An NIDS for Known and Zero–Day Anomalies

A. Hussain*, F. Aguiló–Gost*, E. Simó–Mezquita*, E. Marín–Tordera*, X. Massip*

*Advanced Network Architectures Lab (CRAAX), UPC BarcelonaTech
Vilanova 08800, Spain

***Abstract*—Rapid development in the network infrastructure has resulted in sophisticated attacks which are hard to detect using typical network intrusion detection systems (NIDS). There is a strong need for efficient NIDS to detect these known attacks along with ever-emerging zero-day exploits. Existing NIDS are more focused on detecting known attacks using supervised machine learning approaches, achieving better performance for known attacks but poor detection of unknown attacks. Many NIDS have utilized the unsupervised approach, which results in better detection of unknown anomalies. In this paper, we proposed a Hybrid NIDS based on Semisupervised One-Class Support Vector Machine (OC-SVM) and Supervised Random Forest (RF) algorithms. This detection system has several stages. The First stage is based on OC-SVM, which filters benign and malicious traffic. The next stages use many parallel supervised models and an additional OC–SVM model to separate known and unknown attacks from malicious traffic. The previous process is done so that known attacks are classified by their type, and unknown attacks are detected. The proposed NIDS is tested on the standard public dataset CSE-CIC-IDS-2018. The evaluation results show that the system achieves a high accuracy, 99.45%, for detecting known attacks. Our proposed NIDS achieves an accuracy of 93.99% for unknown or zero-day attacks. The overall accuracy of the proposed NIDS is 95.95%. The system significantly improves the detection of known and unknown anomalies using a hybrid approach.**

***Index Terms*—Cyber Security, Network Intrusion Detection System, Machine Learning, One–Class SVM, Random Forest.**

I. Introduction

Consulting the news about recent network cyberattacks that have occurred worldwide, we can verify how the range of targets of hackers varies, from government and academic institutions to private companies, manufacturers, etc. [1]. They have diverse objectives, seeking to harm in different ways ranging from spying, interrupting processes, and creating chaos to damaging reputations. Recognizing that attackers can be successful in their attempts causes a growing concern regarding cybersecurity in all areas that depend almost 100% on the Internet in their operations. Finding vulnerability gaps in the *Information Technology* (IT) systems should be a primary objective for most companies and manufacturers since having a weakness in the IT system could affect customer operations. No matter how small a company is, if it is part of a supply chain, not taking cybersecurity measures could affect the entire range of companies participating in the chain.

Among the protection methods are *Network Intrusion Detection Systems* (NIDS), tools for network protection, which aim to analyze network traffic to detect malicious traffic. NIDS become one of the first lines of defense [2]. In this research, a NIDS is proposed to detect anomalies in network traffic. This system is based on several supervised learning algorithms for detecting known attacks and unsupervised learning *One–Class Support Vector Machine* (OC–SVM) for detecting unknown attacks. The Network IDS (NIDS) developed in this paper has high accuracy in detecting known attacks and in reducing the rate of false positives, as well as detecting zero-day attacks, showing a whole accuracy of 96.0%.

It is worth mentioning that the above-described model in this paper is an improvement of our previous work [12]. We refer to this previous work as PNIDS. In summary, PNIDS was composed of two components, the supervised models as the first component and the second one the OC–SVM component, classifying between attack or benign entry. Despite the promising results in detecting known attacks, PIDS was not good enough when detecting benign entries. This fact came from two different causes: first, the supervised models in the first component detected many benign entries as known attacks; second, we consider that the OC–SVM hyperparameters can be improved.

Regarding the problem of detecting benign entries and compared with PNIDS, we tried to solve this weakness by changing the order of components and optimizing the OC–SVM hyperparameters. We refer to this architecture as NIDS-1, which is depicted in Fig 1. Although NIDS-1 proved to be better than PNIDS, we consider we consider, that there is room for more improvement and novelty with respect to PNIDS. Thus, we propose adding an additional unsupervised component OC–SVM to NIDS-1, which plays two prominent roles. First, it acts as a filter for those (few) misclassified entries under the action of the supervised component. Second, it works jointly with the previous OC–SVM component to catch those benign entries misclassified as anomalies. The proposed second architecture will be referred to as NIDS-2, as shown in Fig. 2.

