future internet

MDPI

Article

RDTIDS: Rules and Decision Tree-Based Intrusion Detection System for Internet-of-Things Networks

Mohamed Amine Ferrag [1,*], Leandros Maglaras [2], Ahmed Ahmim [3], Makhlouf Derdour [4] and Helge Janicke [5]

1 Department of Computer Science, Guelma University, Guelma 24000, Algeria
2 School of Computer Science and Informatics, De Montfort University, Leicester LE1 9BH, UK; leandrosmag@gmail.com
3 Departement of Mathematics and Computer Science, Mohamed-Cherif Messaadia University, Souk Ahras 41000, Algeria; a.ahmim@gmail.com
4 Departement of Mathematics and Computer Science, University of Larbi Tebessi, Tebessa 12002, Algeria; m.derdour@yahoo.fr
5 Cyber Security Cooperative Research Centre, Edith Cowan University, Perth 6027, Australia; Helge.Janicke@cybersecuritycrc.org.au
* Correspondence: ferrag.mohamedamine@univ-guelma.dz

Received: 24 January 2020; Accepted: 27 February 2020; Published: 2 March 2020

Abstract: This paper proposes a novel intrusion detection system (IDS), named RDTIDS, for Internet-of-Things (IoT) networks. The RDTIDS combines different classifier approaches which are based on decision tree and rules-based concepts, namely, REP Tree, JRip algorithm and Forest PA. Specifically, the first and second method take as inputs features of the data set, and classify the network traffic as Attack/Benign. The third classifier uses features of the initial data set in addition to the outputs of the first and the second classifier as inputs. The experimental results obtained by analyzing the proposed IDS using the CICIDS2017 dataset and BoT-IoT dataset, attest their superiority in terms of accuracy, detection rate, false alarm rate and time overhead as compared to state of the art existing schemes.

Keywords: intrusion detection; IDS; hybrid IDS; learning machine; hierarchical; network security

1. Introduction

In recent years cyberattacks, especially those targeting systems that keep or process sensitive information, are becoming more sophisticated. Critical National Infrastructures are the main targets of cyber attacks since essential information or services depend on their systems and their protection becomes a significant issue that is concerning both organizations and nations [1–3]. Attacks to such critical systems include penetrations to their network and installation of malicious tools or programs that can reveal sensitive data or alter the behavior of specific physical equipment. In order to tackle this growing trend, academics and industry professionals are joining forces in an attempt to develop novel systems and mechanisms that can defend their systems. Along with other preventive security mechanisms, such as access control and authentication, intrusion detection systems (IDSs) are deployed as a second line of defense. IDSs based on some specific rules or patterns of the normal behavior of the system can distinguish between normal and malicious actions [4,5].

Many different taxonomies for IDSs have been proposed until now. Based on the classification model they use, IDSs can be classified as rule-based, misuse detection and mixed systems. IDSs can also be classified as real-time if they use continuous monitoring of the system or periodic or offline if the detection happens in specific time instances or even offline using data that are collected and

stored during a certain period of time. Moreover, when talking about Industrial Control Systems (ICSs) that have specific requirements and characteristics novel taxonomies were recently proposed. Authors in [6] proposed a classification of IDSs ICS that divides them into three new categories: protocol analysis-based, traffic mining-based, and control process analysis-based.

Countermeasures are taken according to the information obtained regarding the detected attacks from the detection systems. The better classification of the type of attack is provided, the more efficient will the chosen countermeasures be and the less will they affect the proper operation of the system or network. Moreover, if we do not detect the exact type of attack the countermeasures can have more serious consequences than the attack itself in some cases. For this reason, our goal is to create an intrusion detection model that correctly classifies each type of attack. In addition, our model must provide a low false alarm rate and a high detection rate both for frequent and infrequent attacks while on the same require low computing in order to perform classification. The latter characteristic is very important when IDSs are deployed in industrial control systems that operate critical infrastructures where correct and fast notification about cyber attacks is crucial [7]. This paper extends our work in [8] in many ways. First, we integrate the proposed system for the internet of things networks. We have present and critically analyze new related works that were published recently. We have redesigned the proposed system, we have presented the integration of RDTIDs into a three-tier fog computing architecture for IoT networks and have evaluated its performance against a Bot-IoT dataset.

The remainder of the paper is organized as follows. Section 2 discusses related work and places the research within that of the wider community. Section 3 introduces the key concepts and the overall architecture of the proposed system. Section 4 presents the experimentation setup, gives the simulation parameters and describes the evaluation of the method. Section 5 concludes the paper.

2. Relevant Work

Table 1 lists representative related works on hybrid intrusion detection systems, including machine learning and data mining methods. It also presents the security issue that each one of these methods tries to address, along with the dataset that was used in order to evaluate their performance.

Aydin et al. [9] proposed a hybrid IDS by combining two approaches, namely, (1) packet header anomaly detection (PHAD), and (2) network traffic anomaly detection (NETAD) with the signature-based IDS Snort. Both PHAD and NETAD methods are anomaly-based IDSs. The aydın et al.'s system is tested on IDEVAL data, which shows that the number of attacks detected increases significantly using the proposed hybrid IDS as compared to signature-based systems. Wang et al. [10] proposed an intrusion detection approach, named FC-ANN, based on artificial neural networks (ANN) and fuzzy clustering. The FC-ANN approach uses three main modules, i.e., fuzzy clustering module, ANN module, and fuzzy aggregation module. The fuzzy clustering module is used to partition a given set of data into clusters. The ANN module is used to learn the pattern of every subset. The fuzzy aggregation module is used to aggregate different ANN's results and reduce any detection errors. The FC-ANN approach was tested on the KDD CUP 1999 dataset and was proven to be efficient against low-frequent attacks, i.e., R2L and U2R attacks.

Govindarajan and Chandrasekaran [11] proposed a neural-based hybrid IDS architecture using two methods, namely, 1) multilayer perceptron neural network (MLP), and 2) radial basis function neural network (RBF). The procedures of hybrid modeling using bagging classifiers are employed in order to increase robustness, accuracy, and better overall generalization. Moreover, the UNM Send-Mail Data is used in this study, which is based on an immune system developed at the University of New Mexico. The performance of the proposed IDS in terms of accuracy was 98.88% and 94.31% for normal as well as abnormal traffic respectively, which is better compared to the performance of the classifiers that compose it.

Table 1. Related works on hybrid intrusion detection systems.

