A Behavior-Based Intrusion Detection System Using Ensemble Learning Techniques

Vincenzo Agate[1,*,†], Felice Maria D'Anna[1,†], Alessandra De Paola[1,†], Pierluca Ferraro[1,†], Giuseppe Lo Re[1,†] and Marco Morana[1,†]

[1]*University of Palermo, Department of Engineering, Viale delle Scienze, ed. 6, 90128 Palermo, Italy*

Abstract

Intrusion Detection Systems (IDSs) play a key role in modern ICT security. Attacks detected and reported by IDSs are often analyzed by administrators who are tasked with countering the attack and minimizing its damage. Consequently, it is important that the alerts generated by the IDS are as detailed as possible. In this paper, we present a multi-layered behavior-based IDS using ensemble learning techniques for the classification of network attacks. Three widely adopted and appreciated models, i.e., Decision Trees, Random Forests, and Artificial Neural Networks, have been chosen to build the ensemble. To reduce the system response time, our solution is designed to immediately filter out traffic detected as benign without further analysis, while suspicious events are investigated to achieve a more fine-grained classification. Experimental evaluation performed on the CIC-IDS2017 public dataset shows that the system is able to detect nine categories of attacks with high performances, according to all the considered metrics.

Keywords

Intrusion Detection, Ensemble Learning, Behavior-Based IDS

1. Introduction and Related Work

The growing availability of services offered by ICT systems through the Internet, and the consequent high amount of information accessible from anywhere, is increasingly tickling the interest of malicious users by offering unexpected opportunities for cyber attacks. Indeed, systems that provide fundamental services, considered critical for modern society, such as electricity [1, 2], transport, electronic voting, and telecommunications, need an adequate protection because, if damaged, could compromise the stability of an entire country [3, 4].

Intrusion Detection Systems (IDSs) deal with the detection and identification of cyber attacks, in order to promptly notify administrators when malicious traffic is discovered on the network, so as to limit any damage. Generally, the detection and mitigation of an attack take place in four phases [5]. In the first phase, network data are collected; then, the features that are

ITASEC'22: Italian Conference on Cybersecurity, June 20–23, 2022, Rome, Italy

*Corresponding author.

†These authors contributed equally.

vincenzo.agate@unipa.it (V. Agate); felicemaria.danna@unipa.it (F. M. D'Anna); alessandra.depaola@unipa.it (A. De Paola); pierluca.ferraro@unipa.it (P. Ferraro); giuseppe.lore@unipa.it (G. Lo Re); marco.morana@unipa.it (M. Morana)

0000-0002-3326-8500 (V. Agate); 0000-0002-7340-1847 (A. De Paola); 0000-0003-1574-1111 (P. Ferraro); 0000-0002-8217-2230 (G. Lo Re); 0000-0002-5963-6236 (M. Morana)

not necessary for the detection of malicious traffic are discarded, while the actual detection of malicious traffic is performed in the third phase. Finally, the fourth phase concerns the mitigation measures taken after the detection of the attack. This last phase is not always performed by an automated system, but can be carried out directly by an administrator. Two different types of IDSs exist, depending on whether they analyze network traffic (network-based IDSs - NIDSs) or the behavior of individual hosts (host-based IDSs - HIDSs) by monitoring the state of the system through the analysis of the log files.

The authors of [6] describe different types of IDSs and how they work to detect intrusions. It is possible to identify different classes of NIDSs, such as *signature based* (also called *misuse based detection*), or *behavior based* (also called *anomaly based detection*), depending on the way traffic is classified. Signature-based IDSs identify anomalous traffic based on a set of rules predefined as anomalous. One of the main disadvantages of this type of IDS is that it is not able to recognize new types of attack, or, in general, attacks that are not included in the set signatures it refers to, which is a well known problem of rule-based reasoning systems [7, 8]. Behavior-based IDSs aim to address this problem through the analysis of system operation by measuring the deviation of that operation from what is considered normal. Such systems would then be able to recognize an unprecedented attack (zero-day) if it affects the normal operation of the system in any way. Unfortunately, these systems generally underperform signature-based ones because of the difficulty in identifying the *normal operation* of the system.