This paper is organized as follows, Section 2 reviews some related works in the literature. Section 3 describes the NIDS system proposed in this paper, Section 4 depictes the implementation of the proposed NIDS, Section 5 presents and compares the results, and finally Section 6 concludes the paper and presents future work.

II. State of the art

There are multiple studies in the literature on NIDS based on ML techniques [3]. Several researchers have used supervised

 DOI 10.1109/DRCN57075.2023.10108319

machine learning techniques as the core of the IDS. In [4], authors present an IDS based on a *Multi-layer Perceptron Neural Network* for intrusion detection. Authors in [5] propose a Support Vector Machines-based IDS to detect routing layer attacks in an IoT system. In [6], the authors perform empirical experiments using four machine learning classifiers, *Random Forest*, *Decision Tree*, *Multi-layer Perceptron*, and *Support Vector Machine*, to test and evaluate the efficiency and performance of IDSs. The main criticism of IDS based on supervised learning is the non detection of zero-day attacks (unknown attacks). In addition, as labelled data is required, a large part of these works are based on public datasets that are not updated [7]. On the other hand, other studies rely on unsupervised algorithms to detect zero-day attacks. In [8], authors compare the performance and computational cost of classification models trained with unsupervised ML techniques: *Principal Components Analysis*, *Isolation Forest*, *One-Class SVM* and *Autoencoder* for the CIC-IDS-2017 dataset. Winter et al. [9] propose an optimized NIDS concerning its false alarm rate; this system uses OC–SVM model as an analysis engine that has been trained using malicious network data. Choi et al. [10] develop a NIDS using autoencoders as the analysis engine; this system verifies that its performance accuracy exceeds the one obtained by a NIDS from previous studies developed with cluster analysis algorithms. Authors in [11] propose a NIDS that consists of two classification stages. In the first stage, an unsupervised OC–SVM model is used, which has been trained with the malicious flow as the positive class to separate the threat flow from benign network traffic. In the second stage, a Self Organizing Map (unsupervised neural network) is used to group the malicious flow into different alert clusters. The main weakness of these reviewed NIDS is that unsupervised ML models are less accurate than those developed with supervised techniques. However, as previously remarked, supervised methods do not detect zero-day attacks.

In this work, we take advantage of both techniques. In the same NIDS, we combine several models trained by both types of learning, supervised algorithms for detecting known attacks and unsupervised algorithms for detecting unknown attacks, achieving a high global accuracy compared with previous works.

III. NIDS Proposed Architecture

Although two different architectures appear in this section, NIDS-1 and NIDS-2, we propose the second one as a novelty. NIDS-2 is the evolution of NIDS-1 and applies two different uses of the OC–SVM algorithm in two steps. The first step is the classical one which makes a binary classification. In the second, this algorithm filters the misclassified entries for proper re-classification.

We briefly describe the algorithms used in the proposed architectures. First we describe the unsupervised algorithm, OC-SVM (used in one or two steps of our architecture), and then the supervised algorithm, Random Forest (one step on both architectures).

The OC–SVM algorithm classifies data from a set with only two different classes, the *positive* and *negative* ones. For training, only one class is needed, the positive one[1]. Here we summarize how this algorithm works.

Let $\{\mathbf{x}_1, \ldots, \mathbf{x}_m\} \subset \mathbb{R}^n$ be the dataset composed by a given class of interest (the positive class). The *radial* kernel is defined by $K(\mathbf{x}, \mathbf{y}) = e^{-\gamma||\mathbf{x}-\mathbf{y}||^2}$ with $\gamma \in [0, +\infty)$. In this summary, there are other possible kernels to play with, so we describe the algorithm using a generic form, $K(\cdot, \cdot)$, for the kernel. Given $\nu \in (0, 1]$ and $\gamma \in [0, +\infty)$, we get a vector $\alpha = (\alpha_1, \ldots, \alpha_m) \in \mathbb{R}^m$ solving the non-linear optimization problem

$$\min_{\alpha} \ \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j)$$

constrained to $0 \leq \alpha_i \leq \frac{1}{\nu m}$ and $\sum_{i=1}^{m} \alpha_i = 1$. This problem allows us to select a set of vectors $\mathbf{x}_i$, for $i \in I \subset \{1, 2, \ldots, m\}$, with $\alpha_i \neq 0$ as a support vectors. Then, an offset ρ is computed by $\rho = \sum_{j=1}^{m} \alpha_j K(\mathbf{x}_i, \mathbf{x}_j)$, where α_i is taken among the values $\{\alpha_1, \ldots, \alpha_m\}$ not being an upper nor lower bound. Then, we decide whether a vector $\mathbf{x}$ belongs to the positive class or not depending on whether the function $f(\mathbf{x}) = \text{sign}\left(\sum_{j=1}^{m} \alpha_j K(\mathbf{x}, \mathbf{x}_j)\right)$ is positive or negative.

