techniques on the Framework

Different oversampling techniques were tested with different oversampling ratio values for the SLFN classification algorithm with *k* value of 3. The SMOTE, ADASYN, SVM-SMOTE, Borderline1-SMOTE, and Borderline2-SMOTE oversampling techniques with the oversampling ratio values of 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, and 1 were considered. Figure 8 shows the performance of the different oversampling techniques and the different values of the oversampling ratio in terms of GM. The oversampling ratio for each oversampling technique having the best GM value was considered for the next stage of comparisons. Based on the figure, the oversampling ratios of 0.9, 0.6, 0.9, 0.6, and 0.5 were considered for the SMOTE, ADASYN, SVM-SMOTE, Borderline1-SMOTE, and Borderline2-SMOTE oversampling techniques, respectively.

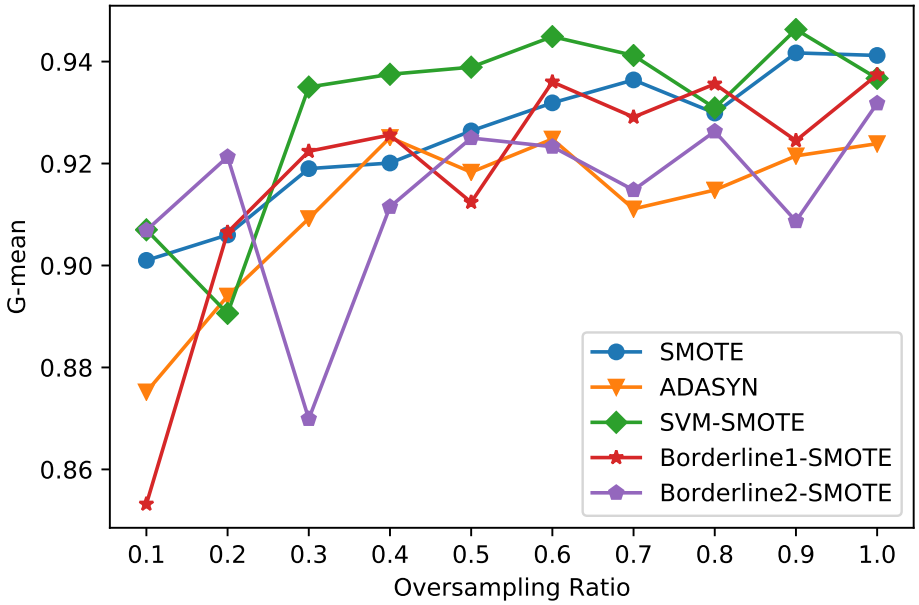

Figure 8. GM evaluation of different values of oversampling ratios and different oversampling techniques (SMOTE, ADASYN, SVM-SMOTE, Borderline1-SMOTE, and Borderline2-SMOTE).

Table 3 is a performance comparison of different oversampling techniques for the selected values of the oversampling ratio. The average and standard deviation values of ACC, PREC, REC, and GM for 10 runs are presented in the table in the form of [value ± std]. The maximum values are determined in bold. As shown in the table, the best values for $PREC_N$, REC_I, and GM were observed for the SVM-SMOTE oversampling technique. Although SMOTE oversampling technique has good results for ACC, $PREC_I$, and REC_N, as discussed above, GM is the most important measure for imbalanced datasets. Further, Figure 9 illustrates the values of the GM measure for different oversampling techniques for 10 runs. It shows the minimum, maximum, mean, and standard deviation values for each technique for the best oversampling ratio value. The figure shows that the SVM-SMOTE has better values than other techniques; it has the best maximum, minimum, and mean values as well as the best standard deviation as the box values have the minimum height compared to the others. This indicates the stable and exceptional performance of SVM-SMOTE compared to the other oversampling techniques. Thus, based on the observations presented in Table 3 and Figure 9, the SVM-SMOTE oversampling technique was considered as the best oversampling technique for the framework for the IoTID20 dataset.

Table 3. Evaluation of SLFN-*k*-means++$_{k=3}$ with different oversampling techniques.