Statistical techniques and Machine learning can be used to overcome this limitation and enable the design of systems that can automatically learn which events correspond to normal or abnormal behaviors. In [9], a statistical based IDS was proposed. This IDS is capable of identifying botnet network traffic by analyzing N-grams in HTTP traffic. This technique was based on the fact that C&Cs use similar communication patterns. In [10] a detection technique based on Multivariate Correlation Analysis was proposed. This system is capable of detecting both known and unknown DoS attacks, by learning normal traffic patterns. In [11], the authors use a statistical technique to simulate the human immune system. The proposed IDS is composed of two layers; the first identifies and classifies malicious traffic according to its type, while the second layer takes into account the traffic classified from the first layer as highly suspicious and identifies features that are relevant for intrusion detection. Although the datasets used to test the system contain multiple classes, the system only performs binary classification. Other systems exploit knowledge models such as ontologies [12] to efficiently classify among different classes of attack [13].

Machine and Deep Learning approaches, instead, allow the development of systems that recognize patterns in malicious traffic and learn how to detect such traffic, as shown in [14] and [15]. Many proposed IDSs use multiple machine learning algorithms to determine whether the captured traffic is benign or malicious. Some researchers use different machine learning algorithms to design hybrid systems [16], while others use ensemble learning techniques to design systems that have higher accuracy than individual weak learners. Traditional machine learning techniques are often used after careful and thorough feature selection. An example of a hybrid method is described in [17], where two types of Neural Networks are used: an instantaneous training Neural Network (CC4), and a Multi Layer Perceptron. The CC4 network works as an anomaly-based detection and is able to detect unknown attacks, whereas the Multi Layer Perceptron Neural Network works as a misuse based detection for known

attacks. However, this system can only identify an attack among four possible classes, besides benign traffic and unknown attacks. Other hybrid models combine Deep Neural Networks and Convolutional Neural Networks. For example, [18] proposes an intrusion detection system that uses a one-dimensional CNN to analyze feature sequences, whereas DNNs are used for high-dimensional feature vectors. In this case, the system proposed by the authors is capable of recognizing attacks among seven different classes.

Several IDSs exploit ensemble learning techniques for intrusion detection, as this allows for increased accuracy and classification performance over individual weak learners. For example, the authors of [19] propose the use of an ensemble learning technique in order to detect both known or new types of DDoS attacks. In this approach different base models focus on different aspects of intrusion. In [20], the authors use ensemble techniques to specifically detect botnet attacks against some protocols used in IoT networks. In particular, AdaBoost is used as an ensemble technique with three machine learning classifiers as weak learners: Decision Tree, Naive Bayes, and Artificial Neural Network. The system can detect only eight different types of botnet attacks. Many of the proposed systems only use a binary classification to identify anomalies. The lack of additional information about the specific type of attack often does not help administrators take appropriate countermeasures. In order to improve the decision making process, it may be useful to leverage a reputation system [21, 22] that takes into account the past behaviors of nodes in the network as proposed by [23].

Summarizing, multiclass IDSs in the literature generally recognize few attack classes, or are too specialized in the sense that they recognize only variants of the same attack. In contrast, our system achieves the right trade-off between number of recognized classes and prediction speed, as shown in the experimental results.

In this paper, we present a multi-layered Intrusion Detection System that uses machine learning and ensemble learning techniques to detect malicious traffic. The first layer provides a fast and efficient binary classification to recognize malicious traffic, while the second layer performs a more finely grained classification, identifying the attack as one of nine different types. In order to balance the dataset used and allow the system to correctly identify the classes represented by few samples, we propose the use of data augmentation techniques. The performances achieved by our system in terms of precision, accuracy, F1-score and FNR (false negative rate) are very high as shown by the extensive experimental evaluation.

The paper is organized as follows. Section 2 describes the proposed system and architecture. Section 3 explains details of the experiments and the results obtained from this study. Finally, Section 4 provides a conclusion of this work.

2. System overview

In this paper, we propose a novel multi-layered behavior-based IDS using ensemble learning that not only classifies traffic as benign or malicious, but also identifies traffic detected as malicious among nine possible categories of attacks.

