LADS: A Live Anomaly Detection System based on Machine Learning Methods

Gustavo Gonzalez-Granadillo[1], Rodrigo Diaz[1], Ibéria Medeiros[2], Susana Gonzalez-Zarzosa[1] and Dawid Machnicki[1]

[1]*Atos Research & Innovation, Cybersecurity Laboratory, Spain*

[2]*LASIGE, Faculty of Sciences, University of Lisboa, Portugal*

Keywords: Machine Learning, One-class SVM, Anomaly Detection, Network Traffic Behavior, NetFlow.

Abstract: Network anomaly detection using NetFlow has been widely studied during the last decade. NetFlow provides the ability to collect network traffic attributes (e.g., IP source, IP destination, source port, destination port, protocol) and allows the use of association rule mining to extract the flows that have caused a malicious event. Despite of all the developments in network anomaly detection, the most popular procedure to detect non-conformity patterns in network traffic is still manual inspection during the period under analysis (e.g., visual analysis of plots, identification of variations in the number of bytes, packets, flows). This paper presents a Live Anomaly Detection System (LADS) based on One class Support Vector Machine (One-class SVM) to detect traffic anomalies. Experiments have been conducted using a valid data-set containing over 1.4 million packets (captured using NetFlow v5 and v9) that build models with one and several features in order to identify the approach that most accurately detects traffic anomalies in our system. A multi-featured approach that restricts the analysis to one IP address and extends it in terms of samples (valid and invalid ones) is considered as a promising approach in terms of accuracy of the detected malicious instances.

1 INTRODUCTION

Anomaly detection solutions work on the basis of the data provided by different network monitoring technologies (e.g., sFlow and NetFlow) or software-defined management solutions (e.g., OpenFlow) that allow access to forwarding devices. These approaches are used to collect data about the traffic as it enters or exits an interface (Bradatsch, 2019). Due to the lightweight and efficient nature of the NetFlow protocol, a number of widely used NetFlow-based solutions have been developed and a great number of algorithms have been proposed to detect abnormal activities in NetFlow data (Vaarandi, 2013).

Despite of all the developments in network anomaly detection, the most popular procedure to detect non-conformity patterns is still manual inspection during the period under analysis (e.g., visual analysis of plots, identification of variations in the number of bytes, packets, flows). Most of the current work focuses on the identification of anomalies based on traffic volume changes. However, since not all of the anomalies are directly reflected in the number of packets, bytes or flows, it is not possible to identify them all with such metrics (Lakhina et al., 2005). Approaches to address this issue propose the use of IP packet header data. Particularly, IP addresses and ports allow the characterization of detected anomalies.

In this paper, we propose a *Live Anomaly Detection System* (denoted by LADS) based on unsupervised machine learning algorithms. The approach explores the use of a well known machine learning method, the One class Support Vector Machine (One-class SVM), to identify abnormal behaviors in the NetFlow traffic captured. A set of experiments have been conducted using one and multiple features of flows in order to identify the best approach to detect anomalies in a given traffic using NetFlow. Results show a promising approach using multiple features that restrict the analysis to a modelled IP address and extended it in terms of samples (valid and invalid ones).

The remainder of this paper is structured as follows: Section 2 discusses related work. Section 3 presents the architecture of LADS. Section 4 introduces the One-class SVM algorithm and dataset used to model the network traffic behavior. Section 5 de-

Gonzalez-Granadillo, G., Diaz, R., Medeiros, I., Gonzalez-Zarzosa, S. and Machnicki, D.
LADS: A Live Anomaly Detection System based on Machine Learning Methods.
DOI: 10.5220/0007948904640469
In *Proceedings of the 16th International Joint Conference on e-Business and Telecommunications (ICETE 2019)*, pages 464-469
ISBN: 978-989-758-378-0

tails the experiments performed to train, test and select the most accurate anomaly detection model. Finally, conclusions and perspective for future work are presented in Section 6.

2 RELATED WORK

Machine learning (ML) techniques have been widely proposed in the literature as a viable approach for network anomaly detection. Wagner et al. (Wagner et al., 2011), for instance, propose a machine learning approach for IP-Flow record anomaly detection. The approach is based on SVM algorithm, in order to analyze and classify large volumes of NetFlow records.

Xu and Wang (Xu and Wang, 2005) propose an adaptive network intrusion detection method based on principal component analysis (PCA) and SVM. The approach aims at optimizing the training of the classifier and the tests using a PCA algorithm. This latter performs the optimization by discarding all unnecessary data coming into the IDS and then running SVM on the data. Although much faster, results show that the approach degrades the detection rate using PCA algorithm.