The computation of α and ρ corresponds to the training process. The validating step uses the resulting function $f(\mathbf{x})$. Obtaining good accuracy (or any other parameter of the resulting model) directly depends upon the values of γ and ν, the hyper-parameters of this algorithm.

Thus, optimizing accuracy corresponds to finding optimal values of both hyper-parameters, which is a time consuming task. This task has been proven to be critical when using this algorithm. We have searched for appropriate values of ν and γ, selecting the radial kernel for our dataset.

Our supervised step in the proposed architecture is based on a Random Forest algorithm. The Random Forest algorithm consists of multiple decision trees. Each tree is created with a randomly chosen subset of features and votes in parallel and independently for a class prediction. The final result is the most voted class after obtaining the predictions made by all the trees [16].

A. NIDS-1 Architecture

The NIDS-1 is shown in Fig. 1. The input data shown in the figure is a network flow, and the system consists of two components. While the first one performs a binary classification of system inputs, categorizing each network flow as benign or abnormal (anomaly), the second performs multiple classifications to detect known attacks. The whole NIDS-1 performance is based on Machine Learning. The detection engine of the first component is based on unsupervised learning, specifically an OC–SVM, and is intended to detect anomalous traffic. Then, when the output of the first component is classified as an anomaly, it is sent to the second component. This component

[1]For this reason this algorithm is said to be a semi-supervised algorithm. Nevertheless, many authors refers it as an unsupervised algorithm.

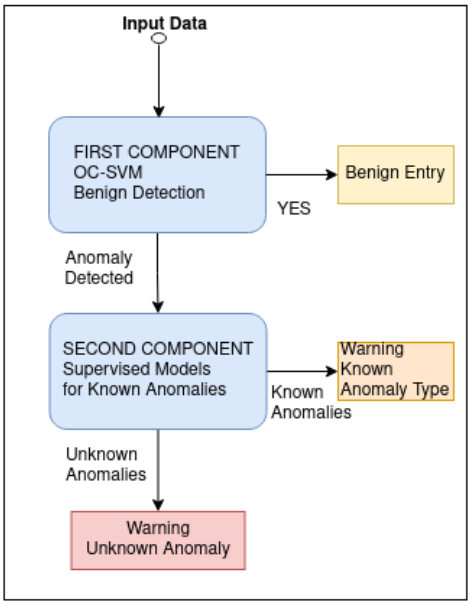

Fig. 1. NIDS-1 architecture

will try to classify the type of attack when the entry is a known attack. Otherwise, the entry is outputted as an unknown attack when the anomaly is unknown. In the case of a known attack, the detection engine is based on supervised learning and contains as many supervised models as the number of known attacks. The system has been developed so that each trained model detects a specific known attack.

These supervised models work in a parallel way to detect known attacks. The input network flows which enter in the second component are passed to all of these trained models. Each trained model tries to classify this input as an attack of a specific type of the nature for which it was trained. The prediction results of all models are stored in a list. Two checks are performed on these results. Firstly, the known attack having maximum accuracy is obtained. Secondly, the maximum accuracy for a particular known attack must be greater than a given threshold value used to classify this entry as an attacking entry. The success of these checks leads to the classification of a particular type of known attack. The failure of the second check suggests that this entry doesn't belong to any of the known attacks, so it should be an unknown attack or false anomaly detection by the first component. The Second component of the NIDS-1 may exploit the parallelism architecture of the hardware; thus, not adding overhead in the detection.

B. NIDS-2 Architecture

As commented before, an additional ML model is proposed to be included in NIDS-1 depicted in Fig. 1. Another OC–SVM is added after the second component. This addition gives the NIDS-2 architecture that is described in Fig. 2.

This second OC–SVM model helps in correcting some failures of the second check, supervised component, (as it has been commented in the previous section) in NIDS-1.