Oversampling	Ratio	ACC	$PREC_I$	$PREC_N$	REC_I	REC_N	GM
SMOTE	0.9	**0.9481** ± 0.0330	**0.6055** ± 0.1389	0.9936 ± 0.0027	0.9098 ± 0.0408	**0.9508** ± 0.0372	0.9293 ± 0.0158
ADASYN	0.6	0.8942 ± 0.0601	0.4211 ± 0.1318	0.9958 ± 0.0024	0.9433 ± 0.0359	0.8909 ± 0.0660	0.9154 ± 0.0262
SVM-SMOTE	0.9	0.9351 ± 0.0287	0.5211 ± 0.1139	**0.9969** ± 0.0028	**0.9578** ± 0.0410	0.9335 ± 0.0335	**0.9453** ± 0.0183
Borderline1-SMOTE	0.6	0.9222 ± 0.0376	0.4915 ± 0.1355	0.9952 ± 0.0042	0.9331 ± 0.0623	0.9214 ± 0.0427	0.9260 ± 0.0264
Borderline2-SMOTE	0.5	0.9023 ± 0.0433	0.4275 ± 0.1276	0.9958 ± 0.0022	0.9433 ± 0.0323	0.8994 ± 0.0479	0.9203 ± 0.0175

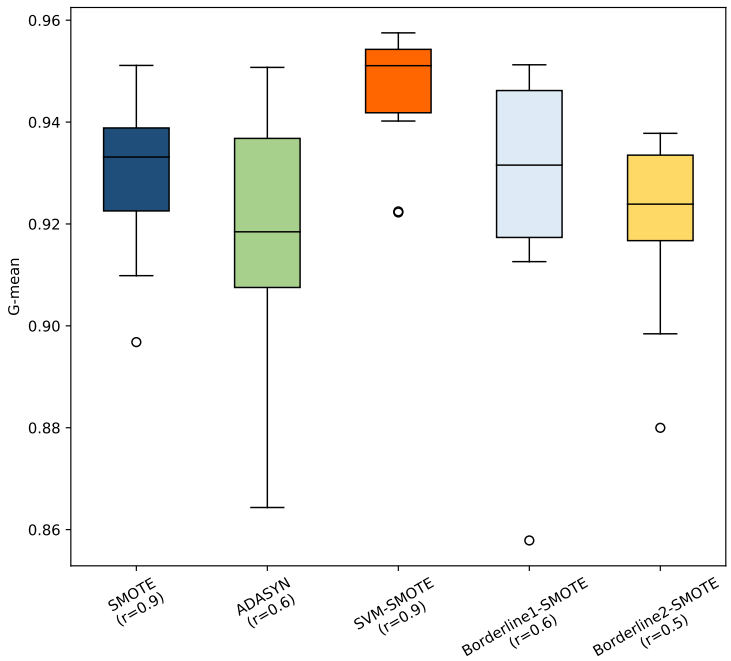

Figure 9. The box plot for the GM value for different oversampling techniques.

6.6. Comparison with Basic Classifiers

Based on the analysis of the previous experiments, the SLFN classifier with *k* value of 3 and SVM-SMOTE oversampling technique with 0.9 oversampling ratio were considered for evaluating the proposed framework, by comparing the results with SVM, SGD, LR, SLFN, and NB. Table 4 shows the values of ACC, PREC, REC and GM for SLFN-SVM-SMOTE $_{(r = 0.9, k = 3)}$, SVM, SGD, LR, SLFN, and NB. Ignoring the results achieved by NB, as discussed in Section 6.4, the maximum values from the other techniques are determined in bold. Thus, the table shows that the $PREC_N$, REC_I, and GM values of the proposed approach are larger than those of the other classifiers. Consequently, the GM value indicates an advanced performance for the proposed framework.

Table 4. Comparison of SLFN-SVM-SMOTE $_{(r = 0.9, k = 3)}$ with basic classifiers.