Zhang et al. (Zhang et al., 2008) present a method based on One-class SVM for detecting network anomalies. Experiments are carried out on three different sets of telecommunication network data and the results show the promising performance of the approach. The main drawback of this approach is that its ability to detect anomalies rely on the choice of a kernel function (i.e., Radial Basic Function), whose parameters are difficult to define. In addition, the approach does not perform comparison tests with other tools, nor it performs anomaly detection tests with different kernels.

Mulay et al. (Mulay et al., 2010) propose an approach that uses ML techniques to apply SVM and decision tree algorithms to a given data set. Authors use SVM and decision tree combined to see which would get the best results with multiclass data.

Amer et al. (Amer et al., 2013) propose two enhanced methods that use One-class SVM for unsupervised anomaly detection. The key idea of both enhancement is that outliers should contribute less to the decision boundary as normal instances. Experiments conducted on well-known data sets show that the proposed modifications on the One-class SVM algorithms are very promising.

Limthong (Limthong, 2013) proposes a real time computer network anomaly detection using ML techniques. The approach compares three standard and well-known ML algorithms (i.e., multivariate normal distribution, k-nearest neighbor, and one-class SVM) and uses a function based on the precision and recall parameters, to evaluate the detection performance of the proposed techniques. Results show that the proposed approach is suitable to be applied in real-time systems. Nevertheless, the approach lacks of information about time consumption during the training and test phase. In addition, the selection of features for detecting particular anomalies needs to be refined.

Our work differs from previous works in the way that it defines features (one or a combination of several IP-based features) to be used for evaluating the performance of a given machine learning algorithm in detecting legitimate and anomalous network traffic.

3 LADS ARCHITECTURE

The general architecture of *Live Anomaly Detection System* (LADS) is presented in Figure 1.

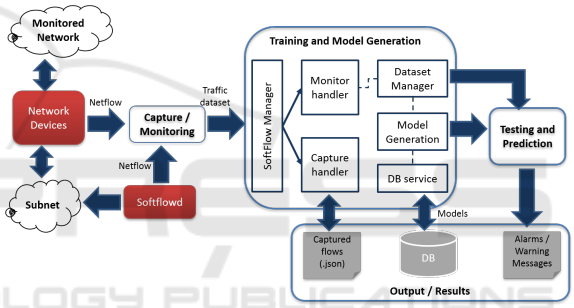

Figure 1: LADS Architecture.

- **Capture/Monitoring:** This module is composed of two elements: (i) the monitored network, which includes NetFlow data from both hardware devices (e.g., routers) and software elements (softflow); and (ii) the softflowd[1], a flow-based network traffic analyzer capable of exporting Cisco NetFlow data.
- **Training and Model Generation:** This module uses machine learning to build a model for testing and predictions over the captured data with a predefined time window that can vary according to the size of the data to be analyzed (e.g. 5-second window). The Model Generator uses a defined machine learning algorithm to build the model based on the legitimate traffic and the defined parameter(s) (e.g., IP addresses, port numbers, protocols). Models are then stored in the database by the DB service.
- **Testing and Prediction:** Once the machine learning model is built, the testing and prediction

[1] https://github.com/irino/softflowd

module performs the network traffic evaluation in real-time. For this, the module requires the dataset (composing of both legitimate and malicious event) from the Dataset Manager. The main objective of the prediction service is to identify anomalous behavior on the network traffic and to generate alarms (warning messages) based on the obtained results. If a sample falls outside the area defined by the model, then the sample is considered as anomalous. More details about this process is provided in Section 5.2.

- **Output/Results Module:** This module is composed of (i) the database that stores all models used for the prediction module. The database needs to be populated with NetFlow data, indicating all the arguments of the captured interface. It is then possible to indicate the root path of the NetFlow data, the profile and source name of the captured traffic; (ii) the captured flows generated by the Capture handler during the training and model generation process; and (iii) the alarms indicating anomalous behavior of the analyzed network traffic. The main objective of this module is to provide information about the generated models whenever these data is required by the testing and prediction module.

4 LADS MACHINE LEARNING ALGORITHM AND DATASET

This section details the machine learning algorithm and the dataset used for testing and validation of the LADS. It is important to note that we have built several prediction models using the same machine learning algorithm. The ultimate goal of this research work is to be able to use different algorithms to build different models and to compare their accuracy in detecting anomalous behavior on the Netflow data.

4.1 Machine Learning Algorithm

Out of the many novel mechanisms that can be used for intrusion detection, there are two promising approaches that can be adopted in order to satisfy the aforementioned requirement (Pedregosa et al., 2011) (i) Novelty detection, aiming at establishing a decision boundary that encompasses legitimate events; and (ii) Outlier detection, aiming at building a data model, based on an event database, which separates largely dense areas from outliers.

