uc3m | Universidad Carlos III de Madrid

Master Degree in Cybersecurity
Academic Year (e.g. 2020-2021)

Master Thesis

“Training dataset selection for anomaly detection algorithms. A cross-validation between different datasets”

Javier Balanzategui Sánchez

Sergio Pastrana Portillo

Madrid, 5th July 2021

Training dataset selection for anomaly detection algorithms. A cross-validation between different datasets.

Author: Javier Balanzategui Sánchez, Professor: Sergio Pastrana Portillo

***Abstract*—One of the functions of a Intrusion Detection System is to detect anomalies that could indicate any intruder within the security perimeter. Current anomaly detection algorithms in Network Intrusion Detection Systems are strongly related with the use of Artificial Intelligence and Machine Learning. These algorithms are capable of detect outliers (anomalies) within massive amounts of data in real-time while re-adapting to new data and learning from them. But, there are several variables to take into account when programming machine learning algorithms: to choose a training dataset, to select the predictor variables and to test which classification paradigm perform best using some precise evaluation metrics. The succeed or failure of the model will totally depend on the selection of these variables. The aim of this paper is to focus on the troublesome of selection of the training dataset and evaluate if there is transfer learning using the same trained model into different scenarios.**

To train a cyber security model, there are a few publicly available datasets that can be used. However, these datasets have been acquired in laboratory environments which tries to simulate real scenarios and they are limited in the quantity and variability of the samples used and attacks. In this paper, it is tested if any of these datasets can be used to train productive Network Intrusion Detection models. Also, it is studied if these datasets can train models that performs well under different environments or if the only solution is to gather a dataset within the specific environment where the model will be deployed.

***Index Terms*—machine learning, cross-validation, cybersecurity, datasets, classifiers, training data, transfer learning**

I. Introduction

ANOMALY detection is one of the main investigation branches in the cyber defense field. The capability of finding out new threats and attacks leads to use artificial intelligence (hereinafter, AI) in detection algorithms [1].

Historically, anomaly detection mechanisms started being more simple than they are now. They started more than 30 years ago using quality metrics and statistics together with rule sets to find any deviations from the regular use [2]. Over the years, the AI field has increased together with the hardware capabilities of the appliances, more automated and sophisticated methods have emerged on scene (this is described more in detail in section II) that allows to apply AI to this problem.

Machine learning (hereafter, ML) is the artificial intelligence branch which is used to this commitment. There are two main types of ML algorithms, unsupervised and supervised algorithms. Both type of algorithms need training data to work. Unsupervised algorithms are capable to find similarities between data and thus, categorize or to make clusters of data. This is useful when there is not information about how many different classes of data do you have. In the other opposite, supervised algorithms are capable to classify new instances of data into known clusters or categories. In Network Intrusion Detection Systems, supervised algorithms are used to detect anomalies that could indicate an intrusion, as it deviates for the regular behaviour of the network. For this family of algorithms, labeled or categorized data are needed to train supervised algorithms. There are several investigation groups that provides labeled cybersecurity datasets that can be used for this purpose as Stratosphere Lab [3], UNSW Canberra [4] or the Canadian Institute for Cybersecurity [5].

In this paper, a cross-validation survey between several different datasets is performed. That means, to measure the difference and effectiveness of training a model with one of the datasets and measure the performance of the trained model against the others. This way, it is possible to discover if models can only detect anomalies in the same scenario where training data were acquired or if they can detect anomalies in unknown environments. This is what is called Transfer Learning which is to employ trained models in similar but different scenarios [6]. Thus, it has been also studied if datasets are very scenario dependant. Thereby, an on-site gathered dataset will be always needed to train a productive model for anomaly detection into an specific organization.

This paper is structured into several parts. In related work section it is explained the main used studies to support this investigation. Methodology section is subdivided in dataset preprocessing, feature selection, evaluation metrics and model training sections. Each part is ordered chronological according to the steps performed. In the end, results and conclusions sections provides the main deductions of the study. Finally, a discussions and limitations section is also included in the paper. This section contains some restrictions and learned lessons from the study.