Technique	ACC	$PREC_I$	$PREC_N$	REC_I	REC_N	GM
SLFN-SVM-SMOTE $_{(r = 0.9, k = 3)}$	0.9351	0.5211	**0.9969**	**0.9578**	0.9335	**0.9453**
SVM	0.9713	0.9179	0.9737	0.6066	0.9963	0.7774
SGD	0.9668	0.8760	0.9707	0.5619	0.9946	0.7475
LR	0.9572	0.8094	0.9625	0.4350	0.9930	0.6573
SLFN	**0.9842**	**0.9879**	0.9841	0.7637	**0.9994**	0.8736
NB	0.3199	0.0862	1.0000	1.0000	0.2733	0.5228

6.7. Discussion

In summary, reduction with clustering, oversampling, and classification stages were tested for the IoTID20 dataset with a selected values of oversampling ratio and *k* for *k*-means clustering technique using SLFN classifier and SVM-SMOTE oversampling technique.

The reduction with clustering stage produced an undersampled but representative dataset having the same distribution of normal and intrusion activities. The SLFN classification technique and *k*-means clustering with *k* value of 3 generated better performance than other classification techniques and other values of *k* by having the highest value of GM. In addition, the performance of the SVM-SMOTE with a value of 0.9 for the over-

sampling ratio gave better results than other oversampling techniques and other values of oversampling ratio. Finally, the proposed approach outperformed the other basic classifiers in terms of GM.

The experiments were limited to the classification of the intrusion identification label, which could extended to include the category label and the subcategory label for the IoTID20 dataset. It also considered the specific distribution of the activities for the IoTID20 dataset and could be tested on different datasets having different distribution of activities. Other works in the future are expected to be observed for the IoTID20 dataset where a comparison and empirical evaluation can hold. In addition, an issue might arise if there is a significant difference between the sizes of the resulted clusters in the first stage (i.e., some clusters are very small compared to other clusters). In this case, the pattern of such instances will be underrepresented for the training algorithm and consequently lead to misclassification of similar instances.

7. Conclusions and Future Work

This paper proposes an intrusion detection approach for a recent IoT dataset named IoTID20. The proposed approach and the value settings can be summarized as follows:

- *k*-means clustering of the training data to three clusters
- Clusters reduction by 10% and then aggregation of the three reduced clusters
- Oversampling the aggregated data into an enlarged one using the SVM-SMOTE oversampling technique with an oversampling ratio value of 0.9
- Generating the classification model of the oversampled data by SLFN classification technique
- Evaluating the model using the testing data in terms of ACC, PREC, REC, and GM

The effects of the data reduction with clustering, different oversampling techniques, and the selection of the classification technique were tested by the proposed technique. The aim of the reduction with clustering stage is to provide a mechanism to maintain a representative dataset but with reduced number of instances while minimizing the processing volume of the computation. On the other hand, the aim of the oversampling stage is to solve the problem of imbalanced dataset. Finally, the classification stage generates the classification model to classify instances into intrusion and normal activities. The results show that the combination of clustering the dataset with *k*-means++ into three clusters with a reduction by 10% and an oversampling by a ratio value of 0.9 with SVM-SMOTE technique is the best approach for producing high-quality results, and this approach outperforms the other approaches on the selected IoTID20 dataset.

For future work, the other two labels presented by the dataset can be considered to detect the intrusion activity type and the sub type. Other unsupervised and supervised learning approaches can also be considered. In addition, different IoT intrusion detection datasets can be tested by the proposed approach.

Author Contributions: Conceptualization, H.F. and R.Q.; Methodology, H.F. and R.Q.; Validation, R.Q. and A.M.A.-Z.; Data curation, R.Q.; Writing—original draft preparation, R.Q., A.M.A.-Z., and I.A.; Writing—review and editing, R.Q., A.M.A.-Z., H.F., and I.A.; Supervision H.F.; and Project administration, I.A. and H.F. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding. The APC is funded by Prince Sultan University.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data available in a publicly accessible repository at https://sites.google.com/view/iot-network-intrusion-dataset/home (accessed on 26 March 2021). The data presented in this study are openly available at https://doi.org/10.1007/978-3-030-47358-7_52 (accessed on 26 March 2021) [29].