Year	Paper	Machine Learning and Data Mining Methods	Cyber Approach	Data Used
2009	Aydin et al. [9]	- Packet header anomaly detection - Network traffic anomaly detection	Hybrid IDS, which combing anomaly-based IDSs	IDEVAL
2010	Wang et al. [10]	- Artificial neural networks Fuzzy clustering	Hybrid IDS, which the fuzzy aggregation module is employed to aggregate the results	KDD CUP 1999
2011	Govindarajan and Chandrasekaran [11]	- Multilayer perceptron neural network - Radial basis function neural network	Neural based hybrid IDS	UNM Send-Mail Data
2012	Chung and Wahid [12]	- Intelligent dynamic swarm - Simplified swarm optimization	Hybrid IDS	KDD CUP 1999
2013	Elbasiony et al. [13]	- Random forests algorithm - K-means clustering algorithm	Combining misuse and anomaly detection into a hybrid framework	KDD CUP 1999
2014	Kim et al. [14]	- C4.5 decision tree algorithm - Support vector machine model	Combining misuse and anomaly detection into a hybrid framework	NSL-KDD
2015	Lin et al. [15]	- k-Nearest Neighbor (k-NN) classifier	Combining cluster centers and nearest neighbors	KDD CUP 1999
2016	Aslahi-Shahri et al. [16]	- Support vector machine - Genetic algorithm	Hybrid IDS	KDD CUP 1999
2017	Kevric et al. [17]	- Random tree - C4.5 decision tree algorithm - NBTree	Combining classifier model based on tree-based algorithms	NSL-KDD
2017	Al-Yaseen et al. [18]	- Support vector machine - Extreme learning machine - K-means clustering algorithm	Hybrid IDS	KDD CUP 1999
2018	Ahmim et al. [19]	- Repeated Incremental Pruning to Produce Error, Reduction (RIPPER), RBF Network (RBFN), Ripple-down rule learner (Ridor), and Random Forests - Naive Bayes (NB)	Combining probability predictions of a tree of classifiers	KDD CUP 1999 + NSL-KDD
2018	Aljawarneh et al. [20]	J48, Meta Pagging, RandomTree, REPTree, AdaBoostM1, DecisionStump, and NaiveBayes	Hybrid IDS, which combing anomaly-based IDSs	NSL-KDD
2019	Ferrag et al. [21]	Recurrent neural networks	Hybrid IDS, which combing recurrent neural networks with blockchain	CICIDS2017, Power System dataset, and Bot-IoT
2019	Derhab et al. [22]	Random subspace learning	Hybrid IDS, which combing random subspace learning with blockchain	Power System dataset
2019	Ferrag et al. [23]	Deep learning techniques	Hybrid IDS, which combing deep learning techniques with blockchain	CSE-CIC-IDS2018 dataset
	RDTIDS	- REP Tree - JRip algorithm - Random Forest	Hybrid IDS, which combining classifier model based on tree-based algorithms	CICIDS2017

IDEVAL: MIT Lincoln Laboratories network traffic data; KDD CUP 1999: The data set is based on DARPA 1998 TCP/IP data and has basic features captured by pcap (with about 4 million records of normal and attack traffic); UNM Send-Mail Data: The data set is based on an immune system developed at the University of New Mexico; NSL-KDD: A modified version of KDD'99 data set, which it does not include redundant records in the train set; CICIDS2017: The dataset contains benign and the most up-to-date common attacks, which resembles the true real-world data (PCAPs) developed at Canadian Institute for Cybersecurity (University of New Brunswick) [24].

Chung and Wahid [12] approach the problem of decision rules generation by employing intelligent dynamic swarm-based rough set (IDS-RS) for feature selection and simplified swarm optimization with weighted local search (SSO–WLS) strategy for data classification. The study provides a full system solution for improving the searching process in SSO rule mining by weighing three predetermined constants. The experimental results on the KDD CUP 1999 dataset show that the proposed hybrid network intrusion detection system using an intelligent dynamic swarm-based rough set shows a good overall performance with 93.3% accuracy in an average of 20 runs.

Elbasiony et al. [13] presented a combination of misuse and anomaly detection into a hybrid framework, which is based on two methods, namely, (1) random forests algorithm, and (2) K-means clustering algorithm. Specifically, this framework employs the random forests algorithm in misuse intrusion detection as well as the k-means clustering algorithm in anomaly detection. Due to correlated variables in random forests, this framework has low model interpretability and performance loss. Kim et al. [14] integrate a misuse detection model and an anomaly detection model in a decomposition structure. This study uses the C4.5 decision tree algorithm and multiple one-class SVM models. The experimental results on the NSL-KDD dataset show that the proposed hybrid intrusion detection method is better than the conventional methods in terms of detection performance, training time, and testing time.

Lin et al. [15] proposed a feature representation approach, named CANN, which combines cluster centers and nearest neighbors. The CANN approach uses three steps, namely, (1) Extraction of cluster centers and nearest neighbors, (2) Measurement and summing of the distance between all data, and (3) Classifier training and testing, which is based on the k-NN algorithm. The experimental results on the KDD CUP 1999 dataset show that CANN approach performs better than the k-NN and SVM classifiers.

Recently, a number of researchers have proposed the combination of classifiers in order to improve the overall performance under the NSL-KDD dataset [17–20,24]. In [17], Kevric et al. proposed a combined classifier approach using tree algorithms, namely, random tree, C4.5 decision tree algorithm, and NBTree. In [18], Al-Yaseen et al. proposed a hybrid IDS that uses support vector machine, extreme learning machine, and K-means clustering algorithm. The experimental results on the KDD CUP 1999 dataset show that the proposed hybrid network intrusion detection system can improve the overall performance and achieve an overall 95.75% accuracy.

In order to evaluate the performance of an IDS system, both KDD'99 and NSL-KDD datasets are commonly used. Ahmim et al. [19] proposed an IDS system, named HCPTC-IDS, which is based on combining probability predictions of a tree of classifiers. The HCPTC-IDS system is composed of two layers, namely, 1) the first layer, which is a tree of classifiers, and 2) the second layer, which is a final classifier that combines the different probability predictions of the first layer. The experiments on KDD'99 and NSL-KDD show that the HCPTC-IDS system is more precise than other recently proposed intrusion detection systems having accuracy equal to 96.27% for KDD'99 and 89.75% for NSL-KDD. The authors in [16] presented an anomaly detection technique based on support vector machine (SVM) and genetic algorithm (GA), in order to improve the performance of classification for SVMs. The experimental results on the KDD CUP 1999 dataset show an outstanding true-positive value of 0.973 that comes with a 0.017 false-positive value.