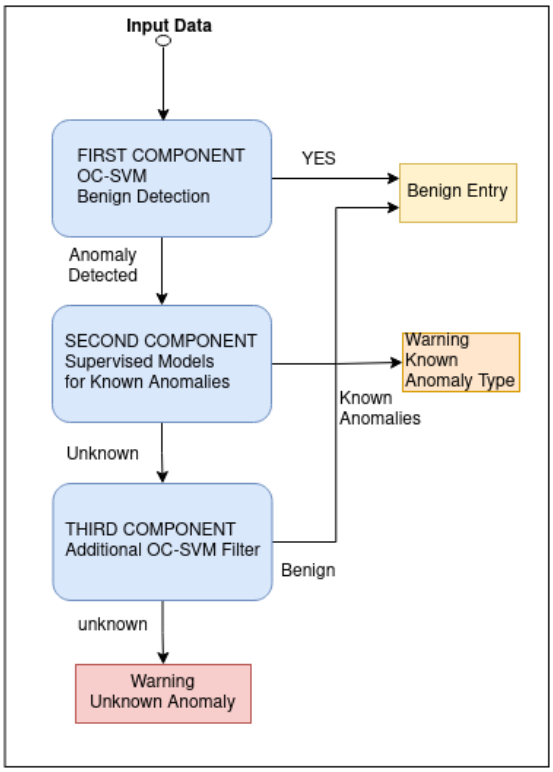

Fig. 2. NIDS-2 architecture

More precisely, when a benign entry has been detected as an unknown attack by the NIDS-1 architecture is passed to the second OC–SVM.

Thus, two OC–SVM different models are included in NIDS-2. The first model classifies entries between benign or anomaly. The second one filters false anomaly detections and corrects them.

IV. NIDS IMPLEMENTATION

We have used the CSE-CIC-IDS-2018 [13] dataset to implement the proposed NIDS. This dataset is a collaboration between the Communications Security Establishment (CSE) and the Canadian Institute for Cybersecurity (CIC). It was created to test and evaluate NIDSs. It includes 80 features of network traffic captured by CICFlowMeter-V3 from 7 different attack scenarios: Brute-force, Heartbleed, Botnet, DoS, DDoS, Web attacks, and network infiltration from inside. We carried out the implementation of the NIDS using the statistical software `R`. We use a Linux virtual box provided with 40 Athlon@2.5Gh processors and 125Gb Ram. `R` code is developed on a MacBook Pro, with a 2.8 GHz Intel Core i7 quad-core processor for debugging and then sent to the Linux box for processing.

Datasets need to be preprocessed. For instance, the Timestamp feature is removed because it was not considered relevant to the study. Entries with missing data were not considered. Finally, numerical features were properly scaled. The first selection of features applied was made through the Boruta algorithm, and 20 out of 80 were considered irrelevant.

A. Training

We used the OC-SVM algorithm to train the unsupervised component. We trained it with data of the same type (positive class) to learn its characteristics and made predictions about inputs that did not fit them. We used OC–SVM because it has advantages in high-dimensional spaces over other single-classifiers [14].

Table I shows the dataset for training and validating OC–SVM models. Notice that attacks of type Bot are not included in this dataset. These types of attacks are selected to act as the unknown attacks.

Type	# Entries
Benign	50000
DoS GoldenEye	3000
DoS Slowloris	1000
DDoS LOIC-http	4000
DDoS HOIC	4850
DDoS LOIC-udp	150
BF ftp	4000
BF ssh	4000

TABLE I
TRAINING AND VALIDATING DATA-SET FOR THE OC–SVM MODEL

Type	# Entries
Benign	1000
DoS GoldenEye	100
DoS Slowloris	100
DDoS LOIC-http	100
DDoS HOIC	100
DDoS LOIC-udp	100
BF ftp	100
BF ssh	100
Bot	200

TABLE II
TEST DATA–SET FOR THE NIDS

The test dataset for the IDS is shown in Table II. It contains 1000 benign entries and 900 anomalies (200 of them are unknown ones).

B. NIDS-1

As we mentioned in the NIDS architecture section, the proposed NIDS is composed of two components. The first component contains an OC-SVM model that has been trained using benign inputs as the positive class. More precisely, the OC–SVM model has been trained using 30% of data. The hyper-parameter optimization obtained the values $\nu = 0.0101$ and $\gamma = 1.1000$. The related accuracy is 90.4% on the validating set of data.