The developed program to preprocess the datasets, together with the training and testing phase is provided under my personal Github[1]

[1] https://github.com/JBalanza/cybersecurity_dataset_crossvalidation

II. Related Work

Some studies have used these public datasets to train cyber defense models that detect anomalies. For example, Kumar et al. [7] used IoT-23 dataset [8] to train a model achieving up to a 99% of precision. Also, Abdulrahman et al. [9] created a model using ISCX-IDS2017 dataset [10] that had a 98% of accuracy. This demonstrated that public available labeled cybersecurity datasets can be used to create well performing models achieving excellent results. However, no studies has been found that use these pre-trained models in real-world productive scenarios and how well they perform. it is important to investigate if these datasets can be exported to use in real productive scenarios. Otherwise, the only way to train an anomaly detection system is to use specific gathered on-site data, increasing the effort for organizations to develop their own anomaly detection mechanisms.

Some studies like the one performed by Zhao et al. [11] studied the transfer learning problem applied to new unknown attacks but within the same scenario/organization. They trained a model with old data and tested with new modern and unknown data (new attack vectors) for the algorithm. However, the importance of the dataset selection in the training stage, comparing and crossvalidating the results with different datasets has not been widely studied.

Another studies have been made comparing different paradigms trained with these datasets. Elmrabit et al. [12] measure the performance of 11 different ML algorithms to detect which ones are the best in anomaly detection. They used tree different cyber security datasets [10] [13] [14] to train the models and extrapolate the result from the used datasets (due to the results are almost the same no matter the used dataset). Some of the paradigms with best results have been used in this paper.

Also, some of the datasets needs preprocessing in order to create a data structure that can be ingested by the ML algorithms. The preprocessing of the datasets has been done using a similar approach of [15]. Nevertheless, some changes had to be implemented in order to speed up the algorithms. These changes are described more in detail in section IV-A.

III. Dataset selection

There are several datasets which can be used for this study. It is important to choose datasets which have the same type of anomalies. Otherwise, models will tend to detect specific attacks which are only in the dataset used for training. For example, it would not be acceptable to train the model with data from brute force attacks and validate the model against botnets attacks. The Table II shows the principal public available datasets used for cybersecurity and its labels or categories.

Most datasets contains a lot of different attacks. To reduce training time, it has been selected only C&C related traffic (asterisk (*) marked in Table II). It has been checked that different datasets uses different labels that are all related to the same threat. C&C traffic is tagged as ”C&C”, ”shellcode”, ”backdoor”, ”infiltration” and even ”botnet” (Ares malware in this case). But all of them points out to an attacker accessing to a machine remotely via backdoor/malware. The rest of the traffic (not C&C related) has been removed to reduce training and experimentation time. With all these in mind, finally three datasets have been selected: IoT-23 [8], IDS-2018 [16] and ToN-IoT [17]. Below, a brief description is provided for each one together with Table I which adds other comparative data.

It is important to notice that the quantity of different labels within a given dataset might differ from Table I and II. This is because in Table II some labels have been summarized. For example in case of IoT23, there are 8 different labels related to C&C (e.g: Mirai, Torii, Heartbeat...) that has been summarized into ”C&C” label only. This also happens with other categories like web attacks, which datasets like ISCX-IDS2018 includes some specific web attacks labels like XSS attacks (Cross-Site Scripting) that have been all included into ”Web attacks” category.

A. IoT-23

The IoT-23 dataset contains 23 different scenarios of network captures. Every scenario is build using IoT devices connected into the same network. Some of the scenarios has IoT devices with well-known malware running on them, while some others are sending benign traffic. The dataset is provided in PCAP files plus a *label.csv* file containing the label and some features (IP source address, IP destination address, source port, destination port, timestamp, etc) of each flow [8].