The blockchain technology [25] is combined with machine learning and data mining methods for cyber security intrusion detection. Derhab et al. [22] proposed an intrusion detection system, named RSL-KNN, against forged command attacks, which is based on the random subspace learning approach and K-Nearest Neighbor (KNN) classifier. The RSL-KNN system is combined with the blockchain technology for detecting in a short time any tampering with the OpenFlow rule. Ferrag et al. [21] proposed a novel deep learning and blockchain-based energy framework, called DeepCoin, smart grids. The DeepCoin framework uses two schemes, namely, a deep learning-based scheme and a blockchain-based scheme. The deep learning-based scheme is used for detecting network attacks based on recurrent neural networks, while the blockchain-based scheme is used for detecting fraudulent transactions. Similarly to DeepCoin framework, Ferrag and Leandros [23]

proposed an intrusion detection system combined with the blockchain-based delivery framework, called DeliveryCoin, for drone-delivered services. Both DeepCoin and DeliveryCoin frameworks demonstrated the efficiency of the deep learning techniques in cyber security intrusion detection. For more detail about the deep learning techniques in cyber security intrusion detection, we refer the reader to our recent studies in [26,27].

Most of the relevant works use the KDD and NSL-KDD datasets, which are outdated and of very limited practical value for a modern IDS. Both benign and malicious network traffic has changed significantly since 1999 when these datasets were produced and the results obtained using them are of a limited value most of the times.

In order to overcome several shortcomings of previous proposed methods, like low detection of rare attack, mis-classification of attacks and time overhead, we propose a novel Hybrid IDS, which combines different classifier models, namely, REP Tree, JRip algorithm and Random Forest. Besides, we use the CICIDS2017 dataset [24], which we split in training and testing datasets, in order to evaluate their performance in detecting network intrusions and we compare it with other machine learning methods proposed by previous researchers, including, WISARD [28], ForestPA [29], J48 Consolidated [30], LIBSVM [31], FURIA [32], RandomForest, REPTree, MLP, NaiveBayes, JRip and J48.

3. The Proposed Model

According to [33], data imbalance is one of the causes that degrades the performance of machine learning algorithms in classifications. This is due to two major causes. First, the simple accuracy as an objective function used in most classification tasks is inadequate when the classifier faces data imbalance and the second comes from the distribution of the classes where the majority class is likely to invade the territory of the minority class. In general, for the same data set, if the number of classes increases, the sub-data set of each class decreases, leading to the decrease of the generalization capability and the increase of classification errors and vice versa. The mis-classification is usually due to the classification of attacks as normal behavior, or as another attack in the same or different category.

In order to minimize these classification errors and to increase the performance of intrusion detection mechanism, we propose the hierarchical intrusion detection model illustrated in Figure 1. The main feature of the RTDITS is that uses a binary and multi-class classifier in parallel, that feed a third classifier, thus minimizing the effect of data imbalance to the final outcome. Moreover, ensemble systems of classifiers are widely used for intrusion detection in networks (maglaras2016combining). These aim to include mutually complementary individual classifiers, which are characterized by high diversity in terms of classifier structure [34], internal parameters or classifier inputs. As stated in [35], ensemble and meta-ensemble methods show a number of advantages with regard to using a single model, i.e., they reduce the variance of the error, the bias, and the dependence on a single dataset and work well in the case of unbalanced classes.

Our hierarchical model aims to correctly classify each attack, providing a low false alarm rate and a high detection rate. It is composed of three classifiers. The first one uses the different features of the data set as inputs, in order to classify each row as benign or malicious. The second one uses different features of the data set as inputs for classifying each row as benign or one of the different categories of attacks. The third classifier uses all the features of the initial data set in addition to the outputs of the first and the second classifier as inputs, in order to classify each row of the data set as benign or a specific type of attack. For the integration of RDTIDS system for IoT networks, we propose a three-tier fog computing architecture, in which RDTIDS system is located in the fog computing layer, as presented in Figure 2. The classifiers that are used are REP Tree [36] Jrip [37] and Forest PA [29]. Based on the values of the features, the classifier makes rules which are used in order to correctly classify the data set.

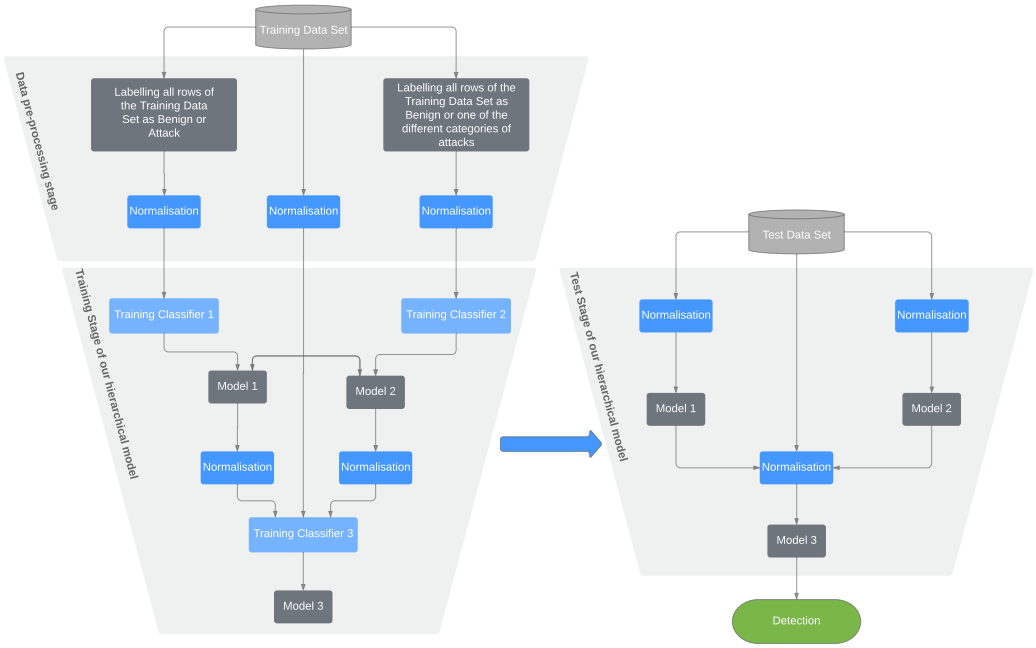


Figure 1. General structure of Rules and Decision Tree-based Intrusion Detection System (RDTIDS) system.