The system architecture is composed of two layers and it is sketched in Figure 1. To prevent the system from being overloaded with all the network traffic, and consequently to prevent delayed detections, traffic filtering is preliminarily performed in order to distinguish “normal”

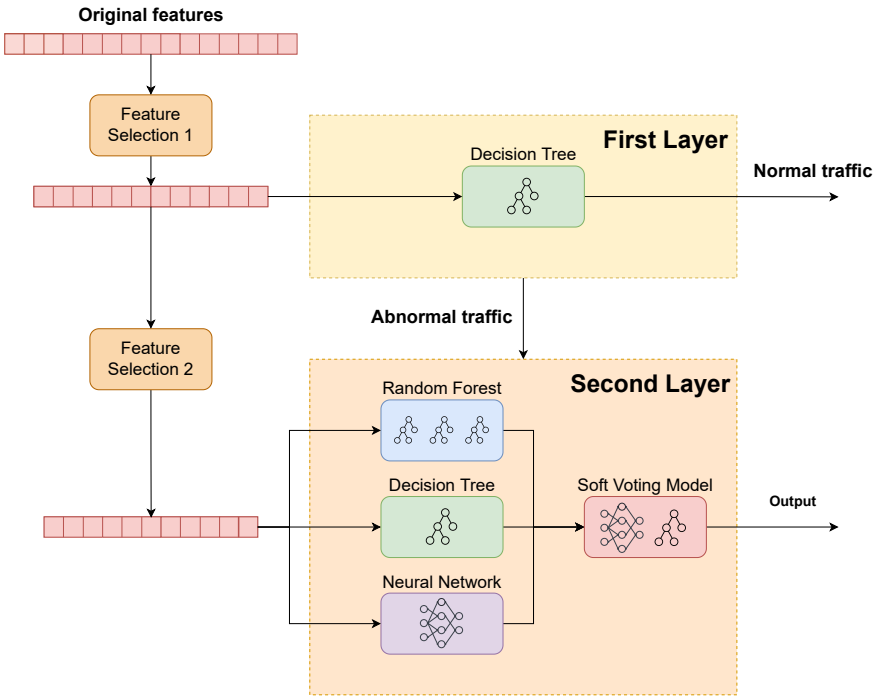

Figure 1: System architecture

or "abnormal" traffic and pass to the second layer only the latter. For the design of the first layer, we decided to adopt a Decision Tree (DT), since experimental evaluation showed its better performance for binary classification, compared to other methods such as Neural Networks, Random Forest, and Gaussian Naive Bayes.

The choice of this classifier is particularly critical because it heavily influences the behavior of the whole IDS. Indeed, traffic classified as benign by the first layer is considered safe and is not further analyzed by subsequent layers. Thus, false negatives at this stage could have very detrimental effects on the entire network.

In the second layer, an accurate analysis of malicious traffic is performed so that the system can generate an alert as detailed as possible. A detailed alert [24] would provide the necessary information to administrators to allow them to neutralize the attack quickly and efficiently.

In particular, our solution uses ensemble learning techniques. This approach involves the use of different learning models, called weak learners. The results of the predictions of the single models are aggregated using appropriate ensemble techniques that yield better classification performances than those of the single weak learners. The weak learners we considered are Neural Networks (NNs), Random Forests (RFs) and Decision Trees (DTs). All the weak learners take as input the same features. The Random Forest is an ensemble method that uses Bagging as technique of aggregation and uses Decision Trees as weak learners. Bagging is a technique of homogeneous ensemble learning that is used to train different models and to aggregate the result using some kind of averaging.

The ensemble technique used to aggregate our weak learners is the so-called voting technique. Voting-based classifier is a type of heterogeneous ensemble learning. This type of ensemble works as an extension of Bagging. The architecture of a voting-based classifier is composed by n homogeneous base models (in our case the Neural Network, the Random Forest and the Decision Tree) whose predictions are estimated in two different ways: hard and soft. In the hard voting mode, the logic of the majority is applied; consequently, the prediction considered correct by the ensemble will be the one that receives the highest number of votes from the weak learners. The soft voting mode considers, instead, the probability calculated from every base model that will be subsequently weighed. The prediction considered by the ensemble will be the one with the highest weighted probability.

One of the characteristics that makes the voting an optimal type of ensemble is its efficiency. Indeed, the full independence between the single base models allows the parallelization of the training phase of the weak learners, making both the training and prediction phases more efficient even in large scale scenarios. This feature is fundamental in systems that deal with anomaly detection, and in particular in intrusion detection systems, since even a slight delay in prediction could cause serious damage to devices connected to the network monitored by the IDS.

In this paper, we chose to use soft voting to merge the outputs of the individual classifiers, which are of different types. Specifically, the confidence values of the neural network prediction for each class is combined with the outputs of the Decision Tree and Random Forest. Future implementations may leverage more sophisticated approaches to assigning weight to classifiers, for example using reputation management systems [25, 26].

