2019 15th International Conference on Distributed Computing in Sensor Systems (DCOSS)

A Novel Hierarchical Intrusion Detection System based on Decision Tree and Rules-based Models

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

***Abstract*—This paper proposes a novel intrusion detection system (IDS) that 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, attest their superiority in terms of accuracy, detection rate, false alarm rate and time overhead as compared to state of the art existing schemes.**

I. INTRODUCTION

In recent years cyber attacks, especially those targeting systems that keep or process sensitive information are becoming more sophisticated. Critical National Infrastructures are main targets of cyber attacks, since essential information or services depend on their systems and their protection becomes a signicant issue that is concerning both organizations and nations [1]. 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 (IDS) are deployed as a second line of defense. IDS based on some specific rules or patterns of normal behavior of the system can distinguish between normal and malicious actions [2].

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 or real time if they use contiguous monitoring of the system or periodic or off line if the detection happens in specific time instances or even off line using data that are collected and stored during a certain period of time. Moreover, when talking about Industrial Control Systems (ICS) that have specific requirements and characteristics novel taxonomies were recently proposed. Authors in [3] proposed a classification of IDSs ICS that divides them in three new categories: protocol analysis-based, traffic mining-based, and control process analysis-based.

Countermeasures are taken accordingly to the information obtained regarding the detected attacks from the detection systems. The better classification of the type of the attack is provided, the more efficient countermeasures will be chosen and the less those will 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 classication. 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 [4].

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

II. RELEVANT WORK

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

Aydin et al. [5] 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. Authors in [12] 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.

[1]A. Ahmim and M. Derdour are with Departement of Mathematics and Computer Science, University of Larbi Tebessi, Tebessa, Algeria `ahmed.ahmim@univ-tebessa.dz`, `makhlouf.derdour@univ-tebessa.dz`

[2]L. A. Maglaras and Helge Janicke are with School of Computer Science and Informatics, De Montfort University, Leicester, United Kingdom `leandros.maglaras@dmu.ac.uk`, `heljanic@dmu.ac.uk`

[3]M. A. Ferrag is with Department of Computer Science, Guelma University, B.P. 401, 24000, Algeria `ferrag.mohamedamine@univ-guelma.dz`

2325-2944/19/$31.00 ©2019 IEEE
DOI 10.1109/DCOSS.2019.00059

TABLE I
RELATED WORKS ON HYBRID INTRUSION DETECTION SYSTEMS

Year	Paper	Machine learning and data mining methods	Cyber approach	Data used
2009	Aydin et al. [5]	- Packet header anomaly detection - Network traffic anomaly detection	Hybrid IDS, which combing anomaly-based IDSs	IDEVAL
2010	Wang et al. [6]	- 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 [7]	- Multilayer perceptron neural network - Radial basis function neural network	Neural based hybrid IDS	UNM Send-Mail Data
2012	Chung and Wahid [8]	- Intelligent dynamic swarm - Simplified swarm optimization	Hybrid IDS	KDD CUP 1999
2013	Elbasiony et al. [9]	- Random forests algorithm - K-means clustering algorithm	Combining misuse and anomaly detection into a hybrid framework	KDD CUP 1999
2014	Kim et al. [10]	- C4.5 decision tree algorithm - Support vector machine model	Combining misuse and anomaly detection into a hybrid framework	NSL-KDD
2015	Lin et al. [11]	- k-Nearest Neighbor (k-NN) classifier	Combining cluster centers and nearest neighbors	KDD CUP 1999
2016	Aslahi-Shahri et al. [12]	- Support vector machine - Genetic algorithm	Hybrid IDS	KDD CUP 1999
2017	Kevric et al. [13]	- Random tree - C4.5 decision tree algorithm - NBTree	Combining classifier model based on tree-based algorithms	NSL-KDD
2017	Al-Yaseen et al. [14]	- Support vector machine - Extreme learning machine - K-means clustering algorithm	Hybrid IDS	KDD CUP 1999
2018	Ahmim et al. [15]	- 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. [16]	J48, Meta Pagging, RandomTree, REPTree, AdaBoostM1, DecisionStump, and NaiveBayes	Hybrid IDS, which combing anomaly-based IDSs	NSL-KDD
	Our work	- 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) [17]

The experimental results on the KDD CUP 1999 dataset show an outstanding true-positive value of 0.973 that comes with a 0.017 of false-positive value.

Wang et al. [6] 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 result 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 [7] 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.

Chung and Wahid [8] 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 intelligent dynamic swarm based rough set, shows a good overall performance with 93.3% accuracy in average of 20 runs.