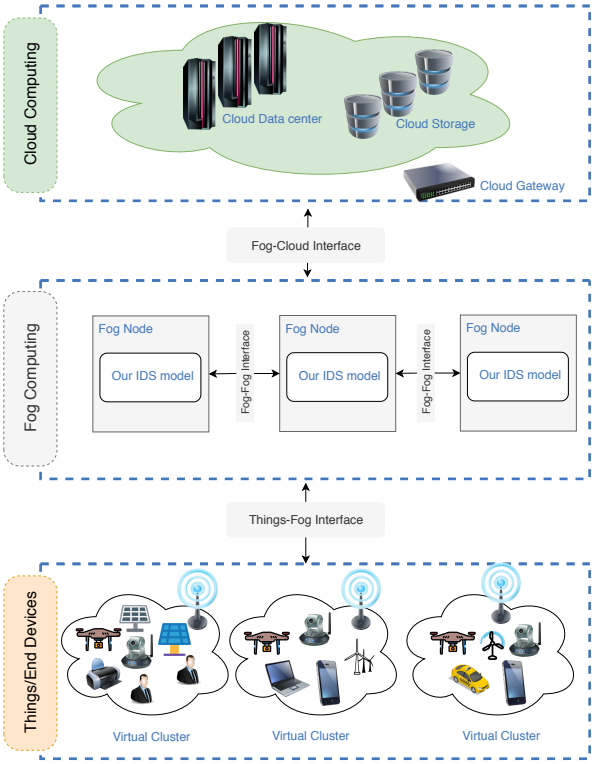


Figure 2. Integration of RDTIDS system in the three-tier fog computing architecture for Internet-of-Things (IoT) networks.

3.1. Operation Mode

The operation mode of RDTIDS system is composed of two steps: training and testing. During the first step, we train the three classifiers that compose our hierarchical model. We start by training the first and the second one and using the results from these two we train the third classifier. Initially,

the training Data set is labelled as benign and specific type of Attack. To train the first classifier, we modify the training data set, by labelling each row as "Attack" and "Benign". Then, we normalize the different features of the data set. After that, we perform the training of this classifier and as result of this sub step, we get model 1. To train the second classifier, we modify the training data set, by labelling the rows with specific attacks. Then, we normalize the different features of the data set. After that, we perform the training of this classifier and as a result of this sub step, we get model 2.

In order to train the third classifier, we modify the training data set, where we add two columns, the first one represents the classification results of model 1 for the rows of the training data set and the second one represents the classification results of model 2 for the rows of the training data set. Then, we normalize the different features of the data set. After that, we perform the training of this classifier and as result of this sub step, we get model 3.

As illustrated in Figure 1, in order to test our hierarchical model, we process each row of the test data set by model 1 and model 2, then we add the outputs of model 1 and model 2 to the features of each row, and finally, we process the result rows by model 3 that classify it as benign or a specific type of attack.

The whole procedure of transforming the dataset into information that can be read and interpreted from our classifiers is call pre-processing and consists of two steps: the first involves mapping symbolic-valued attributes to numeric-valued attributes and the second is implemented scaling. The main advantage of scaling is to avoid attributes in greater numeric ranges dominating those in smaller numeric ranges. Another advantage is to avoid numerical difficulties during the calculations.

4. Experimentation

In this section, we present in detail the Data Set used along with the Data pre-processing procedure. We also give the performance metrics used in our experiments. Moreover, we present the structure of our model. Finally, we provide a comparative study between our model and that of different classifiers. The experiments are done on a Windows 10, 64 bits PC with 8 GB RAM and CPU Intel(R) I5 2.7 GHz. Weka Data Mining Tools and MySQL RDBMS are used to implement our model.

4.1. Data Set and Data Pre-Processing

We use two new real traffic datasets, namely the CICIDS 2017 dataset [24] and the Bot-IoT dataset [38] for the experiments. Tables 2 and 3 summarizes the statistics of attacks in Training and Test in both datasets. Both datasets satisfy the eleven indispensable characteristics of a valid IDS dataset, namely Anonymity, Attack Diversity, Complete Capture, Complete Interaction, Complete Network Configuration, Available Protocols, Complete Traffic, Feature Set, Metadata, Heterogeneity, and Labelling [39].

The BoT-IoT dataset contains more than 72,000,000 records devised on 74 files, each row having 46 features. We use the version proposed by Koroniotis et al. [40], which is a version of training and testing with 5% of the entire dataset.

The CSV version of CICIDS 2017 contains 2,830,743 rows divided in 8 files, each row having 79 features. Each row of CICIDS 2017 is labelled as Benign or one of fourteen types of attack.

In order to create a training and test subset, we concatenate the 8 files in one file containing a unique table that contains all benign and attacks rows. Then, we remove all rows that have the feature "Flow Packets/s" equal to 'Infinity' or 'NaN'. After that, we remove identical rows, namely Bwd PSH Flags,Bwd URG Flags, Fwd Avg Bytes/Bulk, Fwd Avg Packets/Bulk, Fwd Avg Bulk/Rate, Bwd Avg Bytes/Bulk, Bwd Avg Packets/Bulk, Bwd Avg Bulk/Rate and Fwd Avg Bytes/Bulk.

After the elimination of these features, we extract the training and test subsets based on the distribution described in Table 2. In each subset we tried to include rows that contain all the attacks but the same row cannot appear in both subsets. For the training subset, we select the first rows of each type. Then, for the test subset, we select randomly some rows after the suppression of the training subset rows.

Table 2. Attack Types in CICIDS 2017 dataset.

	Category	Total	Total (-Rows with Lack Info)	Training	Test
BENIGN	BENIGN	2,273,097	2,271,320	20,000	20,000
DOS	DDoS	128,027	128,025	2700	3300
	DoS slowloris	5796	5796	1350	1650
	DoS Slowhttptest	5499	5499	2171	1169
	DoS Hulk	231,073	230,124	4500	5500
	DoS GoldenEye	10,293	10,293	1300	700
	Heartbleed	11	11	5	5
PortScan	PortScan	158,930	158,804	3808	4192
Bot	Bot	1966	1956	936	624
Brute-Force	FTP-Patator	7938	7935	900	1100
	SSH-Patator	5897	5897	900	1100
Web Attack	Web Attack-Brute Force	1507	1507	910	490
	Web Attack-XSS	652	652	480	160
	Web Attack-SQL Injection	21	21	16	4
Infiltration	Infiltration	36	36	24	6
Total Attack		471,454	470,365	20,000	20,000
Total		2,830,743	2,827,876	40,000	40,000

Table 3. Attack Types in Bot-IoT dataset.

Category	Attack Type	Flow Count	Training	Test
BENIGN	BENIGN	9543	7634	1909
Information gathering	Service scanning	1,463,364	117,069	29,267
	OS Fingerprinting	358,275	28,662	7166
DDoS attack	DDoS TCP	19,547,603	1,563,808	390,952
	DDoS UDP	18,965,106	1,517,208	379,302
	DDoS HTTP	19,771	1582	395
DoS attack	DoS TCP	12,315,997	985,280	246,320
	DoS UDP	20,659,491	1,652,759	413,190
	DoS HTTP	29,706	2376	594
Information theft	Keylogging	1469	1175	294
	Data theft	118	94	24
Total	/	73,370,443	5,877,647	1,469,413

Finally, each value x_i of the feature j is normalized based on the following equation:

$$\overline{x_i(j)} = \frac{x_i(j) - min(x(j))}{max(x(j)) - min(x(j))} \quad (1)$$

4.2. Performance Metrics

IDS performance is evaluated based on its capability of classifying network traffic into a correct type. Table 4, also known as confusion matrix, shows all the possible cases of classification.