3. Experimental evaluation

To extensively evaluate our system, the CIC-IDS2017 dataset was used. The details and principles behind its realization are described in [27]. Using the CICFlowMeter software, publicly available on the Canadian Institute for Cybersecurity website, more than 80 features were extracted and all traffic was labeled. The dataset is composed by eight traffic monitoring sessions organized into CSV files. These files contain records with features representing benign traffic and malicious traffic divided into different types of attacks. The traffic collected in the dataset contains 15 types of traffic: one represents normal traffic, while the others are different types of attacks.

In order to make the attack category computation less expensive, some classes have been grouped together. This makes the classification less detailed, but at the same time provides better performance in terms of time in detecting and identifying the attack.

The dataset contains several attacks belonging to the Web Attack class, such as SQL-Injection, Brute Force, XSS, and also several attacks belonging to the DoS class, such as DoS GoldenEye, DoS Hulk, DoS Slowhttptest, DoS slowloris. These attacks have been grouped, respectively, in Web Attack and DoS, in order to make the computation lighter while ensuring a rather detailed and precise identification of the malicious event.

Although this grouping causes a loss in the detail of attacks identification, it still allows the administrator to distinguish their category. It would be possible to obtain a more detailed alert on the attack subcategories by using an additional level of classification. However, this would

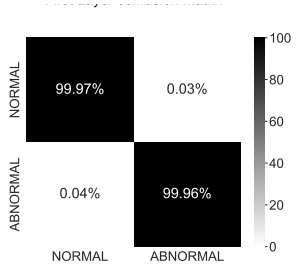

Figure 2
First layer confusion matrix

	Accuracy	Precision	Recall	F1-score	FNR
Normal	0.999685	0.999899	0.999709	0.999804	0.000291
Abnormal	0.999685	0.998813	0.999589	0.999201	0.000411

Table 1
First layer binary classifier metrics calculated on 200 runs

cause an increase in the computational cost of traffic detection and consequently an increase in system response time.

For the first classification layer, 80 relevant features of the dataset have been chosen, excluding the features containing data about source and destination IP addresses, timestamp, and flow ID. The latter contains a string that identifies the flow, consisting of the concatenation between the source and destination IP addresses, the source and destination ports, and the communication protocol. The exclusion of the above mentioned features allows the classifier to discern in the best way the malicious traffic from the benign one. In fact, considering source and destination IP addresses may lead the classifier to label a certain IP address as source or destination of malicious traffic, misclassifying further events, while the flow ID contains redundant information.

Several commonly adopted metrics have been used to evaluate our system, namely accuracy, precision, recall, F1-score; moreover, false negative rate was also considered because of the significant consequences an incorrect classification of malicious traffic could have. Various techniques for binary classification of traffic were implemented in several experiments. In order to perform a binary classification, all traffic in the dataset was relabeled using two labels: “Normal” and “Abnormal”. The best result for the first layer was obtained through the use of a Decision Tree.

In particular, after carrying out different implementations and having evaluated them using the main metrics, we choose to use a tree with maximum depth equal to 15. In fact, a further increase of the tree depth leads to an overfitting and consequently to a worsening of the main evaluation metrics. For the design of our IDS we have used as criterion for the evaluation of the quality of the split the technique of the information gain, that calculates the information acquired in relation to the feature used for the split of the current node of the Decision Tree.

The main results and the corresponding confusion matrix for the execution of the first layer of the system are reported in Table 1 and Figure 2, respectively.

For the training and testing phase of all models, we merged all eight traffic recording sessions, and then we randomly divided the resulting dataset into 70% for the training phase and 30% for the testing phase.

The nine classes considered are Bot, DDoS, DoS, FTP-patator, Heartbleed, Infiltration, Portscan, SSH-patator, Web Attack.

The analysis carried out by this layer takes into account a subset of the features considered in the previous layer. Among the various attack classes mentioned above, some are made up of

	Accuracy	Precision	Recall	F1-score	FNR
Bot	0.999944	0.993898	0.990111	0.991880	0.009889
DDoS	0.999801	0.999664	0.999466	0.999565	0.000534
DoS	0.999507	0.999159	0.999743	0.999451	0.000257
FTP-Patator	0.999905	0.997965	0.995329	0.996643	0.004671
Heartbleed	0.999997	0.999264	0.999998	0.999625	0.000002
Infiltration	0.999913	0.988887	0.990988	0.989471	0.009012
Portscan	0.999715	0.999665	0.999330	0.999497	0.000670
SSH-Patator	0.999708	0.980780	0.991668	0.986186	0.008332
Web Attack	0.999675	0.984755	0.931335	0.957053	0.068665