Elbasiony et al. [9] 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. [10] integrates 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

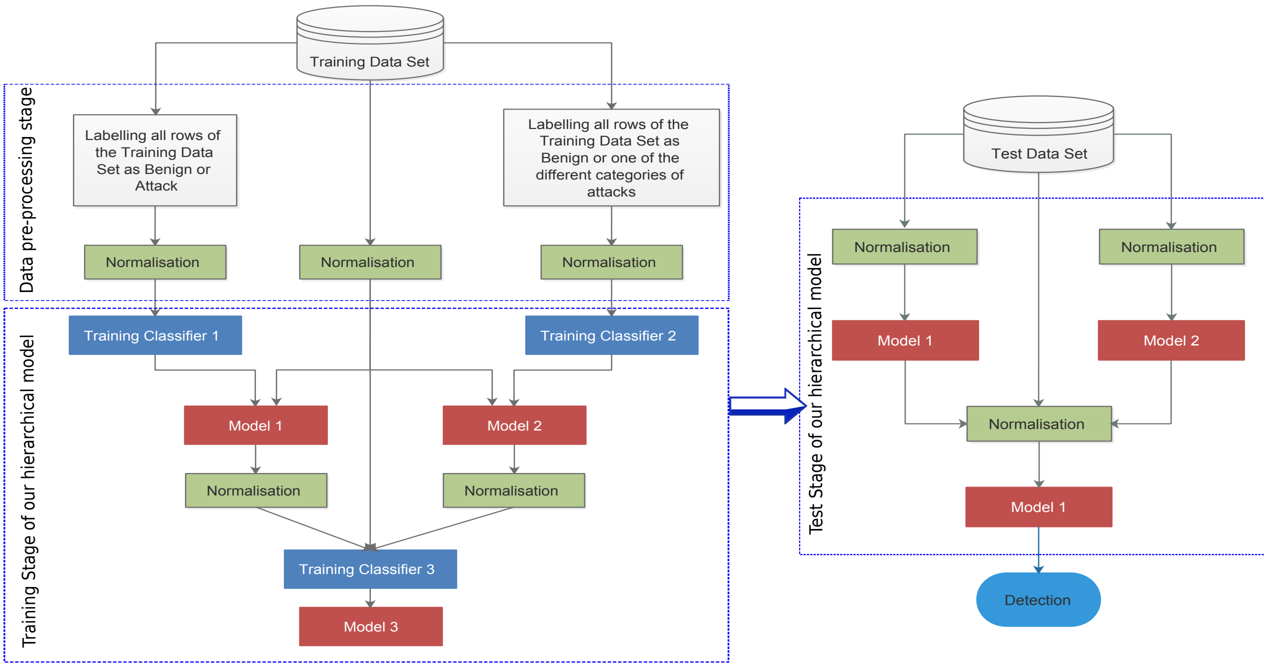

Fig. 1. General structure of our proposed model.

detection performance, training time, and testing time. Lin et al. [11] 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 [13], [14], [15], [16], [17]. In [13], Kevric et al. proposed a combining classifier approach using tree algorithms, namely, random tree, C4.5 decision tree algorithm, and NBTree. In [14], 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. [15] 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.

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, miss-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 [17], 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 [18], ForestPA [19], J48 Consolidated [20], LIBSVM [21], FURIA [22], RandomForest, REPTree, MLP, NaiveBayes, JRip and J48.

III. The Proposed Model

Based on the values of the features the classifier makes rules which are used in order to correctly classify the data set. 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 miss-classification is usually due to the classification of the 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.

Our hierarchical model aims to correctly classify each attack, provide 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.

A. Operation Mode

The operation mode of our proposed model is composed of two steps: training and testing.

1) Training Step: In this 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, where use initial labelling of rows where each specific attack is identified. 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.

2) Test Step: 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.

IV. 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.

A. Data set and Data pre-processing

In our experiments, we use CICIDS 2017 [17], which represents a data set that 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 [23].

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. Table II summarizes the distribution of different attack types and Benign rows.

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 II. 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 sub set, we select the first rows of each type. Then, for the test sub set, we select randomly some rows after the suppression of the training sub set rows.