To evaluate RDTIDS system, we used two groups of metrics. The first group includes specific metrics: the detection rate (DR) of each type of attack and the true negative rate. The second one includes global metrics: global detection rate, false alarm rate (FAR) and accuracy. The following equations summarize how to calculate these metrics.

$$DR_{AttackType} = \frac{TP_{AttackType}}{TP_{AttackType} + FN_{AttackType}} \quad (2)$$

$$TNR_{BENIGN} = \frac{TN_{BENIGN}}{TN_{BENIGN} + FP_{BENIGN}} \quad (3)$$

$$FAR = \frac{FP_{BENIGN}}{TN_{BENIGN} + FP_{BENIGN}} \quad (4)$$

$$DR_{Overall} = \frac{\sum TP_{Of-Each-Attack-Type}}{\sum TP_{Of-Each-Attack-Type} + \sum FN_{Of-Each-Attack-Type}} \quad (5)$$

$$Accuracy = \frac{\sum TP_{Of-Each-Attack-Type} + TN_{BENIGN}}{\sum TP_{Of-Each-Attack-Type} + \sum FN_{Of-Each-Attack-Type} + TN_{BENIGN} + FP_{BENIGN}} \quad (6)$$

Table 4. Confusion Matrix.

		Predicted Class	
		Negative Class	**Positive**
Actual class	Negative Class	True negative (TN)	False positive (FP)
	Positive Class	False negative (FN)	True positive (TP)

4.3. Practical Structure of RDTIDS System

The RDTIDS system demands a pre-processing of data, different labelling of rows according to the model that is used and creation of artificial features that are outputs of two of the classifiers used. For training the first classifier we need to label the training data set based on the attacks classification provided in Table 2 as “Attack” or “Benign”. For the second classifier, we label each attack as follows: DDoS, DoS slowloris, DoS Slowhttptest, DoS Hulk, DoS GoldenEye, Heartbleed as “DoS”; FTP-Patator, SSH-Patator as "Brute-Force"; Web Attack—Brute Force, Web Attack—XSS, Web Attack—SQL Injection as “Web Attack”. Finally, in order to train the third classifier, we use the labeling used for the second classifier. We also add the outputs of the trained classifier 1 and classifier 2 as new features.

The choice of the three classifiers that compose our hierarchical model is the most important and critical step. To choose the best composition, that gives optimal performance, we tested several combinations with different classifiers. The results for all those combinations are quite lengthy and are not represented in this article. Following this demanding procedure, we opt for the following configuration: classifier 1 is REP Tree [36]; classifier 2 is Jrip [37]; classifier 3 is Forest PA [29].

4.4. Comparative Study

To evaluate RDTIDS system, we compare it with some well known classifiers and some recent ones namely J48, Jrip, Naive Bayes, MLP, REP Tree, Random Forest, FURIA [32], LIBSVM [31], J48 Consolidated [30], Forest PA [29], WISARD [28]. In this comparative study we use the different metrics detailed in Section 4.2, in addition to training and testing time.

Table 5 summarizes the performance of RDTIDS system compared to the other classifiers for different attacks and Benign traffic. It shows that RDTIDS system gives the highest true negative rate (TNR) with 98.855% and the higest detection rate (DR) for six attacks type namely DDoS with 99.879%, DoS slowloris with 97.758%, DoS Slowhttptest with 93.841%, DoS GoldenEye with 67.571%, Heartbleed 100%, PortScan 99.881%, Infiltration 100%. Moreover, the RDTIDS system is very close to the highest detection rate for two types of attacks namely FTP Patator with 99.636% and SSH Patator with 99.909%. For the rest of the attacks types, the RDTIDS system gives an average performance compared to the other models. Overall, RDTIDS system is the model that performs best for most of the different attack types and it never has the lowest performance for any type of attack.

Figures 3–6 present the overall performance of RDTIDS and other classifiers in terms of false alarm rate, global detection rate, accuracy, training and test time, respectively. In terms of detection rate, RDTIDS gives the highest overall detection rate (DR Overall) with 94.475% and 95.175% in CICIDS 2017 dataset and Bot-IoT dataset, respectively. In terms of accuracy, RDTIDS gives the highest accuracy with 96.665% and 96.995% in CICIDS 2017 dataset and Bot-IoT dataset, respectively. In terms of false

alarm rate (FAR), RDTIDS gives the lowest false alarm rate with 1.145% and 1.120% in CICIDS 2017 dataset and Bot-IoT dataset, respectively. The training time of RDTIDS is 195.5 s and the test time is 2.27 s, which represents an acceptable training and test time for a hybrid hierarchical model especially when compared to simple models such as MLP and SVM.

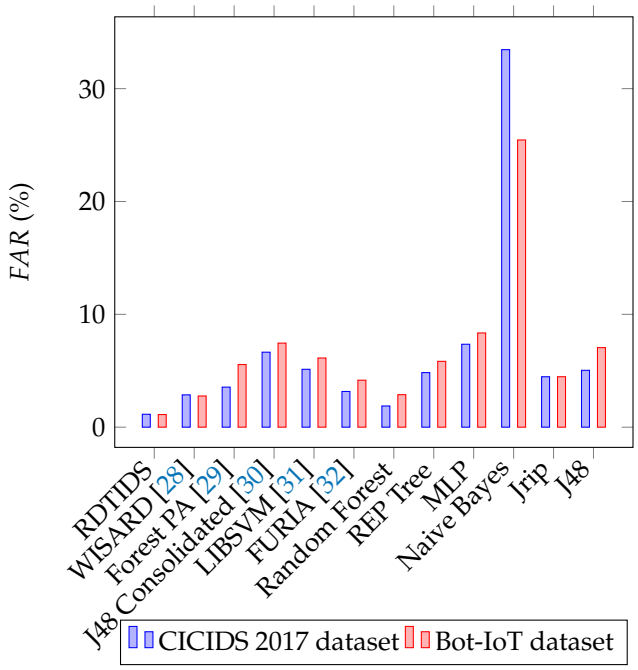


Figure 3. Overall Performance of RDTIDS and other classifiers in terms of false alarm rate.