B. IDS2018

The ISCX-IDS 2018 dataset use two different categories of machines to generate the traffic. *”B-Profiles”* and *”M-profiles”*. The first group generates benign traffic, so every of its flows are labeled as benign. In the other hand, the second group generates some malicious traffic flows as they are generated by some real malware installed within the machine. The data is provided in large CSV files containing all CICFlowMeter’s [18] features plus the flow’s label. This is the standard format to use the data for training (CSV file

	IoT23	ISCX IDS2018	ToN-IoT
Institution	Stratosphere (CZ)	CIC (CA)	UNSW (AU)
Duration (Hours)	486	240	Unknown
Year	2020	2018	2020
Different Labels	16	13	8
Size (GB)	71	51	3
Format ready to use	No	Yes	No

TABLE I
Dataset properties

Label Name	CTU13	CTU Mixed	IoT23	ISCXIDS2012	ISCXIDS2017	ISCXIDS2018	Botnet IoT	ADFA-NB15	ToN-IoT
IRC	X		X						
SPAM	X								
ClickFraud	X								
PortScan	X		X			X			X
Dos/DDoS	X		X	X	X	X		X	X
FastFlux	X								
P2P	X								
Web Attacks	X				X	X		X	X
C&C*		X	X	X	X				
Shellcode*								X	
Backdoor*								X	X
Infiltration*					X	X			
Botnets*					X	X			
File Download			X						
Okiru			X						
Worm								X	
Adware		X							X
Ransomware		X							X
Brute force					X	X			X
HeartBleed					X				
Fuzzer								X	
Analysis								X	
Reconnaissance								X	
Exploit								X	
Generic								X	X
Infection									X

TABLE II
LABELS IN AVAILABLE CYBERSECURITY DATASETS

which each line is composed by features plus the label in the end) so it is ready to use it in AI studies. [16].

C. ToN-IoT

The ToN-IoT dataset has been acquired using different heterogeneous sources (the environment is composed by Linux machines, Windows 7/10, IoT sensors and smartphones). Several malware samples and tools has been used to generate malicious traffic while benign traffic is also present in the network capture. This dataset is provided in PCAP files together with *label.csv* files containing flow's features and a label [17].

Publicly available datasets contains data from different attacks and scenarios. That's the reason why it has been selected only some datasets and some categories to work with, trying to choose the most similar categories between datasets. Two out of the three selected datasets (IoT23 and ToN-IoT) need some preprocessing in order to prepare the data to train the models (since data and labels are provided separately into different files).

IV. METHODOLOGY

The Figure 1 summarizes the steps followed in this study. In the following sections the methodology is explained in detail.

A. Dataset preprocessing

Datasets that are served in PCAP files together with label files needs a preprocessing step before using them in the models. Label files contains a set of features per flow, together with a label tag pointing out the type of flow (benign or C&C, Backdoor, Shellcode, Infiltration or Botnet traffic). But since this feature set only covers a couple of measures, it was interesting to obtain more different features per each flow. The tool CICFlowMeter [18] is capable of extract more than 80 features from every traffic flow. This tool has been used to covert PCAP files into large CSV files of 80+ features per line/flow. Nevertheless, these CSV files does not contain any label at all. To label these new feature entries, the information must be correlated with the label file to have a self-containing CSV with all the information together.

A set of unique features (called hereafter 5-tuple) were used to match unequivocally each label file entry with the new extended entries. We used the label's file features ”source IP”, ”source port”, ”destination IP”, ”destination port”, and ”protocol” to match the label entries the unlabeled ones. Then, we produced a unique CSV file with all the features extracted by CICFlowMeter plus the label. This methodology have been used by Bashir Hussain et al. [15].

Actually, this 5-tuple might not be enough to link up unequivocally a flow with its relative label. A counterexample could be a DNS query to a legit DNS server. Some DNS queries could be benign while others could take profit of recursive DNS queries to contact to a C&C through the legit DNS. Thus, it could be two flows with same source and destination IP address, same source and destination port and same protocol but with different labels. For this reason, the possibility to add the timestamp feature (which is present in both CSV and label files) to match the entries was also