TABLE II
COMPOSITION OF TRAINING AND TEST SUB-SETS

Label		Total	Total(-rows with lack info)	Training	Test
BENIGN	BENIGN	2273097	2271320	20000	20000
DOS	DDoS	128027	128025	2700	3300
	DoS slowloris	5796	5796	1350	1650
	DoS Slowhttptest	5499	5499	2171	1169
	DoS Hulk	231073	230124	4500	5500
	DoS GoldenEye	10293	10293	1300	700
	Heartbleed	11	11	5	5
PortScan	PortScan	158930	158804	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		471454	470365	20000	20000
Total		2830743	2827876	40000	40000

TABLE III
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)

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

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

TABLE IV
PERFORMANCE OF OUR MODEL AND OTHER CLASSIFIERS RELATIVE TO THE DIFFERENT ATTACK TYPE AND BENIGN

	Our Model	**WISARD [18]**	**Forest PA [19]**	**J48 Consolidated [20]**	**LIBSVM [21]**	**FURIA [22]**	**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%

TABLE V
OVERALL PERFORMANCE OF OUR MODEL AND OTHER CLASSIFIERS

	Our Model	**WISARD [18]**	**Forest PA [19]**	**J48 Consolidated [20]**	**LIBSVM [21]**	**FURIA [22]**	**Random Forest**	**REP Tree**	**MLP**	**Naive Bayes**	**Jrip**	**J48**
FAR	1.145%	2.865%	3.550%	6.645%	5.130%	3.165%	1.880%	4.835%	7.350%	33.455%	4.470%	5.040%
DR (Overall)	94.475%	48.175%	92.920%	92.020%	54.595%	90.500%	93.050%	91.640%	77.830%	82.510%	93.400%	91.990%
Accuracy	96.665%	72.655%	94.685%	92.688%	74.733%	93.668%	95.585%	93.403%	85.240%	74.528%	94.465%	93.475%
Training Time	159.5 s	13.47 s	110.64 s	105.46 s	318.6 s	234.98 s	20.03 s	2.73 s	942.98 s	0.45 s	76.65 s	8.34 s
Test Time	2.27 s	243.28 s	0.99 s	0.61 s	343.96 s	0.96 s	1.7 s	0.52 s	1.61 s	12.39 s	0.51 s	1.12 s

B. Performance metrics

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

To evaluate our model, 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.

$$DR_{AttackType} = \frac{TP_{AttackType}}{TP_{AttackType} + FN_{AttackType}} \tag{2}$$

$$TNR_{BENIGN} = \frac{TN_{BENIGN}}{TN_{BENIGN} + FP_{BENIGN}} \tag{3}$$

$$FAR = \frac{FP_{BENIGN}}{TN_{BENIGN} + FP_{BENIGN}} \tag{4}$$

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

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

C. Practical structure of our model

Our model 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 II as ”Attack” or ’Bening’. 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 [24] ; classifier 2 is Jrip [25]; classifier 3 is Forest PA [19].

D. Comparative Study

To evaluate our proposed model, we compare it with some well known classifiers and some recent ones namely J48, Jrip, Naive Bayes, MLP, REP Tree, Random Forest, FURIA [22], LIBSVM [21], J48 Consolidated [20], Forest PA [19], WISARD [18]. In this comparative study we use the different metrics detailed in sub section IV-B, in addition to training and testing time.
Table IV summarizes the performance of our model compared to the other classifiers for different attacks and Benign traffic. It shows that our hierarchical model 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 our hierarchical model is very close to the highest detection rate for two type of attacks namely FTP Patator with 99.636% and SSH Patator with 99.909%. For the rest of attacks type, our model gives an average performance compared to the other models. Overall, our model 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.

Table V summarizes the global performance of our model as compared to the other classifiers. As shown in table V our model gives the highest overall detection rate (DR Overall) with 94.475% , the highest accuracy with 96.665% , and lowest false alarm rate (FAR) with 1.145%. The training time of our model is 195.5 seconds and the test time is 2.27 seconds, which represents an acceptable training and test time for a hybrid hierarchical model especially when compared to simple models such as MLP and SVM.

V. 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' 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, our model gives the highest DR with 94.457%, the highest accuracy with 96.665%, and the lowest FAR with 1.145% while its low computational time makes it ideal for soft real time IDS systems.

REFERENCES

[1] L. A. Maglaras, K.-H. Kim, H. Janicke, M. A. Ferrag, S. Rallis, P. Fragkou, A. Maglaras, and T. J. Cruz, "Cyber security of critical infrastructures," *ICT Express*, vol. 4, no. 1, pp. 42–45, 2018.

[2] C. Alcaraz and S. Zeadally, "Critical infrastructure protection: Requirements and challenges for the 21st century," *International journal of critical infrastructure protection*, vol. 8, pp. 53–66, 2015.

[3] Y. Hu, A. Yang, H. Li, Y. Sun, and L. Sun, "A survey of intrusion detection on industrial control systems," *International Journal of Distributed Sensor Networks*, vol. 14, no. 8, p. 1550147718794615, 2018.