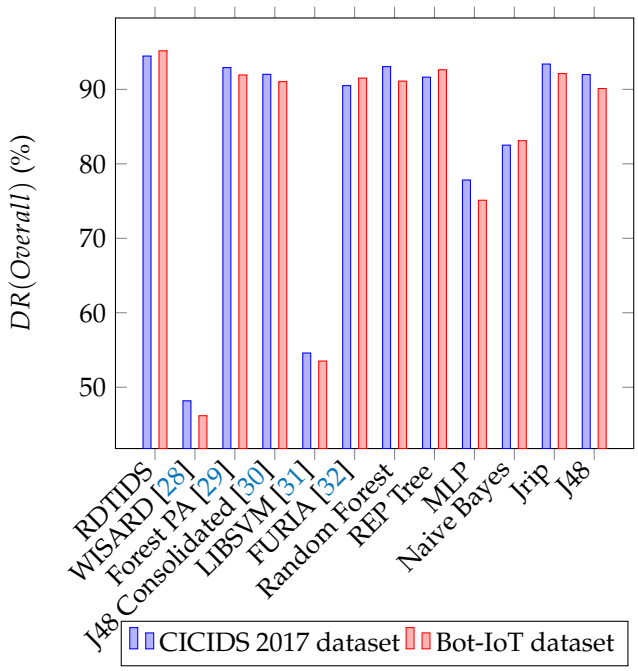


Figure 4. Overall Performance of RDTIDS and other classifiers in terms of global detection rate.

Table 5. Performance of RDTIDS system and other classifiers relative to the different attack type and Benign.

	RDTIDS	[28]	[29]	[30]	[31]	[32]	Random Forest	REP Tree	MLP	Naive Bayes	Jrip	J48
TNR (BENIGN)	98.855%	97.135%	96.450%	93.355%	94.870%	96.835%	98.120%	95.165%	92.650%	66.545%	95.530%	94.960%
DR DDoS	99.879%	54.697%	99.818%	93.212%	55.970%	99.758%	99.818%	99.788%	91.212%	93.879%	99.667%	99.788%
DR DoS slowloris	97.758%	78.909%	92.848%	95.030%	78.182%	93.758%	93.758%	92.727%	78.485%	82.667%	93.333%	93.879%
DR DoS Slowhttptest	93.841%	23.353%	86.826%	83.832%	76.561%	78.358%	81.352%	75.364%	88.537%	70.060%	85.543%	80.325%
DR DoS Hulk	96.782%	67.600%	93.945%	95.891%	73.709%	98.655%	95.164%	92.218%	86.891%	73.782%	97.364%	93.600%
DR DoS GoldenEye	67.571%	48.714%	67.571%	67.143%	57.571%	65.143%	67.571%	66.429%	65.429%	62.143%	63.857%	67.286%
DR Heartbleed	100%	80.000%	100%	80.000%	0.000%	40.000%	100%	100%	0.000%	80.000%	80.000%	100%
DR PortScan	99.881%	51.407%	99.594%	99.046%	48.521%	87.118%	99.881%	99.881%	48.521%	99.499%	99.881%	98.569%
DR Bot	46.474%	1.442%	48.718%	52.083%	0.000%	48.077%	49.679%	47.756%	51.282%	29.968%	46.474%	47.756%
DR FTP-Patator	99.636%	0.000%	99.727%	100%	0.000%	99.636%	99.727%	99.182%	99.000%	99.455%	99.545%	99.545%
DR SSH-Patator	99.909%	0.000%	100%	99.727%	0.000%	100%	99.818%	100%	99.727%	99.182%	100%	100%
DR Web Attack—Brute Force	73.265%	4.694%	73.469%	55.102%	80.816%	49.796%	70.408%	70.816%	90.408%	5.102%	61.837%	60.408%
DR Web Attack—XSS	30.625%	1.250%	34.375%	48.750%	0.000%	38.750%	37.500%	32.500%	1.875%	91.875%	38.125%	41.250%
DR Web Attack—SQL Injection	50.000%	0.000%	50.000%	100%	0.000%	50.000%	100%	50.000%	50.000%	100%	75.000%	50.000%
DR Infiltration	100%	50.000%	83.333%	100%	0.000%	83.333%	83.333%	83.333%	16.667%	83.333%	100%	66.667%
DR Service scanning	99.472%	54.697%	99.111%	92.211%	53.170%	99.158%	99.118%	99.188%	91.212%	93.172%	99.267%	94.718%
DR OS Fingerprinting	98.158%	79.109%	93.141%	94.111%	77.182%	94.7581%	94.751%	93.100%	79.185%	83.967%	94.381%	94.129%
DR DDoS TCP	95.841%	22.322%	87.727%	84.532%	77.869%	79.559%	82.454%	76.666%	89.937%	71.223%	86.599%	80.325%
DR DDoS UDP	98.655%	67.600%	93.945%	95.891%	73.709%	96.782%	96.466%	91.318%	84.441%	72.811%	95.119%	93.600%
DR DDoS HTTP	93.171%	47.914%	68.888%	68.242%	59.971%	65.143%	67.222%	67.666%	66.489%	63.243%	62.111%	61.236%
DR DoS TCP	100%	82.111%	100%	82.123%	2.000%	40.234%	92.123%	100%	3.000%	80.000%	82.123%	100%
DR DoS UDP	100%	55.007%	100%	100%	41.198%	86.119%	100%	99.991%	49.521%	98.422%	99.181%	98.111%
DR DoS HTTP	77.474%	1.442%	48.718%	49.083%	0.000%	49.012%	49.373%	47.756%	51.282%	29.968%	47.174%	46.156%
DR Keylogging	100%	0.000%	99.727%	100%	0.000%	99.145%	99.727%	98.182%	100%	99.855%	98.545%	98.145%
DR Data theft	100%	0.000%	100%	99.222%	0.000%	100%	99.718%	100%	99.927%	99.982%	99.187%	99.276%