Table 2
Second layer Neural Network metrics

	Accuracy	Precision	Recall	F1-score	FNR
Bot	0.999993	0.999314	0.998796	0.999054	0.001204
DDoS	0.999981	0.999939	0.999978	0.999958	0.000022
DoS	0.999783	0.999759	0.999758	0.999759	0.000242
FTP-Patator	0.999992	0.999731	0.999696	0.999714	0.000304
Heartbleed	1.000000	0.999925	0.999962	0.999943	0.000038
Infiltration	0.999983	0.998343	0.997208	0.997773	0.002792
Portscan	0.999853	0.999735	0.999747	0.999741	0.000253
SSH-Patator	0.999966	0.998184	0.998545	0.998363	0.001455
Web Attack	0.999884	0.986143	0.983955	0.985034	0.016045

Table 3
Second layer Decision Tree metrics

a small number of samples. In particular, the Heartbleed and the Infiltration classes consist of 11 and 36 samples, respectively. Such a small number of samples, representing an insignificant amount compared to other classes containing hundreds of thousands of samples, does not allow the models to learn the patterns that distinguish them from other attacks.

Consequently, many samples belonging to such classes are not correctly classified by the models that constitute the second layer, thus causing a great loss of performance of the whole system.

In order to obtain a correct classification of such classes and a consequent increase of models performances, we used some techniques of data augmentation. These techniques augment the samples in the dataset by generating synthetic records from real data. In this work, we used the technique called ADASYN (Adaptive synthetic) sampling, presented in [28]. This technique implements a methodology that detects samples of minority class that are in a space dominated by the class with the largest number of elements.

In this way, synthetic samples are generated in areas of low density. This allows for more accurate classification of minority class elements, even in a region dominated by elements belonging to the most populated class.

After a careful evaluation and several experiments aimed to choose the best parameters, we

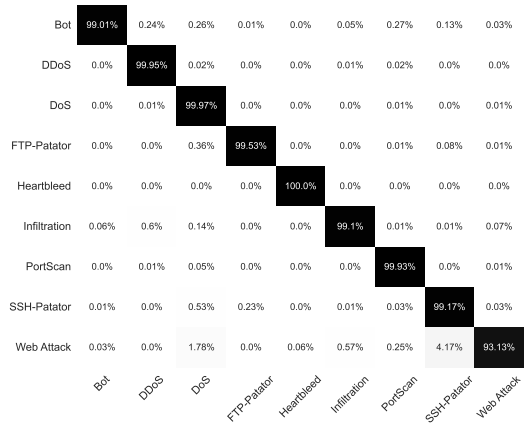

Figure 3: Neural network confusion matrix

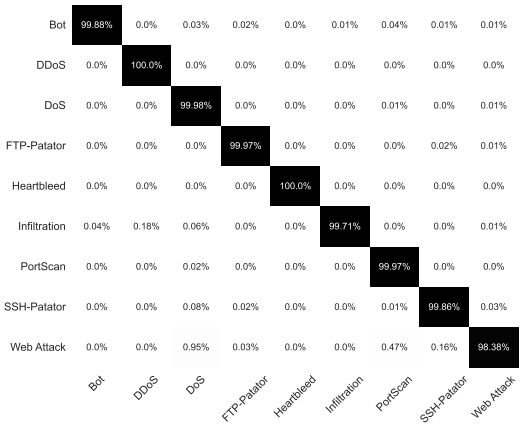

Figure 4: Decision tree confusion matrix

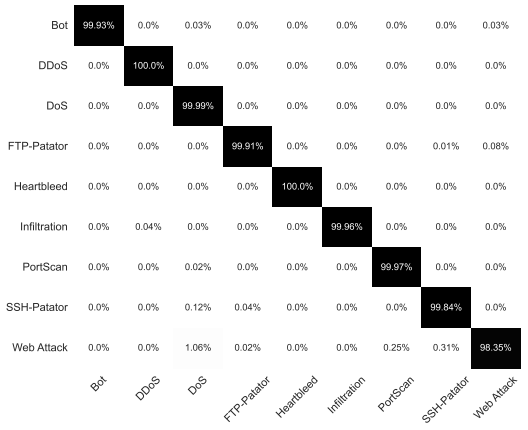

Figure 5: Random Forest confusion matrix

Figure 6: Ensemble confusion matrix

adopted a Neural network constituted by three layers. The input layer is constituted by 128 neurons that use the Rectified Linear Unit as activation function. The first hidden layer consists of 64 neurons, with an activation function which is the same as that of the previous layer.