The second component classifies the anomalies between different types of attacks, including unknown ones. This component contains several monitored models.

Because the different types of attacks in CSE-CIC-IDS-2018 are not balanced, we decided to train a specific model for each type of attack. For this task, we considered binary sets that included data of its corresponding type of attack plus benign data. Table III shows the datasets used for training and validating these supervised models. We used 90% of the data to train and the remaining 10% to validate the models. The Proportion column shows the percentage of data taken from the original dataset.

Attack Type	# Benign	# Attack	Proportion (%)
FT-BF	63635	18370	7.8
SSH-BF	63635	17995	7.8
DoS GoldenEye	95042	3915	9.4
DoS SlowLoris	38016	1043	9.2
DDoS LOIC-http	92860	7140	1.3
DDoS HOIC	34535	65297	9.5
DDoSLOIC-udp	1726	168	0.2

TABLE III
TRAINING AND VALIDATING DATA-SET FOR SUPERVISED COMPONENT

We used the *mlr3* ecosystem of `R` [15] to train supervised models with the *Random Forest*, *Neural Network*, *XGBoost* and *SVM* algorithms. In each case, we made new feature selections by applying the package *mlr3fselect* to identify the significant characteristics of each type of attack. We compared their performances. Models trained using the Random Forest algorithm obtained the best scores.

As it is commented in [7], let us denote TP (true positive) as the number of anomalies correctly detected; TN (true negative) as the number of benign entries correctly detected; FP (false positive) as the number of benign entries detected as anomalies and FN as the number of anomalies detected as benign entries. Thus, standard performance indicators given by default in `R` are

- *Overall Accuracy* (Acc) is the probability that an entry is correctly classified between benign or anomaly.

$$\text{Acc} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

- *Precision* (Pre) is the ratio between detected anomalies that are truly anomalies.

$$\text{Pre} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

- *Recall* (Rec) is the true anomaly rate.

$$\text{Rec} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

Table IV presents the performance metrics of the supervised models included in the second component, and Table V shows the number of features that have been used to train the models.

Model	Accuracy	Precision	Recall
FT-BF	100	100	100
SSH-BF	99.99	100	99.98
DoS GoldenEye	99.99	100	99.94
DoS SlowLoris	99.96	99.89	98.94
DDoS LOIC-http	99.99	99.98	99.92
DDoS HOIC	100	100	100
DDoSLOIC-udp	100	100	100

TABLE IV
SUPERVISED COMPONENT: PERFORMANCE METRICS OF THE MODELS

Model	# Features
FT-BF	15
SSH-BF	50
DoS GoldenEye	31
DoS SlowLoris	43
DDoS LOIC-http	45
DDoS HOIC	19
DDoSLOIC-udp	19

TABLE V
SUPERVISED COMPONENT: # FEATURES OF THE MODELS

C. NIDS-2

We use the same dataset for training the new OC–SVM model we add to NIDS-1 to obtain NIDS-2. Supervised models are known to be very precise generically. Thus, most known attacks detections in the second component are correctly classified. However, non-known attacks outputs of this component will include some benign entries also. This is a main weakness in NIDS-1.

To reduce this weakness a second OC–SVM is added to NIDS-1 architecture. This new OC–SVM filters those benign entries badly forwarded to the second component by the first OC–SVM.

We decided to increase the percentage data to 50% for training this new OC–SVM. Now, the optimization of the hyper-parameter values gives $\nu = 0.07$ and $\gamma = 1.0$, using the same kernel. The resulting training accuracy is 92% over the validating data.

V. RESULTS

We use the standard notation for studying the goodness of a NIDS. Thus, we add two more indicators to those defined in the previous section

- *Fallout* (Fo) is the false anomaly rate.

$$\text{Fo} = \frac{\text{FP}}{\text{TN} + \text{FP}}$$

  A high fallout limits the performance of an IDS. Users tend to be non-confident with it.
- *Miss Rate* (Mr) is the false benign rate.

$$\text{Mr} = \frac{\text{FN}}{\text{TP} + \text{FN}}$$

Obviously, for the measure of the 'global' performance of a NIDS, one wants high values of Acc, Pre and Rec, and low values of Fo and Mr. We also include other 'local' measures for the different components of the proposed NIDS.