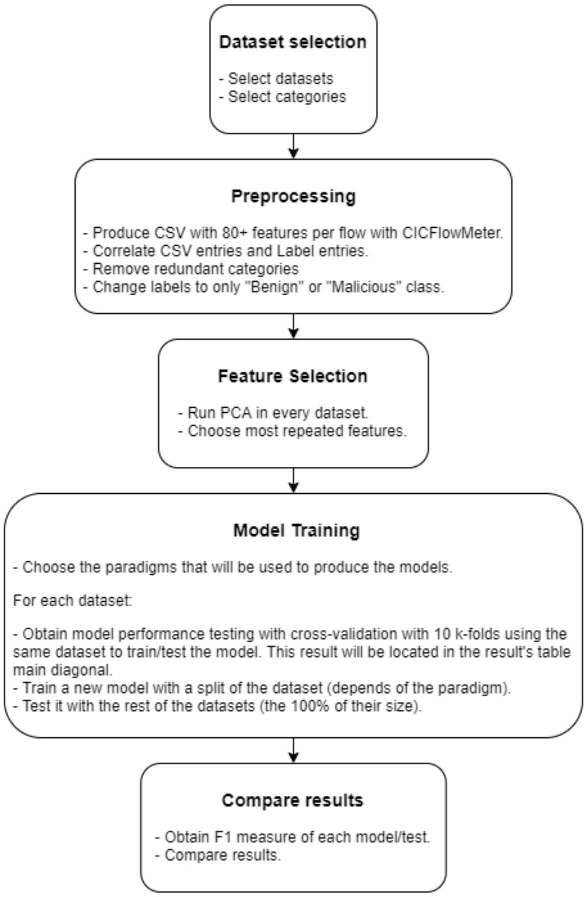
Dataset selection
- Select datasets
- Select categories
Preprocessing
- Produce CSV with 80+ features per flow with CICFlowMeter.
- Correlate CSV entries and Label entries.
- Remove redundant categories
- Change labels to only "Benign" or "Malicious" class.
Feature Selection
- Run PCA in every dataset.
- Choose most repeated features.
Model Training
- Choose the paradigms that will be used to produce the models.
For each dataset:
- Obtain model performance testing with cross-validation with 10 k-folds using the same dataset to train/test the model. This result will be located in the result's table main diagonal.
- Train a new model with a split of the dataset (depends of the paradigm).
- Test it with the rest of the datasets (the 100% of their size).
Compare results
- Obtain F1 measure of each model/test.
- Compare results.

Fig. 1. methodology used in this study

considered. After running some tests with this premise, it was discovered that machines within the scenarios were not synchronised. Within the same network captures some machines worked in UTC+4 timezone while others not. So data could not be matched as a result because the 5-tuples did not match in most cases. Finally, as stated in the original study [15], we continue to do not use timestamp despite it could have helped to avoid some flow mismatches.

In addition, the program that do the matching between labels and flows have been implemented using memory-only data structures instead of a classic database as this study [15] used in order to speed up the execution. In the original procedure, unlabeled data from CSV file is first inserted in a database. Then this data is labeled doing UPDATE queries if matches the 5-tuple identifier from the label file. The new approach uses a memory dictionary (also known as hash tables) which contains more nested dictionaries. This way, when updating data (adding the label to the matching flow) the search operations are speed up and a lot of time is saved compared to R/W operations in a classic database.

B. Feature selection

To create a ML model, it is needed to select a set of features that will be used by the classifier. Not all the features could be used by our model since some of them could not have any correlation with the label. One of the most extended methods to select variables is Principal Component Analysis (PCA) [19]. It is important to highlight that it will be only selected features that are available in every of our three selected datasets. Using the same prediction features in every of the models, will avoid the influence of the feature selection in the results. In this study, the only variance between models should be the training data to obtain a conclusion.

