An Improved Live Anomaly Detection System (I-LADS) based on Deep Learning Algorithms

Gustavo Gonzalez-Granadillo, Alejandro G. Bedoya and Rodrigo Diaz
Atos Research & Innovation, Cybersecurity Laboratory, Spain

Keywords: Deep Learning, Neural Network, Anomaly Detection, Network Traffic Behavior, AutoEncoder.

Abstract: Network Anomaly detection is an open issue that considers the problem of finding patterns in data that do not conform to expected behavior. Anomalies exhibit themselves in network statistics differently; therefore developing general models of normal network behavior and anomalies is a challenging task. This paper presents an Improved Live Anomaly Detection System (I-LADS) based on AutoEncoder (AE), a well known deep learning algorithm, to detect network traffic anomalies. I-LADS comes in two versions: (i) I-LADS-v1, that uses filters to independently model IP addresses from the NetFlow dataset, making it possible to train one model for each filtered IP address; and (ii) I-LADS-v2, that uses no filter and therefore a single algorithm is trained for all IP addresses. Experiments have been conducted using a valid dataset containing over two million connections to build a model with multiple features in order to identify the approach that most accurately detects traffic anomalies in the target network. Preliminary results show a promising solution with 99% and 94% of accuracy for the supervised and unsupervised learning approaches respectively.

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. 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).

Network Anomaly detection is an open research that considers the problem of finding patterns in data that do not conform to expected behavior. 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 an *Improved Live Anomaly Detection System* (denoted by I-LADS) based on deep learning algorithms. The approach explores the use of a well known deep learning algorithm: AutoEncoder (hereinafter denoted by AE), to identify abnormal behaviors in the NetFlow traffic captured. AE play a fundamental role in unsupervised learning and in deep architectures for transfer learning and other tasks. They are simple learning circuits aiming to transform inputs into outputs with the least possible amount of distortion (Baldi, 2012). AE have been widely used not only for data compression and visualization, but also for pre-training neural networks. They address issues on the magnitudes of gradients, help finding good local optimums for complex objective functions, and perform well in generalizing data with multiple parameters(Le et al., 2015).

I-LADS is proposed in two distinct versions: one that uses filters to independently model IP addresses from the NetFlow dataset, making it possible to use one model per filtered IP address (I-LADS-v1); and another with no filter that uses a single algorithm to train all IP addresses. Training and testing have been performed using different datasets and the two versions of the proposed solution have been evaluated for

Gonzalez-Granadillo, G., Bedoya, A. and Diaz, R.
An Improved Live Anomaly Detection System (I-LADS) based on Deep Learning Algorithms.
DOI: 10.5220/0010573705680575
In *Proceedings of the 18th International Conference on Security and Cryptography (SECRYPT 2021)*, pages 568-575
ISBN: 978-989-758-524-1

both supervised and unsupervised learning.

Experiments have been conducted using a valid data-set containing over two million connections to build a model with multiple features (e.g., duration, protocol, ports, packets, flows, etc.), in order to identify the approach that most accurately detects traffic anomalies in the target network. The performance of the proposed approach has been evaluated for each of the algorithms. Preliminary results show a promising solution with 99% and 94% of accuracy for the supervised and unsupervised learning approaches respectively (considering the best I-LADS version).

The remainder of this paper is structured as follows: Section 2 discusses related work. Section 3 presents I-LADS and details its architecture, methodology, algorithms used and performance. Section 4 details the experiments performed to train, test and select the most accurate anomaly detection model, and show preliminary results and performance metrics associated to each algorithm used. Finally, conclusions and perspective for future work are presented in Section 5.

2 RELATED WORK

Network Flow Anomaly Detection has been widely studied in the literature. Brauckhoff (Brauckhoff, 2010), for instance, uses histogram-based anomaly detectors to pre-filter a set of suspicious flows and apply association rule mining to extract the flows that have caused a malicious event. The model monitors one of the following attributes: transport protocol, source IP address, destination IP address, source port number, destination port number, packets per flow, bytes per flow, inter-arrival times, flow duration, and TCP flags. Labels that identify when an anomaly has happened are required for evaluating whether an anomaly detection system is accurate or not.

The approach of (Attak et al., 2018) builds upon this idea, and proposes a SHIELD architecture and different ML algorithms to the detection of anomalies using the Netflow traffic protocol. Authors compare the performance of the different algorithms: one-class SVM, Isolation Forest, Local Outlier Factor and DL autoencoders. The autoencoders developed using DL techniques have a very high potential, although the fine tuning is necessary to stable results, obtaining an accuracy score of 87%.

In addition, (Lopez-Martin et al., 2017) propose an IoT network-based intrusion detection method that is based on a conditional variational autoencoder that integrates the intrusion labels inside the decoder layers. The method is able to recover missing features from incomplete training datasets and provides a very high accuracy in the reconstruction process, even for categorical features with a high number of distinct values. As a result, the proposed method is less complex and provides better classification results than other familiar classifiers.