Acknowledgments: The authors would like to acknowledge the support of Prince Sultan University for paying the Article Processing Charges (APC) of this publication.

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

References

1. Qadri, Y.A.; Nauman, A.; Zikria, Y.B.; Vasilakos, A.V.; Kim, S.W. The Future of Healthcare Internet of Things: A Survey of Emerging Technologies. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 1121–1167. [CrossRef]
2. Ashton, K. That 'internet of things' thing. *RFID J.* **2009**, *22*, 97–114.
3. Evans, D. The internet of things: How the next evolution of the internet is changing everything. *CISCO White Pap.* **2011**, *1*, 1–11.
4. Balogh, Z.; Magdin, M.; Molnár, G. Motion Detection and Face Recognition using Raspberry Pi, as a Part of, the Internet of Things. *Acta Polytech. Hung.* **2019**, *16*, 167–185.
5. AbuNaser, M.; Alkhatib, A.A. Advanced survey of blockchain for the internet of things smart home. In Proceedings of the 2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT), Amman, Jordan, 9–11 April 2019; pp. 58–62.
6. Ronaghi, M.H.; Forouharfar, A. A contextualized study of the usage of the Internet of things (IoTs) in smart farming in a typical Middle Eastern country within the context of Unified Theory of Acceptance and Use of Technology model (UTAUT). *Technol. Soc.* **2020**, *63*, 101415. [CrossRef]
7. Casta neda-Miranda, A.; Casta no-Meneses, V.M. Internet of things for smart farming and frost intelligent control in greenhouses. *Comput. Electron. Agric.* **2020**, *176*, 105614. [CrossRef]
8. Sadiq, A.S.; Faris, H.; Ala'M, A.Z.; Mirjalili, S.; Ghafoor, K.Z. Fraud detection model based on multi-verse features extraction approach for smart city applications. In *Smart Cities Cybersecurity and Privacy*; Elsevier: Amsterdam, The Netherlands, 2019; pp. 241–251.
9. Vinayakumar, R.; Alazab, M.; Srinivasan, S.; Pham, Q.V.; Padannayil, S.K.; Simran, K. A visualized botnet detection system based deep learning for the Internet of Things networks of smart cities. *IEEE Trans. Ind. Appl.* **2020**, *56*, 4436–4456. [CrossRef]
10. Gupta, M.; Sandhu, R. Authorization framework for secure cloud assisted connected cars and vehicular internet of things. In Proceedings of the 23nd ACM on Symposium on Access Control Models and Technologies, Indianapolis, IN, USA, 13–15 June 2018; pp. 193–204.
11. Talboom, J.S.; Huentelman, M.J. Big data collision: the internet of things, wearable devices and genomics in the study of neurological traits and disease. *Hum. Mol. Genet.* **2018**, *27*, R35–R39. [CrossRef]
12. Hamidi, H. An approach to develop the smart health using Internet of Things and authentication based on biometric technology. *Future Gener. Comput. Syst.* **2019**, *91*, 434–449. [CrossRef]
13. Laxmi, A.R.; Mishra, A. RFID based logistic management system using internet of things (IoT). In Proceedings of the 2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, 29–31 March 2018; pp. 556–559.
14. Williams, R.; McMahon, E.; Samtani, S.; Patton, M.; Chen, H. Identifying vulnerabilities of consumer Internet of Things (IoT) devices: A scalable approach. In Proceedings of the 2017 IEEE International Conference on Intelligence and Security Informatics (ISI), Beijing, China, 22–24 July 2017; pp. 179–181.
15. Thamilarasu, G.; Chawla, S. Towards deep-learning-driven intrusion detection for the internet of things. *Sensors* **2019**, *19*, 1977. [CrossRef]
16. Griffiths, J. 'Internet of Things' or 'Vulnerability of Everything'? Japan Will Hack Its Own Citizens to Find Out. 2019. Available online: http://epicenterla.org/amp/2019/02/03/cnn-internet-of-things-or-vulnerability-of-everything-japan-will-hack-its-own-citizens-to-find-out/ (accessed on 26 March 2021)
17. Larson, S. FDA Confirms that St. Jude's Cardiac Devices Can be Hacked. 2017. Available online: https://www.fox61.com/article/news/local/outreach/awareness-months/fda-confirms-that-st-judes-cardiac-devices-can-be-hacked/520-9a16749b-751c-4132-b019-b87959c128aa (accessed on on 26 March 2021)
18. Kumar, C.S. Correlating Internet of Things. *Int. J. Manag. (IJM)* **2017**, *8*, 68–76.
19. Aljawarneh, S.; Aldwairi, M.; Yassein, M.B. Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model. *J. Comput. Sci.* **2018**, *25*, 152–160. [CrossRef]
20. Sarker, I.H.; Kayes, A.; Badsha, S.; Alqahtani, H.; Watters, P.; Ng, A. Cybersecurity data science: an overview from machine learning perspective. *J. Big Data* **2020**, *7*, 1–29. [CrossRef]
21. Alqahtani, H.; Sarker, I.H.; Kalim, A.; Hossain, S.M.M.; Ikhlaq, S.; Hossain, S. Cyber Intrusion Detection Using Machine Learning Classification Techniques. In Proceedings of the International Conference on Computing Science, Communication and Security, Gujarat, India, 26–27 March 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 121–131.
22. Mishra, P.; Varadharajan, V.; Tupakula, U.; Pilli, E.S. A detailed investigation and analysis of using machine learning techniques for intrusion detection. *IEEE Commun. Surv. Tutor.* **2018**, *21*, 686–728. [CrossRef]
23. Sharafaldin, I.; Lashkari, A.H.; Ghorbani, A.A. Toward generating a new intrusion detection dataset and intrusion traffic characterization. In Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP 2018), Funchal, Portugal, 22–24 January 2018; pp. 108–116.