Variables dataset	IoT23	ISCX IDS2018	ToN-IoT	Total
Active Max		X	X	2*
Flow Duration	X		X	2*
Fwd IAT Max	X		X	2*
Fwd IAT Std	X		X	2*
Fwd Packet/s		X	X	2*
Packet Length Min	X		X	2*
Protocol	X	X		2*
Totlen Bwd Pkts	X	X		2*
Totlen Fwd Pkts	X	X		2*
ACK Flag Count			X	1
Active Min			X	1
Active Std		X		1
Bwd Bulk Rate Avg			X	1
Bwd Bytes/Bulk Avg			X	1
Bwd IAT Max		X		1
Bwd IAT Min		X		1
Bwd IAT Std			X	1
Bwd IAT Total			X	1
Bwd Init Win Bytes			X	1
Bwd Packet Length Min			X	1
Bwd Pkts/s		X		1
CWE Flag Count		X		1
CWR Flag Count			X	1
Down/Up Ratio			X	1
ECE Flag Cnt			X	1
FIN Flag Count		X		1
Flow Bytes/s		X		1
Flow IAT Mean		X		1
Fwd Packet Length Min			X	1
Fwd PSH Flags		X		1
Idle Max			X	1
Idle Min			X	1
Init Fwd Win Bytes		X		1
Packet Length Std			X	1
Pkt Len Max		X		1
PSH Flag Count			X	1
RST Flag Count			X	1
SYN Flag Count			X	1
Total Fwd Packet			X	1
Pkts Len var		X		1
Init Bwd Win Byts		X		1
Fwd Header Len		X		1
Flow IAT Max	X			1
Flow IAT Min	X			1
Subflow Bwd Bytes	X			1
Bwd IAT Mean	X			1
Fwd Seg Size Min	X			1

TABLE III
FEATURES CHOSEN BY PCA

After running PCA in our datasets, it was selected only features that are highly correlated with the label in at least two or more datasets. In Table III, it can be seen that there are only 9 features which are relevant in 2 or more datasets. Thus, this are the final set of features from the three datasets that will be used to train the models:

- Active Max: Maximum time a flow was active before becoming idle.
- Flow Duration: Duration of the flow in Microseconds.
- Fwd IAT Max: Maximum time between two packets sent in the forward direction.
- Fwd Packet/s: Number of forward packets per second.
- Packet Lenght Min: Minimum length of a packet.
- Protocol: Protocol number present in IPv4 header.
- Totlen Bwd Pkts: Total size of packet in backward direction.
- Totlen Fwd Pkts: Total size of packet in forward direction.

C. Evaluation metrics

The last thing to choose before comparing the results is to choose which quality measure it is going to be used to compare the performance of different models. As it can be seen in Table IV, label quantities by class are unbalanced. There are much more benign instances than malicious ones. Therefore, models might be biased towards the majority class and it will produce higher misclassification rates for the minority class [20], producing a higher false positive rate. It is important to focus on the success of classification of the smallest class (anomalies, malicious). For that reason, measures like Accuracy or Recall are not used for this purpose since the number of True Positives are several times higher than True Negatives, giving us a biased result.

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \qquad (1)$$

According to [21] and [22], F1 measure (combine both Precision and Recall measures) in Equation 1 is much more interesting to compare results between these models. In this case, F1 has been selected as it takes into account True Positives and True Negatives in a balanced way.

D. Model training

At this point, training data are ready and also the features (predictor variables) that will be used to create the models. Now it is time to choose the ML algorithm to implement. Several machine learning models have been created in order to abstract the results independently of the used algorithm. It has been tested different classification paradigms, but due this is a classification problem with imbalanced datasets, some of the paradigms are not convenient. As there are much more elements from the benign class than from the malicious (according to Table IV), some paradigms tend to be biased to the bigger class. According to Elmrabit et al. [12] it has been selected the paradigms that best work under this circumstances. Consequently, Random Forest [23], a Decision Tree classifier [24] and a K-Nearest Neighbors [25] classifiers have been selected. All of them brings good results at anomaly detection (out-layers) with unbalanced datasets [26] [27].

Other algorithms like SVM [28] and MLP [29] have been also tested. Although these algorithms does not perform well under this circumstances without a more exhaustive preprocess of the data, they have been tested to provide a more general overview of all different classification paradigms.