In the machine learning domain, there are several solutions from both learning groups: supervised (requiring classified data to train on) and unsupervised (autonomously identifying the classes within the data). Our research focuses on one particular unsupervised method:

- **One-class Support Vector Machine**[2] **(SVM)** is a variation of a widely used SVM algorithm that belongs to the novelty detection family. The training process is carried out only on samples representing legitimate (normal) behavior/events, in this case, the functional margin is established to encompass all the training samples and separate them from the rest of the space.

LADS uses the One-Class SVM algorithm to construct a hyper-plane or set of hyper-planes in a high or infinite dimensional space. A good separation is achieved by the hyper-plane that has the largest distance to the nearest training data points of any class (functional margin), therefore, the larger the margin the lower the generalization error of the classifier.

4.2 Dataset

The One-Class SVM algorithm has been tested against a valid dataset, containing NetFLow traffic from an training internal network of an electrical company infrastructure. The network is divided by rooms which contain devices that communicate with each other and, in some cases communicate with devices on other rooms.

The network has two capture points, one on Room 1 and another on Room 2, which were running Wireshark[3]. The same program was used to create two packet capture files that were later converted into NetFlow v5[4] and NetFlow v9[5] with a maximum time interval of 3 minutes for each flow and stored using nfdump[6] tools. Table 1 summarizes the information of the used dataset.

Before the training process, we cleaned the dataset so that broadcast, multicast and non-internal IP addresses were discarded. We split the dataset into two:

- Dataset 1: IP addresses from Room 1 excluding broadcast and multicast (6,000 flows); and
- Dataset 2: keeping only internal traffic inside Room 1 (5,800 flows).

[2] http://scikit-learn.org/stable/auto_examples/svm/plot_oneclass.html

[3] https://www.wireshark.org/

[4] https://www.plixer.com/support/NetFlow-v5

[5] https://www.plixer.com/support/NetFlow-v9

[6] https://github.com/phaag/nfdump

Table 1: Dataset Statistics.

NetFlow	Room	Packets	Flows	Start Time	End Time	Total Time	Vol. (MB)	IP Src	IP Dst	Port Src	Port Dst
v5	1	511,875	8,294	16:22:37	18:18:46	01:56:09	46.23	9	16	2,241	107
	2 and 3	208,365	3,759	17:08:14	18:15:51	01:07:37	26.47	12	19	721	24
	All	720,240	12,053	16:22:37	18:18:46	01:56:09	72.70	21	30	3,052	122
v9	1	514,713	8,716	16:22:37	19:52:47	03:30:10	46.54	13	24	2,523	179
	2 and 3	209,034	4,006	17:08:13	19:23:28	02:15:15	26.54	14	23	909	25
	All	723,747	12,722	16:22:37	19:52:47	03:30:10	73.08	27	38	3,293	194

5 LADS TRAINING AND TESTING

This section presents the validation process of our approach.

5.1 Training

We trained the LADS only with information about the IP addresses provided in the dataset. Since the training process requires to build a model based on the distance among IPs within the dimensional space at which they are embedded, an IP transformation is required. In order to reduce the dimensionality of the dataset used by the LADS, maintaining at the same time as much as possible information carried by the samples, the Principal Component Analysis function (PCA) is essential to find alternative features that maintain around 99% of the data variance, meaning that around 99% of information is carried by the original dataset.

After these steps, we trained the One-Class SVM algorithm on the clean data provided in the dataset and validated the trained models against the test set (see next experiments). The accuracy and precision of the trained models vary largely depending on the evaluated features.

5.2 Testing

Aiming at validating the models built by the LADS, we performed a series of tests with one, two, three and four features.

5.2.1 Experiment 1

Measuring Distance among IP Addresses (2 Features): The goal of this experiment is to compute the distance between the modelled IP and the new observed IP addresses. Such a distance could indicate how close or far the new observed IP is with respect to the modelled one. The rule in this experiment is to classify as anomalous those IPs whose distance exceeds a predefined threshold.

For this experiment we transformed all IP addresses into integer values (e.g., 172.18.21.4 is transformed to 2886735108) and we subtract the resulting integer of the modelled IP with the integer of the new observed IP. The LADS has been trained with dataset 1 and 2. We have used the One-Class SVM algorithm to compute the distance from the closest to the farthest IP address. A model based on these data is created and a region is designed accordingly. During the testing part, we have added seven IP addresses from outside the range used during the training process. All IPs from the block are considered legitimate and all those that fall outside the boundaries are considered anomalous. Results are shown in Figure 2.