24. Moustafa, N.; Slay, J. UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In Proceedings of the 2015 Military Communications and Information Systems Conference (MilCIS), Canberra, Australia, 10–12 November 2015; pp. 1–6.
25. Shiravi, A.; Shiravi, H.; Tavallaee, M.; Ghorbani, A.A. Toward developing a systematic approach to generate benchmark datasets for intrusion detection. *Comput. Secur.* **2012**, *31*, 357–374. [CrossRef]
26. 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]
27. Pahl, M.O.; Aubet, F.X. All eyes on you: Distributed Multi-Dimensional IoT microservice anomaly detection. In Proceedings of the 2018 14th International Conference on Network and Service Management (CNSM), Rome, Italy, 5–9 November 2018; pp. 72–80.
28. Damasevicius, R.; Venckauskas, A.; Grigaliunas, S.; Toldinas, J.; Morkevicius, N.; Aleliunas, T.; Smuikys, P. LITNET-2020: An annotated real-world network flow dataset for network intrusion detection. *Electronics* **2020**, *9*, 800. [CrossRef]
29. Ullah, I.; Mahmoud, Q.H. A Scheme for Generating a Dataset for Anomalous Activity Detection in IoT Networks. In Proceedings of the Canadian Conference on Artificial Intelligence, online, 13–15 May 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 508–520.
30. Liu, J.; Kantarci, B.; Adams, C. Machine learning-driven intrusion detection for contiki-NG-based IoT networks exposed to NSL-KDD dataset. In Proceedings of the 2nd ACM Workshop on Wireless Security and Machine Learning, Miami, FL, USA, 28 June–2 July 2020; pp. 25–30.
31. Hindy, H.; Bayne, E.; Bures, M.; Atkinson, R.; Tachtatzis, C.; Bellekens, X. Machine Learning Based IoT Intrusion Detection System: An MQTT Case Study. *arXiv* **2020**, arXiv:2006.15340.
32. Alharbi, S.; Rodriguez, P.; Maharaja, R.; Iyer, P.; Bose, N.; Ye, Z. FOCUS: A fog computing-based security system for the Internet of Things. In Proceedings of the 2018 15th IEEE Annual Consumer Communications & Networking Conference (CCNC), Las Vegas, NV, USA, 12–15 January 2018; pp. 1–5.
33. Verma, A.; Ranga, V. Machine learning based intrusion detection systems for IoT applications. *Wirel. Pers. Commun.* **2019**, *111*, 2287–2310. [CrossRef]
34. Yong, B.; Wei, W.; Li, K.C.; Shen, J.; Zhou, Q.; Wozniak, M.; Połap, D.; Damaševičius, R. Ensemble machine learning approaches for webshell detection in Internet of things environments. *Trans. Emerg. Telecommun. Technol.* **2020**, *2020*, e4085. [CrossRef]
35. Qaddoura, R.; Faris, H.; Aljarah, I. An efficient evolutionary algorithm with a nearest neighbor search technique for clustering analysis. *J. Ambient. Intell. Humaniz. Comput.* **2020**, 1–26.
36. Talavera, J.M.; Tobón, L.E.; Gómez, J.A.; Culman, M.A.; Aranda, J.M.; Parra, D.T.; Quiroz, L.A.; Hoyos, A.; Garreta, L.E. Review of IoT applications in agro-industrial and environmental fields. *Comput. Electron. Agric.* **2017**, *142*, 283–297. [CrossRef]