Label \Dataset	IoT23	ISCX IDS2018	ToN-IoT
Benign	12699950	13484708	2515236
C&C	14302		
Infiltration		161934	
Bot		286191	
Backdoor			27145

TABLE IV
LABEL ABSOLUTE QUANTITY BY DATASET

When training and testing data come from the same dataset, special considerations have to be taken. If same data entries are used for training and testing, the model will be biased and will provide better result than it will do in a real scenario. Another option is to split the dataset in two parts (can have different size, for example 60%-40%) and train the model with one and test with the other. The problem with this is that all data would not be used for training and that might result into underfitting (model have not enough quality data to train). For all this reasons, cross-validation with k-folds is the best solution to measure the efectiveness of the model [30]. Data is splitted in K-folds. For each fold, the model is trained with the rest K-1 folds and tested against the current fold (not used for training in that iteration). This process is done K-times changing the folds each iteration. Finally, each one of the folds have been used at least one time for testing and K-1 times for training. The mean of all K results indicates the real performance of the model with a high reliability. In this paper, 10 folds have been used to test the models when the training and testing data come from the same dataset. In the rest of the cases where the model will be trained from one dataset and tested with other, different training sizes have been used depending of the algorithm. Some of the algorithms works better with large training sets while others provides better results under shorter training sets. Different training dataset sizes have been tested in order to avoid underfitting (not enough training) and overfitting (too much training). Both situations would give bad results. In section V section, it is shown the configuration (internal parameters of some algorithms due to its nature) and training sizes which best performed of every algorithm.

V. RESULTS

Models were created with the selected features and classification algorithms, giving F1 as output to test the efectiveness of the classification. In the following tables, F1 results of training the model with a dataset and testing against other are shown.

Train \Test	ToN-IoT	ISCX IDS2018	IoT23
ToN-IoT	0.99	0.00004	0
ISCX IDS2018	0.00003	0.71	0.03
IoT23	0.04	0.11	0.9

TABLE V
Using a Decision Tree classifier

Train \Test	ToN-IoT	ISCX IDS2018	IoT23
ToN-IoT	0.99	0.0001	0
ISCX IDS2018	0.27	0.74	0
IoT23	0	0.003	0.97

TABLE VI
Using a Random Forest classifier

Train \Test	ToN-IoT	ISCX IDS2018	IoT23
ToN-IoT	0.04	0.01	0.22
ISCX IDS2018	0.06	0.01	0.46
IoT23	0.00005	0.044	0.44

TABLE VII
Using a Linear SVM with max iterations = 1000 and a train size split of 20%

Train \Test	ToN-IoT	ISCX IDS2018	IoT23
ToN-IoT	0.61	0	0
ISCX IDS2018	0.00007	0	0
IoT23	0	0	0

TABLE VIII
Using a Multi-Layer Perceptron with 300 hidden layers, max 500 iterations and a train size split of 80%

Train \Test	ToN-IoT	ISCX IDS2018	IoT23
ToN-IoT	0.99	0.0003	0
ISCX IDS2018	0.00003	0.79	0.0005
IoT23	0.0002	0.05	0.99

TABLE IX
Using a K-Neighbor classifier a train size split of 80%

There are three tested models that performs reasonably good: Decision Tree (Table V), Random Forest Classifier (Table VI) and K-Nearest Neighbours (Table IX). Others paradigms like Linear SVM (Table VII) or Multi-Layer Perceptron (Table VIII) need more extra tuning or preprocessing to adapt to unbalanced datasets (for example, converting unbalanced datasets in balanced ones having an homogeneous distribution of the labels [20] [9]). It can be observed in the main diagonal, the cross-validation results using 10 folds using the same dataset to train and test. Looking at the main diagonal in the three best working models, it is confirmed that they're doing reasonably good in classifying data from the same dataset it was used to train the models. So it can be stated due to the great results that these models are ready to find anomalies. Despite that, it is possible to see that models work really bad when tries to detect anomalies in datasets different from the training one. Models do not perform well classifying data from different scenarios from the training one. There are not any tested paradigm in this study which is effective in these situations, results are deafening in all cases no matter the paradigm used. For this reason, the results are independent of the paradigm used. Nevertheless, ISCX-IDS 2018 results (in all models) could be improved by adding new features to our model despite it will not help probably with the other datasets.