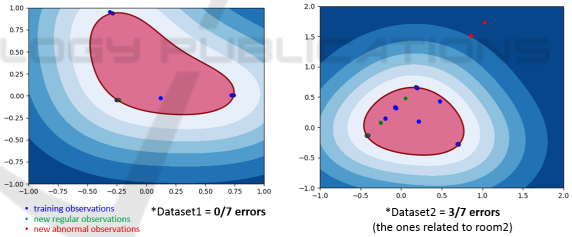

Figure 2: Results from Experiment 1.

As shown in Figure 2, the blue dots are the training observations, the green dots are the new regular observations, whereas the red dots are the new abnormal observations. Results from dataset 1 show that there is no error in the analysis (i.e., all IP addresses are considered legitimate, although seven of them do not belong to the range), whereas results from dataset 2 show that there are three errors (i.e., three out of seven IP addresses are considered anomalous). Precision (i.e., percentage of instances correctly classified as anomalies over the discovered anomaly) is very low in this case. In order to improve these results, it is necessary to build and test other models.

5.2.2 Experiment 2

Measuring Distance among IP Addresses and Protocols (3 Features): For this experiment we included not only IP source and destination, but also the protocol used in the connections as the features to be analyzed by the LADS. We have used the same datasets and the One-Class SVM to compute the distance form the closest to the farthest IP address. A model based on these data is created and a 3D region is designed accordingly. During the testing part, we have added seven IP addresses from outside the range used during the training process. All IPs from the block are considered legitimate and all those that fall outside the boundaries are considered anomalous. Results are shown in Figure 3.

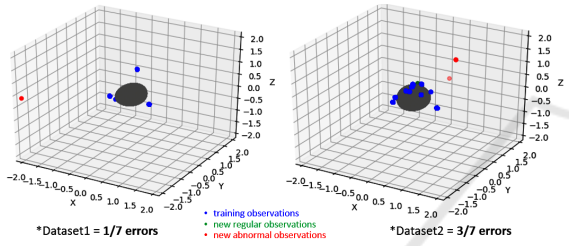

Figure 3: Results from Experiment 2.

Results from dataset 1 show that there is one error in the analysis (i.e., one out of seven IP addresses is considered anomalous), whereas result from dataset 2 show that there are three errors (i.e., three out of seven IP addresses are considered anomalous). Results are similar from those obtained in experiment 1, we decided to generate another model and run other types of experiments.

5.2.3 Experiment 3

Measuring Distance of Each IP Octet (4 Features): Instead of transforming the whole IP address into one integer, we split the IP address into four octets and each octet is treated as one different feature. Octets have been transformed into integer values (e.g., 172.18.21.4 is transformed to 288,67,35,108). Similar to previous training, a model based on these data is created and a region is designed accordingly. During the testing part, we have added seven IP addresses from outside the range used during the training process. All IPs from the block are considered legitimate and all those that fall outside the boundaries are considered anomalous. Since there are four features, it is not possible to obtain plots in four dimensions. However, results are similar to those obtained in experiment 1 (i.e., dataset 1 = 0/7 errors, dataset 2 = 3/7 errors). No improvements have been obtained with the use of a greater number of features.

5.2.4 Experiment 4

Converting IPs into Binaries (1 Feature): for this experiment we transformed each IP into its corresponding binary using the Label Binarizer encoding method[7], which consists on assigning zeros or ones to data points depending on what category they are in. During the training process, the LADS learns one binary classifier per class, by doing this, multi-class labels are converted into binary labels whether they belong or not to the class. During the testing process, one assigns the class for which the corresponding model gave the greatest confidence.

Results from this experiment show that no errors have been detected with the current training/testing dataset (i.e., all IP addresses are considered to be legitimate, including the seven outsiders). Still, precision is very low. No improvements have been obtained with the use of this approach.

5.2.5 Experiment 5

Evaluating 3 Features to Represent IP Source and Destination: A dedicated model has been created for one of the IP addresses included in the dataset (i.e., 172.18.21.4). We only included the incoming and outgoing traffic that goes through the selected IP address. The model considers three features:

- IP Location: we evaluate if the IP address of the analyzed instance is source or destination, for which a value of zero or one will be allocated accordingly (i.e., 0 if it is a source IP, 1 if it is a destination IP).
- IP Distance: we compute the distance between the modelled IP and the new one (as performed in previous experiments).
- IP Knowledge: we evaluate if the IP address of the analyzed instance is known or unknown, for which a value of zero or one will be allocated accordingly (i.e., 0 if it is a known IP, 1 if it is an unknown IP).