Then, a further layer, called Dropout layer, randomly resets a percentage of the inputs, in order to prevent overfitting of the model (in particular, the percentage of inputs that are reset is 30%). Finally, an output layer consisting of 9 neurons (one for each class) was used. In this case, we chose a softmax activation function that allows the Neural Network to obtain the probability of belonging to one of the classes.

The performances obtained by this model are reported in Table 2 and Figure 3. The model achieves high performance in terms of precision and recall. The lowest recall is obtained by the Web Attack class, but the value achieved is still above 93%. This means that a small percentage of these attacks are misclassified by the classifier. Specifically, as can be seen from the confusion matrix, about 4% of the entire Web Attack traffic is mistaken by the classifier for an SSH-Patator attack while about 1.7% is mistaken for a DoS attack. These classification errors are corrected by

	Accuracy	Precision	Recall	F1-score	FNR
Bot	0.999997	0.999812	0.999348	0.999580	0.000652
DDoS	0.999984	0.999976	0.999954	0.999965	0.000046
DoS	0.999830	0.999718	0.999902	0.999810	0.000098
FTP-Patator	0.999983	0.999675	0.999116	0.999395	0.000884
Heartbleed	1.000000	1.000000	1.000000	1.000000	0.000000
Infiltration	0.999997	0.999637	0.999594	0.999615	0.000406
Portscan	0.999894	0.999910	0.999716	0.999813	0.000284
SSH-Patator	0.999958	0.997543	0.998437	0.997989	0.001563
Web Attack	0.999894	0.989248	0.983579	0.986393	0.016421

Table 4
Second layer Random Forest metrics

	Accuracy	Precision	Recall	F1-score	FNR
Bot	0.999997	0.999803	0.999331	0.999566	0.000669
DDoS	0.999991	0.999972	0.999987	0.999979	0.000013
DoS	0.999835	0.999737	0.999896	0.999816	0.000104
FTP-Patator	0.999996	0.999900	0.999835	0.999867	0.000165
Heartbleed	1.000000	0.999992	1.000000	0.999996	0.000000
Infiltration	0.999991	0.999096	0.998604	0.998848	0.001396
Portscan	0.999894	0.999926	0.999699	0.999812	0.000301
SSH-Patator	0.999958	0.997135	0.998926	0.998029	0.001074
Web Attack	0.999902	0.990538	0.984217	0.987356	0.015783

Table 5
Ensemble model metrics

the ensemble. Another classifier that performs well is the Decision Tree. Several experiments carried out by varying the parameters of the Decision Tree have shown good performance with a maximum depth of 50 and information gain as split criterion.

An increase in the maximum depth of the tree leads to an overfitting of the model and consequently to worse model performances. The performance achieved from this model is listed in Table 3 and Figure 4. The results show that the system achieves very high performance.

The lowest accuracy achieved is relative to the class of DoS attacks, which still maintains a percentage of 99.97%. The lowest values for precision and recall are related to the Web Attack class (more than 98%). This means that about 1% of the Web Attack traffic is mistaken by the classifier for DoS traffic.

We chose a Random Forest as the last classifier. The model used consists of 100 estimators, each with a maximum depth of 50, and information gain as split criterion. The results obtained by this model are shown in Table 4 and Figure 5. Even in this case, the performances are high. The lowest values for recall and precision are obtained for Web Attack class, but are still higher than 98%. Again, about 1% of the Web Attack traffic is mistaken by the classifier for a DoS attack. Finally, we present the results obtained by the entire ensemble after the soft voting phase.

The main metrics for the evaluation of the ensemble are shown in Figure 6 and Table 5. Since

	First Layer	Second Layer				Total System
	DT	DT	RF	NN	Ensemble	
Avg Time	5.56e-05s	1.87e-05s	0.00445s	0.016713s	0.0268s	0.0306s

Table 6
System time performances for anomaly prediction

all the classifiers that make up the ensemble mistake a small percentage of the Web Attack traffic for a DoS attack, even in the ensemble about 1% of the Web Attack traffic is mistaken by the classifier for a DoS attack.

For the evaluation of the system, all the models that constitute the proposed IDS have been run 1000 times using different train and test sets at every execution. The results have been averaged and are shown in the tables and figures. All tests have been performed on off-the-shelf laptops equipped with Intel 3805U 1.9GHz CPU and 4GB RAM.