A. NIDS-1

After the training process, NIDS-1 has been tested using the previously described test dataset. We have obtained the following values of the basic global parameters TP $= 896$, TN $= 890$, FP $= 110$ and FN $= 4$. So, the related metrics are Acc $= 0.9400$, Pre $= 0.8907$, Rec $= 0.9956$, Fo $= 0.1100$ and Mr $= 0.0044$.

Although the accuracy of 94% is not bad, the precision should be over 90% and the false anomaly rate should be under 10%. These figures have to be improved. On the contrary, the false benign rate 0.44% is low, which is essential to be confident with the NIDS-1.

Focusing on some improvements of NIDS-1, let's see some local figures:

- Component C1:

  Detects 890 benign entries.
  Misdetects 4 false benigns.
  True attacks forwarded to C2 896.
  False attacks forwarded to C2 110.
- Component C2:

  696 true known attacks are detected correctly.
  All of them are correctly classified.
  5 false known attacks are misdetected.
- Unknown Attacks Detection:

  True Unknown Attacks detected 200.
  False Unknown Attacks misdetected 105.

These figures show some weak points: 4 false benigns are misdetected by the first component, 110 benign entries have been sent to the second component and 105 false unknown attacks are misdetected by the second component. That is, the OC-SVM model has misclassified some benign entries in the first component. These facts suggest us to filter these benign entries after they pass through the second component.

B. NIDS-2

The basic global parameters for NIDS-2, using the same test dataset, are TP $= 896$, TN $= 927$, FP $= 73$ and FN $= 4$. Clearly, the value of TN has been improved from 890 to 927 as we expected. Thus, this additional model acts like a filter that recovers some false attacks forwarded by the first component of the previous NIDS-1.

The global parameters of NIDS-2 are Acc $= 0.9595$, Pre $= 0.9247$, Rec $= 0.9956$, Fo $= 0.0730$ and Mr $= 0.0044$. Now, accuracy and precision have been significantly improved. Moreover, the true anomaly rate is almost full, the false anomaly rate is under 10%, and the false benign rate is very low.

As it has been done before, we include here additional local figures of NIDS-2:

- Components C1 and C3 (both OC-SVM models):

  Detects 927 benign entries.
  Missdetects 4 false benigns.
- Component C1:

  True attacks forwarded to C2 896.
  False attacks forwarded to C2 73.
- Component C2:

  696 true known attacks are detected correctly.
  All of them are correctly classified.
  5 false known attacks are misdetected.
- Unknown Attacks Detection:

  True Unknown Attacks detected 200.
  False Unknown Attacks detected 68.

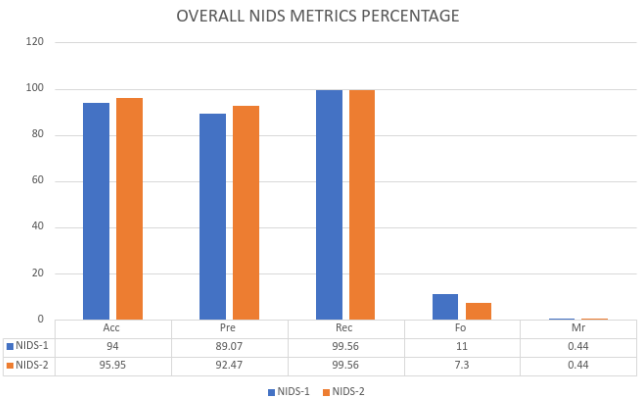

Fig. 3. Overall NIDS' metrics percentages

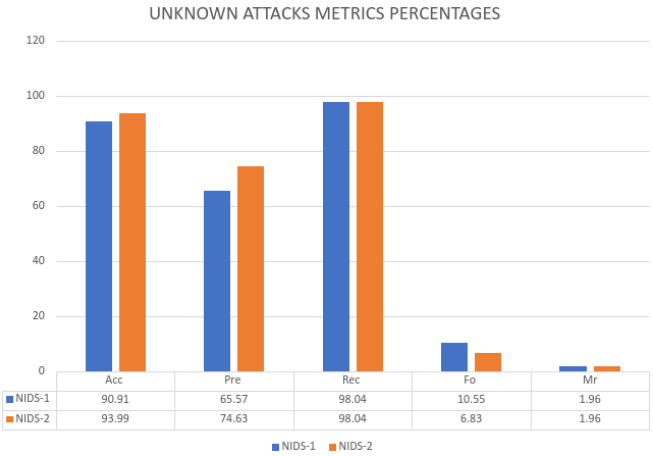

Fig. 5. Unknown attacks metrics percentages

VI. Performance Evaluation

The overall metrics of NIDS-1 and NIDS-2 are included in Figure 3. The recall and miss rate are the same because the overall parameters TP and FN remains unchanged in both architectures. Values of TN and FN have been improved in NIDS-2, thus accuracy, precision and fallout have been improved also. The previous section shows that classification between different known attacks has been correctly done by the second component in both architectures. This fact is a consequence of the generic good behaviour of supervised models included in this component.