The work of (Mirsky et al., 2018) refers to an unsupervised plug and play NIDS that can learn to detect attacks on the local network. The approach uses autoencoders as an ensemble of neural networks to collectively differentiate between normal and abnormal traffic patterns. It supports feature extraction that tracks patterns of every network channel. Results show the possibility of detecting various attacks with a performance comparable to offline anomaly detectors.

Furthermore, (Nguyen et al., 2019) present a framework for detecting and explaining anomalies in network traffic based on Variational Autoencoder; and a gradient fingerprinting technique for explaining anomalies. Results demonstrate an approach effective in detecting different anomalies as well as identifying fingerprints that are good representations of these various attacks

More recently, (Delplace et al., 2020) propose using ML and DL models for the detection of botnets using Netflow datasets. In their work, authors generated new features extracting those from the Netflow datasets with the objective to improve the models. According to authors, the Random Forest (RF) classifier has the best performance in 13 different scenarios with an accuracy of more than 95%.

One of our previous work, (Gonzalez-Granadillo et al., 2019) we proposed an approach for the analysis of the behavior of IoT devices that generates few signals. This approach uses a one-class SVC algorithm to detect network anomalies with features related to IP addresses (IP source, IP destination, IP distance, IP known and IP unknown). Our proposed approach improves previous works by adding multiple features to the analysis and using other learning algorithms to detect anomalies in the network traffic. Results show significant improvements in the performance of the tool.

3 IMPROVED LIVE ANOMALY DETECTION SYSTEM (I-LADS)

I-LADS results from an adaptation of Convolutional Autoencoders, that instead of processing images to check for anomalies, it processes NetFlow data categorize network traffic into normal or anomalous, based on the modeling of multiple NetFlow parameters. This section details the architecture and main aspects related to the usage and evaluation of the I-LADS.

3.1 I-LADS Architecture

The I-LADS architecture (as depicted in Figure 1) encompasses modules dedicated to data gathering, training of the machine learning models and predictions using the trained models in near real time, processing these data, and generating outputs/results of this processing.

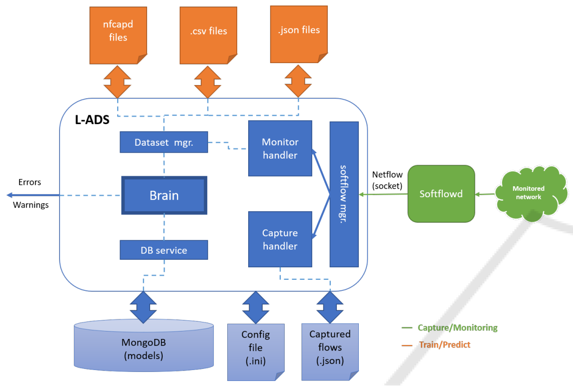

Figure 1: I-LADS Architecture.

Depending on the requested service, I-LADS is able to operate in one of the following modes:

i **Cleaning.** This mode deletes all models stored in the database.

ii **Capturing.** This mode collects traffic from the managed infrastructure and stores the received NetFlow traffic. I-LADS uses the Monitored Network, which includes NetFlow data from both hardware devices (e.g., routers) and software elements (softflow); as well as the Softflowd[1], a flow-based network traffic analyzer capable of exporting Cisco NetFlow data.

iii **Training.** In training mode, the NetFlow traffic is processed by the SoftFlow Manager and sent to the Monitor handler and the Capture handler. The Brain uses a defined learning algorithm to build the model based on the input data and the defined parameter(s) (e.g., IP addresses, port numbers, protocols). Models are then stored in the database by the DB service.

iv **Monitoring.** This mode evaluates in real time the NetFlow traffic received. Once the Softflow Manager receives the traffic, it uses either the Monitor handler or the Capture handler to pre-process the traffic and stored in the database. The Capture handler assumes the traffic does not contain anomalies, whereas the Monitor handler assumes the traffic contains both legitimate and anomalous traffic and redirect the traffic to the Dataset Manager.

v **Prediction.** This mode tests the models against text files containing a dataset of the network traffic. The main objective of the prediction service is to identify anomalous behavior on the network traffic and to generate alarms (error and/or 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.

vi **Exit.** This mode closes the application.

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

The I-LADS output is composed of (a) the models used for the prediction module stored in the Mongo DB. This latter 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; (b) the captured flows generated by the Capture handler during the training and model generation process; and (c) the alarms indicating errors and/or warnings about the anomalous behavior of the analyzed network traffic. The warnings are used when the flows have a certain IP that is not modelled or when there is not a valid model, whereas errors are interpreted as flows classified as anomalous connections.