The execution times of the systems were estimated by averaging the prediction times of 10000 randomly chosen samples from the test set. The results of the execution times are shown in Table 6. The execution times of the entire system (both the first and second layers) have also been reported in this table. As can be seen, the total execution time is low, so the proposed system is suitable for real-time use.

4. Conclusions

In this work, we proposed a multi-layered behavior-based IDS to recognize malicious traffic and classify it into one of nine possible attack classes. The first of the two layers of the proposed system allows for preemptive traffic filtering through a very fast binary classification, which makes the whole IDS more efficient. Since malicious traffic generally constitutes a small percentage of the total traffic, this initial filtering greatly reduces the events analyzed by the second layer, improving the overall response time.

Numerous tests performed on the system have demonstrated its reliability and accuracy in detecting malicious traffic, as well as its time efficiency. Our IDS is able to recognize and identify 9 different types of attack in real-time, promptly alerting administrators to minimize serious consequences. Future implementations may provide an additional layer of classification that more finely identifies the type of DoS or Web attacks. However, this would require a further computational cost and, consequently, a delay in the identification of the attack. It is always necessary to make a trade-off between the granularity of the classification and the computational cost. A possible solution could be to immediately send an alert if one of the grouped attack categories is recognized, and then subsequently perform a more detailed analysis, performing a two-stage classification.

Acknowledgments

This work is partially funded by the European Union - FESR o FSE, PON Ricerca e Innovazione 2014-2020 - DM 1062/2021.

References

[1] A. Timilsina, A. R. Khamesi, V. Agate, S. Silvestri, A reinforcement learning approach for user preference-aware energy sharing systems, IEEE Transactions on Green Communications and Networking (2021).

[2] V. Agate, A. R. Khamesi, S. Silvestri, S. Gaglio, Enabling peer-to-peer user-preference-aware energy sharing through reinforcement learning, in: ICC 2020 - 2020 IEEE International Conference on Communications (ICC), 2020.

[3] K. Thakur, M. L. Ali, N. Jiang, M. Qiu, Impact of cyber-attacks on critical infrastructure, in: 2016 IEEE 2nd International Conference on Big Data Security on Cloud (BigDataSecurity), 2016, pp. 183–186. doi:10.1109/BigDataSecurity-HPSC-IDS.2016.22.

[4] V. Agate, A. De Paola, P. Ferraro, G. Lo Re, M. Morana, Secureballot: A secure open source e-voting system, Journal of Network and Computer Applications 191 (2021).

[5] A. Khraisat, I. Gondal, P. Vamplew, J. Kamruzzaman, Survey of intrusion detection systems: techniques, datasets and challenges, Cybersecurity 2 (2019) 1–22.

[6] H.-J. Liao, C.-H. Richard Lin, Y.-C. Lin, K.-Y. Tung, Intrusion detection system: A comprehensive review, Journal of Network and Computer Applications 36 (2013) 16–24. URL: https://www.sciencedirect.com/science/article/pii/S1084804512001944. doi:https://doi.org/10.1016/j.jnca.2012.09.004.

[7] A. De Paola, P. Ferraro, S. Gaglio, G. L. Re, Autonomic behaviors in an ambient intelligence system, in: 2014 IEEE Symposium on Computational Intelligence for Human-like Intelligence (CIHLI), IEEE, 2014, pp. 1–8.

[8] V. Agate, P. Ferraro, S. Gaglio, A cognitive architecture for ambient intelligence systems (????).

[9] R. Tyagi, T. Paul, B. S. Manoj, T. B., A novel http botnet traffic detection method, in: 2015 Annual IEEE India Conference (INDICON), 2015, pp. 1–6. doi:10.1109/INDICON.2015.7443675.

[10] Z. Tan, A. Jamdagni, X. He, P. Nanda, R. P. Liu, A system for denial-of-service attack detection based on multivariate correlation analysis, IEEE Transactions on Parallel and Distributed Systems 25 (2014) 447–456. doi:10.1109/TPDS.2013.146.

[11] I. Dutt, S. Borah, I. K. Maitra, Immune system based intrusion detection system (isids): A proposed model, IEEE Access 8 (2020) 34929–34941. doi:10.1109/ACCESS.2020.2973608.

[12] P. Ferraro, G. Lo Re, Designing ontology-driven recommender systems for tourism, in: Advances onto the Internet of Things, Springer, 2014, pp. 339–352.