Now we comment the local metrics, that is metrics related to either known-only attacks or unknown-only ones.

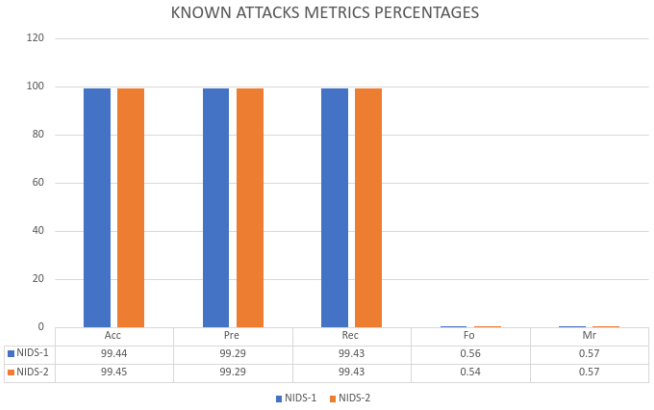

Fig. 4. Known attacks metrics percentages

Looking at Figure 4, related to known-only attacks, it can be noted that minor improvements appear. However, accuracy and fallout are a little better. In fact, metrics related to known-only attacks can not be sensibly improved because component C2 has not been changed. So, Figure 4 reflects expected results. The addition of the second OC-SVM model to NIDS-2 results in a better performance of accuracy, precision and fallout related to unknown-only attacks. This fact can be noted at Figure 5. As we have commented in the previous section, the overall accuracy is the probability that a NIDS correctly classifies an entry between anomaly or benign. If we denote by X_i the random variable that represents the number of entries correctly classified by the NIDS–i, $i \in \{1, 2\}$, this variable has a Binomial distribution. And the overall accuracy Acc_i represents the second parameter of the Binomial distribution. With these hypotheses we performed the test $H_0 : Acc_1 - Acc_2 = 0, H_1 : Acc_1 - Acc_2 < 0$, and we obtained the p-value $p = 0.003$. This result allows us to conclude that the metric Acc_2 is greater than Acc_1 not only at the sample level but for the entire statistical population. In addition, we performed the test $H_0 : Acc_2 = 0.96, H_1 : Acc_2 \neq 0.96$, and obtained the p-value $p = 0.907$. This allows us to conclude that the overall accuracy of NIDS-2 has an order of 96.0%. Therefore, the NIDS-2 developed in this paper has high accuracy in detecting known and unknown attacks.

VII. Conclusions

The literature contains NIDS against closed or open datasets, that is, those datasets containing benign & known or benign & unknown attacks, respectively. In this work, we propose the NIDS-2 against closed and open datasets.

NIDS-2 combines unsupervised and supervised ML models. Components one and three contains unsupervised OC–SVM models, while component two contains multiple trained supervised models. Working together, these two types of ML techniques perform well in detecting and classifying known attacks and, at the same time, detecting unknown attacks. Generically, detecting unknown attacks is usually hard to achieve.

The first OC–SVM model does the binary classification of benign or not–benign entries. This task is the same in NIDS–1 and NIDS–2. Nonetheless, adding a second OC–SVM in NIDS-2 (third component) has been used for correcting false unknown attacks (thus, improving the Fo value). Figures in Figure 3 show that this addition enhances precision and accuracy.

Local analysis of the detection of unknown-only attacks shows a better performance than known-only ones. See the

metrics related to NIDS-2 in figures 4 and 5. Thus, for future detection improvements against unknown-only attacks, several binary classification actions working together are worth considering.

Finally, it is worth noting that we are currently working on extending the proposed work. The main idea is to check the NIDS-2 architecture against several different datasets. Different types of anomalies playing the role of unknown attacks will also be tested.