37. Asghari, P.; Rahmani, A.M.; Javadi, H.H.S. Internet of Things applications: A systematic review. *Comput. Netw.* **2019**, *148*, 241–261. [CrossRef]
38. Iqbal, A.; Ullah, F.; Anwar, H.; Ur Rehman, A.; Shah, K.; Baig, A.; Ali, S.; Yoo, S.; Kwak, K.S. Wearable Internet-of-Things platform for human activity recognition and health care. *Int. J. Distrib. Sens. Netw.* **2020**, *16*, 1550147720911561. [CrossRef]
39. Zielonka, A.; Sikora, A.; Woźniak, M.; Wei, W.; Ke, Q.; Bai, Z. Intelligent Internet-of-Things system for smart home optimal convection. *IEEE Trans. Ind. Inform.* **2020**, *17*, 4308–4317. [CrossRef]
40. Kamble, S.S.; Gunasekaran, A.; Parekh, H.; Joshi, S. Modeling the internet of things adoption barriers in food retail supply chains. *J. Retail. Consum. Serv.* **2019**, *48*, 154–168. [CrossRef]
41. Abdel-Basset, M.; Manogaran, G.; Mohamed, M.; Rushdy, E. Internet of things in smart education environment: Supportive framework in the decision-making process. *Concurr. Comput. Pract. Exp.* **2019**, *31*, e4515. [CrossRef]
42. Ahmed, N.; De, D.; Hussain, I. Internet of Things (IoT) for smart precision agriculture and farming in rural areas. *IEEE Internet Things J.* **2018**, *5*, 4890–4899. [CrossRef]
43. Da Costa, K.A.; Papa, J.P.; Lisboa, C.O.; Munoz, R.; de Albuquerque, V.H.C. Internet of Things: A survey on machine learning-based intrusion detection approaches. *Comput. Netw.* **2019**, *151*, 147–157. [CrossRef]
44. Fu, Y.; Yan, Z.; Cao, J.; Koné, O.; Cao, X. An automata based intrusion detection method for internet of things. *Mob. Inf. Syst.* **2017**, *2017*, 1–13. [CrossRef]
45. Ioulianou, P.; Vasilakis, V.; Moscholios, I.; Logothetis, M. A signature-based intrusion detection system for the internet of things. *Inf. Commun. Technol. Form* **2018**, 1–6. in press.
46. Asharf, J.; Moustafa, N.; Khurshid, H.; Debie, E.; Haider, W.; Wahab, A. A review of intrusion detection systems using machine and deep learning in internet of things: Challenges, solutions and future directions. *Electronics* **2020**, *9*, 1177. [CrossRef]
47. Smys, S.; Abul, B.; Haoxiang, W. Hybrid Intrusion Detection System for Internet of Things (IoT). *J. ISMAC* **2020**, *2*, 190–199. [CrossRef]
48. Jan, S.U.; Ahmed, S.; Shakhov, V.; Koo, I. Toward a lightweight intrusion detection system for the internet of things. *IEEE Access* **2019**, *7*, 42450–42471. [CrossRef]
49. Almomani, I.; Alenezi, M. Efficient Denial of Service Attacks Detection in Wireless Sensor Networks. *J. Inf. Sci. Eng.* **2018**, *34*, 977–1000.
50. Almomani, I.; Al-Kasasbeh, B.; Al-Akhras, M. WSN-DS: A dataset for intrusion detection systems in wireless sensor networks. *J. Sens.* **2016**, *2016*, 1–16. [CrossRef]