3.2 I-LADS Data Pre-processing

Table 1 provides a list of features obtained by Softflowd. Neural networks require all input data in numeric values, therefore, the dataset needs to be transformed accordingly. This section details the steps used by I-LADS to prepare the dataset before it is used to build the model.

3.2.1 Version Selection

I-LADS has two available versions (i.e., I-LADS-v1 using filters, and I-LADS-v2 using no filters). Its selection depends on the use case and the required performance.

- **I-LADS-v1.** This version uses filters to independently model the different IP addresses from the NetFLow dataset. Once the dataset generated by Softflowd is retrieved, it is possible to get both the source and destination IPs. For every IP, the flow can be filtered. This implementation reduces the number of flows to each subset,

Table 1: Softflowd Parameters.

Features	Description
Date first seen.	Start time of the flow
Duration.	Duration of the flow
Proto.	Protocols used
Src IP Addr.	Source IP address
Src Pt.	Source Port
Dst IP Addr.	Destination IP address
Dst Pt.	Destination Port
Flags.	TCP flags of the connection
Packets.	N. of packets in the flow
Bytes.	N. of bytes in the flow
Flows.	N. of flows in the connection
Tos.	Type of service

which in turn improves efficiency. By filtering the dataset, we filter the flow of a single IP, which reduces the noise. However, as it generates *k* subsets (*k* corresponds to the number of unique IP addresses), then *k* associated algorithms are generated simultaneously, which will be trained with a subset in a local environment. In this case, it is a local environment from a certain IP. In addition, a further issue can arise related to some subsets containing insufficient flows to train the associated algorithm, which is translated into a bad performance. In order not to lose information, a new dummy variable called *is_source* is created. As a result, 1 is assigned if the IP appears as a source, and 0 if it appears as a destination).

- **I-LADS-v2.** In this version, the Softflowd dataset is not filtered, and a single algorithm will be trained for all the IP addresses. In other words, the algorithm tries to get the similarities between all the features except for the IPs. Unlike the first version, in this case there is just one dataset with an associated algorithm. The dataset may contain flows from a wide variety of devices (e.g., in a home network, we can capture the flows from a smart TV, a printer, a tablet, a mobile phone, and other appliances), which in turn represents different behaviors from each device. Consequently, it is possible to have some noisy flows, but the data processing and analysis will be faster compared to the first version.

3.2.2 Feature Engineering

Feature engineering refers to the creation of more features with the aim to improve its performance. I-LADS has a simple feature engineering with the following features: (i) **Packets_speed.** The number of Packets divided by the Duration; (ii) **Bytes_speed.** The number of Bytes divided by the Duration; (iii) **Packets_per_flow.** The number of Packets divided by the flows, and (iv) **Bytes_per_flow.** The number of Bytes divided by the flows.

3.2.3 Categorical Variables

Categorical features (variables) are transformed into numeric features before they are used as input to the learning algorithm. The protocol variable can take two values: TCP and UDP. The conversion process creates a new binary variable called *proto_UDP* for which a value 1 is assigned if the protocol is UDP and 0 if it is TCP. This generates one feature per value of the categorical variable, in other words, two new binary columns will be generated: *proto_UDP* and *proto_TCP*. All the categorical variables are transformed using the same technique. The new variables generated are called dummy variables.

3.2.4 Standardization

The normalization of all input data is necessary to get a valid dataset, namely train dataset, to use as input in the deep learning algorithm. After the transformation in the dataset is performed, we obtained a standardized and clean dataset containing only numeric features which can be used as input to train the algorithm and build the corresponding model.

3.3 I-LADS Training and Prediction

During the training, the algorithm is fed with legitimate traffic and the features already defined in Section 3.2 (e.g., IP addresses, port numbers, protocols, etc.) to build a model for testing and predictions with a predefined time window that can vary according to the size of the data to be analyzed (e.g. 5-second window).

Our research focuses on a semi-supervised method using a well known deep learning algorithm, namely Autoencoder (AE). This latter is a type of Artificial Neural Network (ANN) used to learn efficient data coding in an unsupervised manner. AEs have a symmetric structure (i.e., input and output layers have the same dimension), where the middle layer represents an encoding of the input data. Particularly, AEs are trained to reconstruct their input onto the output layer, while verifying restrictions which prevent them from simply copying the data along the network (Charte et al., 2018).

AEs consist of at least two main parts: (i) **An encoder part** aiming to retrieve the input dataset and compress the information; and (ii) **A decoder part** aiming to retrieve the compressed information and decode it to obtain the initial train dataset. In the decoder part, the input dimension is the same as the train dataset dimension, while the output dimension is compressed, whereas in the decoder part the input and output must be the same.