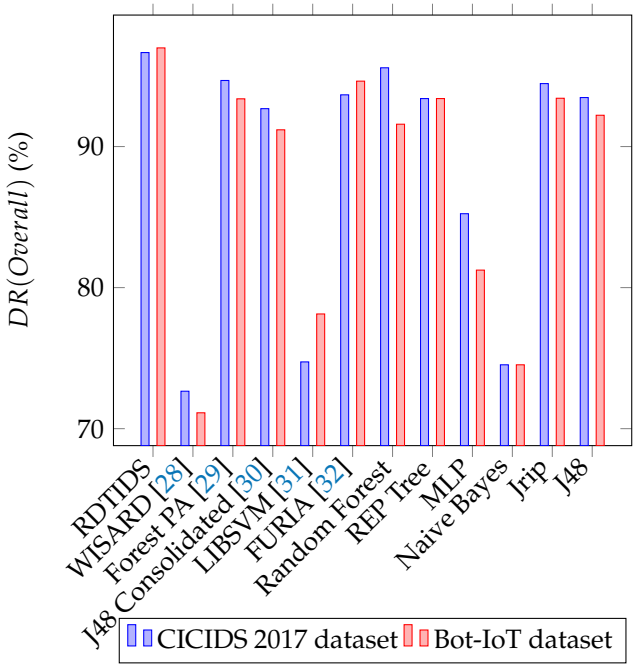
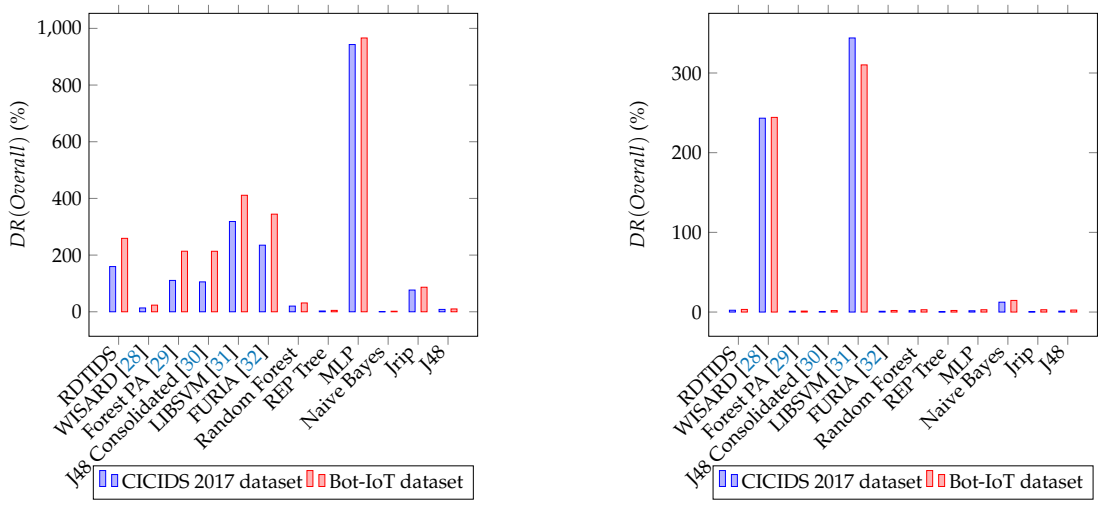


Figure 5. Overall Performance of RDTIDS and other classifiers in terms of accuracy.



(**a**) Training time (**b**) Test time

Figure 6. Overall Performance of RDTIDS and other classifiers in terms of training and test time.

5. Conclusions

In this paper, we proposed a hierarchical intrusion detection system based on the combination of three different classifiers, namely REP Tree, JRip algorithm and Forest PA. The proposed model consists of three classifiers, where two of them operate in parallel and feed the third one. The evaluation using a real traffic data set 'CICIDS2017' and 'BoT-IoT' showed that our hierarchical model outperformed different well known and recent machine learning models, giving the highest TNR and highest DR for seven types of attacks. Overall, RDTIDS gives the highest DR with 94.475% and 95.175%, the highest accuracy with 96.665% and 96.995%, and the lowest FAR with 1.145% and 1.120%.

Author Contributions: Conceptualization, M.A.F., A.A., and L.M.; Methodology, M.A.F., A.A., and L.M.; Software, M.A.F., A.A., and L.M.; Validation, M.A.F., A.A., and L.M.; formal analysis, M.A.F. investigation, M.A.F., A.A., and L.M.; resources, M.A.F., A.A., and L.M.; data curation, M.A.F., A.A., and L.M.; writing—original draft preparation, M.A.F., A.A., and L.M.; writing—review and editing, M.D., H.J., and L.M.; visualization, M.A.F., M.D., H.J., and L.M.; supervision, L.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

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

References

1. Maglaras, L.A.; Kim, K.H.; Janicke, H.; Ferrag, M.A.; Rallis, S.; Fragkou, P.; Maglaras, A.; Cruz, T.J. Cyber security of critical infrastructures. *ICT Express* **2018**, *4*, 42–45. [CrossRef]
2. Ferrag, M.A. EPEC: An efficient privacy-preserving energy consumption scheme for smart grid communications. *Telecommun. Syst.* **2017**, *66*, 671–688. [CrossRef]
3. Ferrag, M.A.; Nafa, M.; Ghanemi, S. EPSA: An efficient and privacy-preserving scheme against wormhole attack on reactive routing for mobile ad hoc social networks. *Int. J. Secur. Netw.* **2016**, *11*, 107–125. [CrossRef]
4. Alcaraz, C.; Zeadally, S. Critical infrastructure protection: Requirements and challenges for the 21st century. *Int. j. Crit. Infrastruct. Prot.* **2015**, *8*, 53–66. [CrossRef]
5. Maglaras, L.; Cruz, T.; Ferrag, M.A.; Janicke, H. Teaching the process of building an Intrusion Detection System using data from a small-scale SCADA testbed. *Internet Technol. Lett.* **2020**, *3*, e132. [CrossRef]
6. Hu, Y.; Yang, A.; Li, H.; Sun, Y.; Sun, L. A survey of intrusion detection on industrial control systems. *Int. J. Distrib. Sens. Netw.* **2018**, *14*, 1550147718794615. [CrossRef]
7. Cruz, T.; Rosa, L.; Proença, J.; Maglaras, L.; Aubigny, M.; Lev, L.; Jiang, J.; Simoes, P. A cybersecurity detection framework for supervisory control and data acquisition systems. *IEEE Trans. Ind. Inf.* **2016**, *12*, 2236–2246. [CrossRef]
8. Ahmim, A.; Maglaras, L.; Ferrag, M.A.; Derdour, M.; Janicke, H. A novel hierarchical intrusion detection system based on decision tree and rules-based models. In Proceedings of the 2019 15th International Conference on Distributed Computing in Sensor Systems (DCOSS), Santorini Island, Greece, 29–31 May 2019; pp. 228–233.
9. Aydın, M.A.; Zaim, A.H.; Ceylan, K.G. A hybrid intrusion detection system design for computer network security. *Comput. Electr. Eng.* **2009**, *35*, 517–526. [CrossRef]
10. Wang, G.; Hao, J.; Ma, J.; Huang, L. A new approach to intrusion detection using Artificial Neural Networks and fuzzy clustering. *Expert Syst. Appl.* **2010**, *37*, 6225–6232. [CrossRef]
11. Govindarajan, M.; Chandrasekaran, R. Intrusion detection using neural based hybrid classification methods. *Comput. Netw.* **2011**, *55*, 1662–1671. [CrossRef]
12. Chung, Y.Y.; Wahid, N. A hybrid network intrusion detection system using simplified swarm optimization (SSO). *Appl. Soft Comput.* **2012**, *12*, 3014–3022. [CrossRef]
13. Elbasiony, R.M.; Sallam, E.A.; Eltobely, T.E.; Fahmy, M.M. A hybrid network intrusion detection framework based on random forests and weighted k-means. *Ain Shams Eng. J.* **2013**, *4*, 753–762. [CrossRef]
14. Kim, G.; Lee, S.; Kim, S. A novel hybrid intrusion detection method integrating anomaly detection with misuse detection. *Expert Syst. Appl.* **2014**, *41*, 1690–1700. [CrossRef]
15. Lin, W.C.; Ke, S.W.; Tsai, C.F. CANN: An intrusion detection system based on combining cluster centers and nearest neighbors. *Knowl. Syst.* **2015**, *78*, 13–21. [CrossRef]
16. Aslahi-Shahri, B.; Rahmani, R.; Chizari, M.; Maralani, A.; Eslami, M.; Golkar, M.; Ebrahimi, A. A hybrid method consisting of GA and SVM for intrusion detection system. *Neural Comput. Appl.* **2016**, *27*, 1669–1676. [CrossRef]
17. Kevric, J.; Jukic, S.; Subasi, A. An effective combining classifier approach using tree algorithms for network intrusion detection. *Neural Comput. Appl.* **2017**, *28*, 1051–1058. [CrossRef]
18. Al-Yaseen, W.L.; Othman, Z.A.; Nazri, M.Z.A. Multi-level hybrid support vector machine and extreme learning machine based on modified K-means for intrusion detection system. *Expert Syst. Appl.* **2017**, *67*, 296–303. [CrossRef]
19. Ahmim, A.; Derdour, M.; Ferrag, M.A. An intrusion detection system based on combining probability predictions of a tree of classifiers. *Int. J. Commun. Syst.* **2018**, *31*, e3547. [CrossRef]