VI. Discussions and Limitations

Prior to the implementation of the models, a large time has been spend preparing the datasets to be consumed by the models. Since the data and labels are provided separated in most of the cases (except in the case of ISCX-IDS 2018), a large amount of time was spent programming the algorithms to join the data together with their related label. Also, a machine with high resources (48 GB of RAM and 24 threads) were employed for this task because programs have to be running for several days after producing the result.

The main goal was to find a ”shared” model which works well under every different training datasets, without making any distinction between them when training the models, all models uses the same predictor variables for all datasets. It is possible that with much more effort and AI knowledge, paradigms that did not performed well could provide better results. In any case, at least three of them provided good results on self-classification (training and test data come from the same dataset) which means that the model works well under some circumstances (self-classification) but not under another ones (to classify data from different scenarios). Thus, we can use the results of these three models to make a conclusion. As we have more than only one different paradigm that works well, we can infer that the results are not paradigm-related but related to the training data. It cannot be used training data acquired in a different environment to create a working anomaly detection model in a different environment.

Furthermore, in this study only have been used C&C related data due to the time and the resources available. As these datasets are over 125 GB in total, it has to be decided only one topic or label among all provided (See section III). Otherwise, training and testing time would not be suitable for this study. Selecting another label or topic might bring different results.

From this paper, important lessons should be learned for further studies. First, no generic training data should be used to train our models. Training data should be acquired on-site. Also it should be taken into account that usually gathered data are unbalanced (the distribution of the labels is not homogeneous, there are much more elements for some classes than others). This is important when selecting the paradigm and training our models as it could need some extra tuning than regular Machine Learning scenarios.

VII. Conclusions

Despite the number of publicly available cybersecurity labelled datasets, they cannot be used for training real production models in specific environments. On-site data should be acquired within the particular organization to train the models. If external datasets are used for training, models will do extremely bad at finding real anomalies. Every scenario has its own peculiarities that makes it unique, from a classification point of view.

Public available datasets can be used in the experimentation phase to select a suitable paradigm, test real-time processing algorithms, etc. But training data should be acquired and labeled on-site to train the models. The most complicated part might be the acquisition of malicious traffic because tool and malware sample could produce different fingerprints. The higher the variety of the malicious traffic, the better the model would be adapted to recognize real-world attacks and anomalies.

Our work confirms that off-the-shelf ML solutions are not appropriate anomaly detection, since they require a complex work of tuning before they are ready for being deployed in real scenarios.

References

[1] S. Omar, M. Ngadi, H. Jebur, and S. Benqdara, "Machine learning techniques for anomaly detection: An overview," *International Journal of Computer Applications*, vol. 79, 10 2013.

[2] D. Denning, "An intrusion-detection model," *IEEE Transactions on Software Engineering*, vol. SE-13, no. 2, pp. 222–232, 1987.

[3] S. lab, "Stratosphere Lab, CTU Prague University," https://www.stratosphereips.org/datasets-overview, [Online].

[4] U. C. at the Australian Defence Force Academy, "NB15 dataset," https://research.unsw.edu.au/projects/unsw-nb15-dataset, [Online].

[5] U. of New Brunswick, "Canadian Institute for Cybersecurity," https://www.unb.ca/cic/datasets/index.html, [Online].

[6] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, "A comprehensive survey on transfer learning," *CoRR*, vol. abs/1911.02685, 2019. [Online]. Available: http://arxiv.org/abs/1911.02685

[7] A. Kumar, M. Shridhar, S. Swaminathan, and T. J. Lim, "Machine learning-based early detection of iot botnets using network-edge traffic," *CoRR*, vol. abs/2010.11453, 2020. [Online]. Available: https://arxiv.org/abs/2010.11453

[8] S. Garcia, A. Parmisano, and M. J. Erquiaga, *IoT-23: A labeled dataset with malicious and benign IoT network traffic*. Zenodo, Jan. 2020, More details here https://www.stratosphereips.org /datasets-iot23. [Online]. Available: https://doi.org/10.5281/zenodo.4743746

[9] A. Abdulrahman and M. K. Ibrahem, "Toward constructing a balanced intrusion detection dataset based on cicids2017," 01 2020.