ACKNOWLEDGMENT

This project has received funding from the European Union's H2020 research and innovation programme under the grant agreement No. 952644, from the Spanish Ministry of Science and Innovation under contract PID2021-124463OB-I00, and from the Catalan Government under contract 2021 SGR 00326.

REFERENCES

[1] Online site: https://konbriefing.com/en-topics/cyber-attacks.html access on November 9,2022.

[2] M. Fahad Umer, M. Sher, Y. Bi, Flow-based intrusion detection: Techniques and challenges, *Computers & Security*, **Vol. 70** (2017) pp. 238-254, ISSN 0167-4048. https://doi.org/10.1016/j.cose.2017.05.009.

[3] K. Shaukat, S. Luo, V. Varadharajan, I. A. Hameed and M. Xu, A Survey on Machine Learning Techniques for Cyber Security in the Last Decade, *IEEE Access*, **vol. 8**, pp. 222310-222354 (2020) doi: 10.1109/ACCESS.2020.3041951.

[4] J. Olamantanmi Mebawondu, et al., Network intrusion detection system using supervised learning paradigm, *Scientific African*, **Vol. 9** (2020) e00497, ISSN 2468-2276, https://doi.org/10.1016/j.sciaf.2020.e00497

[5] C. Ioannou, V. Vassiliou, Classifying Security Attacks in IoT Networks Using Supervised Learning, *2019 15th International Conference on Distributed Computing in Sensor Systems (DCOSS)* (2019) pp. 652-658, doi: 10.1109/DCOSS.2019.00118.

[6] A. S. Ahanger, S. M. Khan and F. Masoodi, An Effective Intrusion Detection System using Supervised Machine Learning Techniques, *2021 5th International Conference on Computing Methodologies and Communication (ICCMC)* (2021) pp. 1639-1644, doi: 10.1109/ICCMC51019.2021.9418291.

[7] H. Hindy et al., A Taxonomy of Network Threats and the Effect of Current Datasets on Intrusion Detection Systems, *IEEE Access*, **vol. 8** pp. 104650-104675 (2020) doi: 10.1109/ACCESS.2020.3000179

[8] M. Verkerken, L. D'hooge, T. Wauters, B. Volckaert and F. De Turck, Unsupervised Machine Learning Techniques for Network Intrusion Detection on Modern Data, *2020 4th Cyber Security in Networking Conference (CSNet)* (2020) pp. 1-8, doi: 10.1109/CSNet50428.2020.9265461.

[9] P. Winter, E. Hermann and M. Zeilinger, Inductive Intrusion Detection in Flow-Based Network Data Using One-Class Support Vector Machines, *2011 4th IFIP International Conference on New Technologies, Mobility and Security* (2011) pp. 1-5, doi: 10.1109/NTMS.2011.5720582.

[10] Choi, H., Kim, M., Lee, G. et al. Unsupervised learning approach for network intrusion detection system using autoencoders, *J. Supercomput 75*, 5597–5621 (2019). https://doi.org/10.1007/s11227-019-02805-w

[11] Umer MF, Sher M, Bi Y, A two-stage flow-based intrusion detection model for next-generation networks, *PLoS ONE* **13(1)** (2018) e0180945. https://doi.org/10.1371/journal.pone.0180945

[12] F. Aguiló–Gost; E. Simó–Mezquita; E. Marín–Tordera; A. Hussain, A Machine Learning IDS for Known and Unknown Anomalies, *2022 International Workshop on Design of Reliable Communication Networks (DRCN)*.

[13] A Realistic Cyber Defense Dataset (CSE-CIC-IDS2018) was accessed on January 2020 from https://registry.opendata.aws/cse-cic-ids2018.

[14] Raj, Mehedi Hasan, et al., Iot botnet detection using various one-class classifiers, *Vietnam Journal of Computer Science* **8.02** (2021): 291-310.

[15] Lang, Michel, et al., mlr3: A modern object-oriented machine learning framework in R, *Journal of Open Source Software* **4.44** (2019): 1903.

[16] Mariana Belgiu, Lucian Drăguţ, Random forest in remote sensing: A review of applications and future directions, *ISPRS Journal of Photogrammetry and Remote Sensing* **vol. 114** (2016): 24-31. ISSN 0924-2716, https://doi.org/10.1016/j.isprsjprs.2016.01.011.