[4] T. Cruz, L. Rosa, J. Proença, L. Maglaras, M. Aubigny, L. Lev, J. Jiang, and P. Simoes, "A cybersecurity detection framework for supervisory control and data acquisition systems," *IEEE Transactions on Industrial Informatics*, vol. 12, no. 6, pp. 2236–2246, 2016.

[5] M. A. Aydın, A. H. Zaim, and K. G. Ceylan, "A hybrid intrusion detection system design for computer network security," *Computers & Electrical Engineering*, vol. 35, no. 3, pp. 517–526, 2009.

[6] G. Wang, J. Hao, J. Ma, and L. Huang, "A new approach to intrusion detection using artificial neural networks and fuzzy clustering," *Expert systems with applications*, vol. 37, no. 9, pp. 6225–6232, 2010.

[7] M. Govindarajan and R. Chandrasekaran, "Intrusion detection using neural based hybrid classification methods," *Computer networks*, vol. 55, no. 8, pp. 1662–1671, 2011.

[8] Y. Y. Chung and N. Wahid, "A hybrid network intrusion detection system using simplified swarm optimization (sso)," *Applied Soft Computing*, vol. 12, no. 9, pp. 3014–3022, 2012.

[9] R. M. Elbasiony, E. A. Sallam, T. E. Eltobely, and M. M. Fahmy, "A hybrid network intrusion detection framework based on random forests and weighted k-means," *Ain Shams Engineering Journal*, vol. 4, no. 4, pp. 753–762, 2013.

[10] G. Kim, S. Lee, and S. Kim, "A novel hybrid intrusion detection method integrating anomaly detection with misuse detection," *Expert Systems with Applications*, vol. 41, no. 4, pp. 1690–1700, 2014.

[11] W.-C. Lin, S.-W. Ke, and C.-F. Tsai, "Cann: An intrusion detection system based on combining cluster centers and nearest neighbors," *Knowledge-based systems*, vol. 78, pp. 13–21, 2015.

[12] B. Aslahi-Shahri, R. Rahmani, M. Chizari, A. Maralani, M. Eslami, M. Golkar, and A. Ebrahimi, "A hybrid method consisting of ga and svm for intrusion detection system," *Neural computing and applications*, vol. 27, no. 6, pp. 1669–1676, 2016.

[13] J. Kevric, S. Jukic, and A. Subasi, "An effective combining classifier approach using tree algorithms for network intrusion detection," *Neural Computing and Applications*, vol. 28, no. 1, pp. 1051–1058, 2017.

[14] W. L. Al-Yaseen, Z. A. Othman, and M. Z. A. Nazri, "Multi-level hybrid support vector machine and extreme learning machine based on modified k-means for intrusion detection system," *Expert Systems with Applications*, vol. 67, pp. 296–303, 2017.

[15] A. Ahmim, M. Derdour, and M. A. Ferrag, "An intrusion detection system based on combining probability predictions of a tree of classifiers," *International Journal of Communication Systems*, vol. 31, no. 9, p. e3547, 2018.

[16] S. Aljawarneh, M. Aldwairi, and M. B. Yassein, "Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model," *Journal of Computational Science*, vol. 25, pp. 152–160, 2018.

[17] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization." in *ICISSP*, 2018, pp. 108–116.

[18] M. De Gregorio and M. Giordano, "An experimental evaluation of weightless neural networks for multi-class classification," *Applied Soft Computing*, vol. 72, pp. 338–354, 2018.

[19] M. N. Adnan and M. Z. Islam, "Forest pa: Constructing a decision forest by penalizing attributes used in previous trees," *Expert Systems with Applications*, vol. 89, pp. 389–403, 2017.

[20] I. Ibarguren, J. M. Pérez, J. Muguerza, I. Gurrutxaga, and O. Arbelaitz, "Coverage-based resampling: Building robust consolidated decision trees," *Knowledge-Based Systems*, vol. 79, pp. 51–67, 2015.

[21] C.-C. Chang and C.-J. Lin, "Libsvm: a library for support vector machines," *ACM transactions on intelligent systems and technology (TIST)*, vol. 2, no. 3, p. 27, 2011.

[22] J. C. Huehn and E. Huellermeier, "Furia: An algorithm for unordered fuzzy rule induction," *Data Mining and Knowledge Discovery*, 2009.

[23] A. Gharib, I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "An evaluation framework for intrusion detection dataset," in *Information Science and Security (ICISS), 2016 International Conference on*. IEEE, 2016, pp. 1–6.

[24] E. Frank and I. H. Witten, "Reduced-error pruning with significance tests," 1999.

[25] W. W. Cohen, "Fast effective rule induction," in *Twelfth International Conference on Machine Learning*. Morgan Kaufmann, 1995, pp. 115–123.