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

[11] J. Zhao, S. Shetty, J. Pan, C. Kamhoua, and K. Kwiat, "Transfer learning for detecting unknown network attacks," *EURASIP Journal on Information Security*, vol. 2019, 02 2019.

[12] N. Elmrabit, F. Zhou, F. Li, and H. Zhou, "Evaluation of machine learning algorithms for anomaly detection," in *2020 International Conference on Cyber Security and Protection of Digital Services (Cyber Security)*, 2020, pp. 1–8.

[13] N. Moustafa and J. Slay, "Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set)," in *2015 Military Communications and Information Systems Conference (MilCIS)*, 2015, pp. 1–6.

[14] U. Adhikari, S. Pan, and T. Morris, "Power System Datasets in Industrial Control System (ICS) Cyber Attack Datasets," https://sites.google.com/a/uah.edu/tommy-morris-uah/ics-data-sets, [Online].

[15] F. B. Hussain, S. G. Abbas, U. U. Fayyaz, G. A. Shah, A. Toqeer, and A. Ali, "Towards a universal features set for iot botnet attacks detection," *CoRR*, vol. abs/2012.00463, 2020. [Online]. Available: https://arxiv.org/abs/2012.00463

[16] I. Sharafaldin, A. Habibi Lashkari, and A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," 01 2018, pp. 108–116.

[17] A. Alsaedi, N. Moustafa, Z. Tari, A. Mahmood, and A. Anwar, "Ton-iot telemetry dataset: A new generation dataset of iot and iiot for data-driven intrusion detection systems," *IEEE Access*, vol. 8, pp. 165 130–165 150, 2020.

[18] A. Habibi Lashkari, "Cicflowmeter-v4.0 (formerly known as iscxflowmeter) is a network traffic bi-flow generator and analyser for anomaly detection. https://github.com/iscx/cicflowmeter," 08 2018.

[19] Y. Mori, M. Iizuka, T. Tarumi, and Y. Tanaka, "Variable selection in principal component analysis," *Statistical Methods for Biostatistics and Related Fields*, pp. 265–283, 01 2007.

[20] V. López, A. Fernández, S. García, V. Palade, and F. Herrera, "An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics," *Information Sciences*, vol. 250, p. 113–141, 11 2013.

[21] Y. Lu, Y.-M. Cheung, and Y. Y. Tang, "Bayes imbalance impact index: A measure of class imbalanced data set for classification problem," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 9, pp. 3525–3539, 2020.

[22] Yanminsun, A. Wong, and M. S. Kamel, "Classification of imbalanced data: a review," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 23, 11 2011.

[23] R. Primartha and B. Adhi Tama, "Anomaly detection using random forest: A performance revisited," 11 2017, pp. 1–6.

[24] D. Sinwar and M. Kumar, "Anomaly detection using decision tree based classifiers," vol. 3, 01 2014.

[25] G. Guo, H. Wang, D. Bell, and Y. Bi, "Knn model-based approach in classification," 08 2004.

[26] P. Mitra, D. Akhiyarov, M. Araya-Polo, and D. Byrd, "Machine learning-based anomaly detection with magnetic data," 12 2020.

[27] C.-O. Truică and C. Leordeanu, "Classication of an imbalanced data set using decision tree algorithms," *University Politehnica of Bucharest Scientific Bulletin Series C - Electrical Engineering and Computer Science*, vol. 79, pp. 69–, 12 2017.

[28] L. Ladicky and P. Torr, "Linear support vector machines," 01 2011, pp. 985–992.

[29] P. Marius, V. Balas, L. Perescu-Popescu, and N. Mastorakis, "Multilayer perceptron and neural networks," *WSEAS Transactions on Circuits and Systems*, vol. 8, 07 2009.

[30] P. Refaeilzadeh, L. Tang, and H. Liu, *Cross-Validation*. Boston, MA: Springer US, 2009, pp. 532–538. [Online]. Available: https://doi.org/10.1007/978-0-387-39940-9_565