20. 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]
21. Ferrag, M.A.; Maglaras, L. DeepCoin: A novel deep learning and blockchain-based energy exchange framework for smart grids. *IEEE Trans. Eng. Manag.* **2019**, 1–13. [CrossRef]
22. Derhab, A.; Guerroumi, M.; Gumaei, A.; Maglaras, L.; Ferrag, M.A.; Mukherjee, M.; Khan, F.A. Blockchain and random subspace learning-based IDS for SDN-enabled industrial IoT security. *Sensors* **2019**, *19*, 3119. [CrossRef] [PubMed]
23. Ferrag, M.A.; Maglaras, L. DeliveryCoin: An IDS and Blockchain-Based Delivery Framework for Drone-Delivered Services. *Computers* **2019**, *8*, 58. [CrossRef]
24. Sharafaldin, I.; Lashkari, A.H.; Ghorbani, A.A. Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization. In Proceedings of the ICISSP, Funchal, Portugal, 22–24 January 2018; pp. 108–116.
25. Ferrag, M.A.; Derdour, M.; Mukherjee, M.; Derhab, A.; Maglaras, L.; Janicke, H. Blockchain technologies for the internet of things: Research issues and challenges. *IEEE Int. Thing. J.* **2018**, *6*, 2188–2204. [CrossRef]
26. Ferrag, M.A.; Maglaras, L.; Janicke, H.; Smith, R. Deep Learning Techniques for Cyber Security Intrusion Detection: A Detailed Analysis. In Proceedings of the 6th International Symposium for ICS & SCADA Cyber Security Research 2019, Athens, Greece, 10–12 September 2019; pp. 126–136.
27. Ferrag, M.A.; Maglaras, L.; Moschoyiannis, S.; Janicke, H. Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study. *J. Inf. Secur. Appl.* **2020**, *50*, 102419. [CrossRef]
28. De Gregorio, M.; Giordano, M. An experimental evaluation of weightless neural networks for multi-class classification. *Appl. Soft Comput.* **2018**, *72*, 338–354. [CrossRef]
29. Adnan, M.N.; Islam, M.Z. Forest PA: Constructing a decision forest by penalizing attributes used in previous trees. *Expert Syst. Appl.* **2017**, *89*, 389–403. [CrossRef]
30. Ibarguren, I.; Pérez, J.M.; Muguerza, J.; Gurrutxaga, I.; Arbelaitz, O. Coverage-based resampling: Building robust consolidated decision trees. *Knowl. Syst.* **2015**, *79*, 51–67. [CrossRef]
31. Chang, C.C.; Lin, C.J. LIBSVM: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.* **2011**, *2*, 27. [CrossRef]
32. Huehn, J.C.; Huellermeier, E. FURIA: An Algorithm for Unordered Fuzzy Rule Induction. *Data Min. Knowl. Discov.* **2009**, *19*, 293–319. [CrossRef]
33. Kang, P.; Cho, S. EUS SVMs: Ensemble of under-sampled SVMs for data imbalance problems. In Proceedings of the International Conference on Neural Information Processing, Hong Kong, China, 3–6 October 2006; pp. 837–846.
34. Tsoumakas, G.; Katakis, I.; Vlahavas, I. Effective voting of heterogeneous classifiers. In Proceedings of the European Conference on Machine Learning, Pisa, Italy, 20–24 September 2004; pp. 465–476.
35. Folino, G.; Pisani, F.S. Evolving meta-ensemble of classifiers for handling incomplete and unbalanced datasets in the cyber security domain. *Appl. Soft Comput.* **2016**, *47*, 179–190. [CrossRef]
36. Frank, E.; Witten, I.H. Reduced-Error Pruning with Significance Tests. Available online: https://researchcommons.waikato.ac.nz/bitstream/handle/10289/1039/uow-cs-wp-1999-10.pdf?sequence=1&isAllowed=y (accessed on 29 February 2020).
37. Cohen, W.W. Fast Effective Rule Induction. In Proceedings of the Twelfth International Conference on Machine Learning, Tahoe City, CA, USA, 9–12 July 1995; pp. 115–123.
38. Bot-IoT Dataset. Available online: https://www.unsw.adfa.edu.au/unsw-canberra-cyber/cybersecurity/ADFA-NB15-Datasets/bot_iot.php (accessed on 30 May 2019).
39. Gharib, A.; Sharafaldin, I.; Lashkari, A.H.; Ghorbani, A.A. An evaluation framework for intrusion detection dataset. In Proceedings of the 2016 International Conference on Information Science and Security (ICISS), Pattaya, Thailand, 19–22 December 2016; pp. 1–6.
40. 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]