[13] M. Sarnovsky, J. Paralic, Hierarchical intrusion detection using machine learning and knowledge model, Symmetry 12 (2020) 203.

[14] S.-W. Lee, H. Mohammed sidqi, M. Mohammadi, S. Rashidi, A. M. Rahmani, M. Masdari, M. Hosseinzadeh, Towards secure intrusion detection systems using deep learning techniques: Comprehensive analysis and review, Journal of Network and Computer Applications 187 (2021) 103111. URL: https://www.sciencedirect.com/science/article/pii/S1084804521001314. doi:https://doi.org/10.1016/j.jnca.2021.103111.

[15] J. Lansky, S. Ali, M. Mohammadi, M. K. Majeed, S. H. T. Karim, S. Rashidi, M. Hosseinzadeh, A. M. Rahmani, Deep learning-based intrusion detection systems: A systematic review,

IEEE Access 9 (2021) 101574–101599. doi:10.1109/ACCESS.2021.3097247.

[16] V. Agate, F. Concone, P. Ferraro, A resilient smart architecture for road surface condition monitoring, in: The Proceedings of the International Conference on Smart City Applications, Springer, 2021, pp. 199–209.

[17] G. Mylavarapu, J. Thomas, A. K. TK, Real-time hybrid intrusion detection system using apache storm, in: 2015 IEEE 7th International Symposium on Cyberspace Safety and Security, 2015, pp. 1436–1441. doi:10.1109/HPCC-CSS-ICESS.2015.241.

[18] C. Ma, X. Du, L. Cao, Analysis of multi-types of flow features based on hybrid neural network for improving network anomaly detection, IEEE Access 7 (2019) 148363–148380. doi:10.1109/ACCESS.2019.2946708.

[19] S. Das, A. M. Mahfouz, D. Venugopal, S. Shiva, Ddos intrusion detection through machine learning ensemble, in: 2019 IEEE 19th International Conference on Software Quality, Reliability and Security Companion (QRS-C), 2019, pp. 471–477. doi:10.1109/QRS-C.2019.00090.

[20] N. Moustafa, B. Turnbull, K.-K. R. Choo, An ensemble intrusion detection technique based on proposed statistical flow features for protecting network traffic of internet of things, IEEE Internet of Things Journal 6 (2019) 4815–4830. doi:10.1109/JIOT.2018.2871719.

[21] V. Agate, A. De Paola, G. Lo Re, M. Morana, Vulnerability Evaluation of Distributed Reputation Management Systems, in: InfQ 2016 - New Frontiers in Quantitative Methods in Informatics, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), ICST, Brussels, Belgium, Belgium, 2016, pp. 1–8.

[22] V. Agate, A. De Paola, S. Gaglio, G. Lo Re, M. Morana, A framework for parallel assessment of reputation management systems, in: Proceedings of the 17th International Conference on Computer Systems and Technologies 2016, 2016, pp. 121–128.

[23] K. Gerrigagoitia, R. Uribeetxeberria, U. Zurutuza, I. Arenaza, Reputation-based intrusion detection system for wireless sensor networks, in: 2012 Complexity in Engineering (COMPENG). Proceedings, 2012, pp. 1–5. doi:10.1109/CompEng.2012.6242969.

[24] A. De Paola, P. Ferraro, S. Gaglio, G. Lo Re, M. Morana, M. Ortolani, D. Peri, A context-aware system for ambient assisted living, in: International Conference on Ubiquitous Computing and Ambient Intelligence, Springer, 2017, pp. 426–438.

[25] V. Agate, A. D. Paola, G. Lo Re, M. Morana, A simulation software for the evaluation of vulnerabilities in reputation management systems, ACM Transactions on Computer Systems (TOCS) 37 (2021) 1–30.

[26] V. Agate, A. De Paola, G. L. Re, M. Morana, A platform for the evaluation of distributed reputation algorithms, in: 2018 IEEE/ACM 22nd International Symposium on Distributed Simulation and Real Time Applications (DS-RT), IEEE, 2018, pp. 1–8.

[27] I. Sharafaldin, A. H. Lashkari, A. A. Ghorbani, Toward generating a new intrusion detection dataset and intrusion traffic characterization. (2018).

[28] H. He, Y. Bai, E. A. Garcia, S. Li, Adasyn: Adaptive synthetic sampling approach for imbalanced learning, in: 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), 2008, pp. 1322–1328. doi:10.1109/IJCNN.2008.4633969.