51. Choudhary, S.; Kesswani, N. Cluster-Based Intrusion Detection Method for Internet of Things. In Proceedings of the 2019 IEEE/ACS 16th International Conference on Computer Systems and Applications (AICCSA), Abu Dhabi, United Arab Emirates, 3–7 November 2019; pp. 1–8.
52. Bostani, H.; Sheikhan, M. Hybrid of anomaly-based and specification-based IDS for Internet of Things using unsupervised OPF based on MapReduce approach. *Comput. Commun.* **2017**, *98*, 52–71. [CrossRef]
53. Telikani, A.; Gandomi, A.H. Cost-sensitive stacked auto-encoders for intrusion detection in the Internet of Things. *Internet Things* **2019**, 100122, in press. [CrossRef]
54. Ullah, I.; Mahmoud, Q.H. A two-level hybrid model for anomalous activity detection in IoT networks. In Proceedings of the 2019 16th IEEE Annual Consumer Communications & Networking Conference (CCNC), Las Vegas, NV, USA, 11–14 January 2019; pp. 1–6.
55. Shahriar, M.H.; Haque, N.I.; Rahman, M.A.; Alonso, M., Jr. G-IDS: Generative Adversarial Networks Assisted Intrusion Detection System. *arXiv* **2020**, arXiv:2006.00676.
56. Maniriho, P.; Niyigaba, E.; Bizimana, Z.; Twiringiyimana, V.; Mahoro, L.J.; Ahmad, T. Anomaly-based Intrusion Detection Approach for IoT Networks Using Machine Learning. In Proceedings of the 2020 International Conference on Computer Engineering, Network, and Intelligent Multimedia (CENIM), Surabaya, Indonesia, 17–18 November 2020; pp. 303–308.
57. Qaddoura, R.; Faris, H.; Aljarah, I.; Castillo, P.A. Evocluster: an open-source nature-inspired optimization clustering framework in python. In Proceedings of the International Conference on the Applications of Evolutionary Computation (Part of EvoStar), Seville, Spain, 15–17 April 2020; pp. 20–36.
58. Qaddoura, R.; Faris, H.; Aljarah, I. An efficient clustering algorithm based on the k-nearest neighbors with an indexing ratio. *Int. J. Mach. Learn. Cybern.* **2020**, *11*, 675–714. [CrossRef]
59. Fernández, A.; Garcia, S.; Herrera, F.; Chawla, N.V. SMOTE for learning from imbalanced data: Progress and challenges, marking the 15-year anniversary. *J. Artif. Intell. Res.* **2018**, *61*, 863–905. [CrossRef]
60. Yang, J.; Ma, J. Feed-forward neural network training using sparse representation. *Expert Syst. Appl.* **2019**, *116*, 255–264. [CrossRef]
61. Dobbin, K.K.; Simon, R.M. Optimally splitting cases for training and testing high dimensional classifiers. *BMC Med. Genom.* **2011**, *4*, 1–8. [CrossRef] [PubMed]
62. Lemaître, G.; Nogueira, F.; Aridas, C.K. Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning. *J. Mach. Learn. Res.* **2017**, *18*, 1–5.
63. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.