The testing dataset has been restricted only to the modelled IP and extended in terms of samples: 15 samples (3 valid and 12 invalid). Results are shown in Figure 4, from which 11 out the 12 invalid samples were detected as anomalous observations (1 false positive). In terms of the dataset, even without filtering external traffic, this approach is able to detect most of the anomalous samples.

[7]https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.LabelBinarizer.html

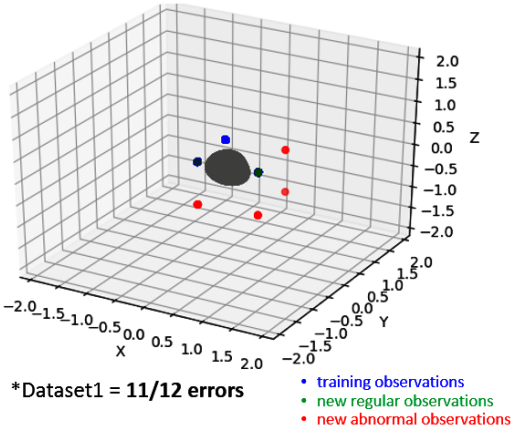

Figure 4: Results from Experiment 5.

6 CONCLUSION AND FUTURE WORK

We have proposed LADS, a Live Anomaly Detection System that uses unsupervised machine learning algorithms to detect anomalies after a training process of a given dataset. The approach analyses the performance of a well known machine learning algorithm: One class support vector machine (One-Class SVM) under different test environments (e.g., using one or multiple IP features), making it possible to identify which approach performs better in detecting legitimate and anomalous traffic. Training is done offline using valid datasets, but predictions are done in real time, making it possible to detect anomalous behavior of the current system's traffic.

Results show that a combination of multiple features (i.e., IP source, IP destination, distance between IPs, IP known, IP unknown) provides more accurate results and reduces considerably the false rates in the analysis performed.

It is important to highlight that these experiments are best suited for the behavior analysis of IoT devices (e.g., smart security systems, monitoring devices, jam detectors, smart thermostats, etc.) that generates one or few signals, and whose behavior can be easily modelled by the LADS. We should exclude computers, servers and mobile phones in our analysis, as the heterogeneity of the traffic they generate will make it difficult to represent accurate models.

Future work will consider other features to evaluate traffic behavior (e.g., connection time, traffic size, port source and destination, protocols, etc.) and will assign a percentage value to each of the evaluated features (based on its importance in identifying anomalies) to tune the tool and compare the accuracy of the results.

ACKNOWLEDGEMENTS

The research in this paper has been partially supported by the European Commission through the DiSIEM project, (Grant Agreement No. 700692), STOP-IT project, (Grant Agreement No. 740610), and the SerIoT project, (Grant Agreement No. 780139), as well as the LASIGE Research Unit (ref. UID/CEC/00408/2019).

REFERENCES

Amer, M., Goldstein, M., and Abdennadher, S. (2013). Enhancing one-class support vector machines for unsupervised anomaly detection. In *Proceedings of the ACM SIGKDD Workshop on Outlier Detection and Description*.

Bradatsch, L. (2019). Anomaly detection based on traffic records. In *International Conference on Networked Systems*.

Lakhina, A., Crovella, M., and Diot, C. (2005). Mining anomalies using traffic feature distributions. In *Conference on Applications, technologies, architectures, and protocols for computer communications*, pages 217–228.

Limthong, K. (2013). Real-time computer network anomaly detection using machine learning techniques. In *Journal of Advances in Computer Networks*, volume 1(1).

Mulay, S. A., Devale, P., and Garje, G. (2010). Intrusion detection system using support vector machine and decision tree. In *International Journal of Computer Applications*, volume 3(3), pages 40–43.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in python. In *Journal of Machine Learning Research*, volume 12, pages 2825–2830.

Vaarandi, R. (2013). Detecting anomalous network traffic in organizational private networks. In *International Multi-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support*.

Wagner, C., Francois, J., State, R., and Engel, T. (2011). Machine learning approach for ip-flow record anomaly detection. In *International IFIP TC 6 Conference on Networking*, pages 28–39.

Xu, X. and Wang, X. (2005). An adaptive network intrusion detection method based on pca and support vector machines. In *Advanced Data Mining and Applications*, pages 696–703.

Zhang, R., Zhang, S., Lan, Y., and Jiang, J. (2008). Network anomaly detection using one class support vector machine. In *Proceedings of the International MultiConference of Engineers and Computer Scientists*.