When the AE evaluates an anomalous connection, the transformed connection should be very different from the initial input. If the resulting value is bigger than a specific value or threshold, the connection will be categorized as an anomalous connection. Otherwise, if the value is lower than the threshold (or close to zero) it will be categorized as a legitimate connection. The threshold is set through experimentation, i.e., in an unsupervised learning problem, this threshold is predetermined, whereas in a supervised learning problem, the threshold could be adjusted to improve performance.

Since the threshold can be adjusted using the labels of a supervised learning problem, this is considered a semi-supervised learning process. The key part of this approach is to define an appropriate threshold to determine which connections are anomalous or legitimate.

4 I-LADS TESTING AND VALIDATION

Validation comprises the training and generation of prediction models, as well as a set of tests performed with different features that characterize NetFlows aiming to analyzing aspects such as flow duration, packet size, ports and protocols used in the communications, among other features, to determine whether connections are legitimate, or anomalous.

4.1 Training Data

In order to check the validity of the Autoencoder as a classification algorithm, we used the CIDDS-001 dataset, which is part of the Coburg Intrusion Detection datasets (CIDDS) (Ring et al., 2017), a set of labelled Flow-based public datasets. This dataset contains unidirectional NetFlow data consisting of traffic generated by emulating small business environment consisting of two main networks: (i) OpenStack composed of four internal subnetworks; and (ii) External Server, consisting of a file synchronization and a web server deployed on the Internet to capture real and up-to-date traffic.

The CIDDS-001 dataset consists of large number of traffic instances (172,839 instances from OpenStack and 153,026 instances from the External Server) from which communication is captured over the course of a month. The dataset provides a list of attributes including those depicted in Table 1 and others classifying the connection as normal, attacker, victim, suspicious and unknown, as well as identifying the type of attack (e.g., Brute Force, DoS, PortScan, PingScan).

The CIDDS-001 dataset has been split into two subsets: (i) Training, composed of the data from week 4 because it has no attacks; and (ii) Testing, composed of the data from week 2 because it contains both legitimate and attack instances. Given that our algorithm classifies the connections as legitimate or anomalous, all connections labeled as victim or attacker have been treated as anomalous.

Considering that the CIDDS is labeled, we adapted the model to be used both as supervised and unsupervised learning problems. The former is used when labels are included in the dataset, whereas the latter is used with no labels. The unsupervised learning problem should be a more realistic scenario. For our tests, we only use internal connections because they are a representative sample of our expected real-life scenario.

4.2 Experimentation

Validation comprises the training and generation of prediction models, as well as a set of tests (experiments) performed with different features that characterize NetFlows aiming to measure aspects e.g., number of flows/packets/bytes per second, among others, to determine whether connections are legitimate or they are considered as anomalous since they fall outside the boundaries of the model.

In order to validate the models built by the I-LADS and to identify the features that best contribute in the detection of normal and anomalous connections, we performed a series of tests with multiple features, using three Recurrent Neural Networks (RNN) layers having a size that directly depends on the input data size. The training data passes through the model 10,000 entries (batch = 10,000) at a time until all of the entries have passed through, making the end of one epoch, and the process is repeated 500 times (epoch = 500). The algorithm encodes 30 features into 15 and then, decodes them to output 30 features. The remainder of this section details two of the main experiments and the preliminary obtained results.

4.2.1 Experiment 1: I-LADS-V1

This experiment evaluates a dataset composed of 2,232,715 internal connections using both supervised learning (labeled data) and unsupervised learning (no labeled data). This experiment focuses on analyzing the performance of the I-LADS-V1 (which filters IP addresses during the training) in detecting normal and anomalous behavior on the 2,232,715 internal connections. The unsupervised learning scenario requires a threshold to be determined before the training of the Autoencoder, however, this version of the I-LADS does not define any threshold because its analysis requires the IP addresses.

4.2.2 Experiment 2: I-LADS-V2

This experiment uses the same dataset used in experiment 1 to evaluate the I-LADS-V2 (using no filters on the data training).The dataset has been tested against the two versions of the I-LADS and has been labeled as legitimate (with a value of one) or anomalous (with a value of zero). The unsupervised learning scenario requires a threshold to be determined before the training of the Autoencoder. For this version of I-LADS (using no filters), we used the results of the supervised learning approach to determine the value of the threshold that provides the highest performance. All metrics have been evaluated using different thresholds (in this experiment from 0.1 until 100) and the best performance is obtained with a threshold equals to 10.0.

4.3 Preliminary Results

This section presents preliminary results of the experiments performed with both version of our proposed solution by grouping them in two categories: supervised and unsupervised learning results.

4.3.1 Supervised Learning Results

The performance of the I-LADS for a supervised learning scenario using Autoencoder against the CIDDS-001 dataset is summarized in Table 2. As can be seen, both versions of the I-LADS tool have been analyzed. for the case of normal detection, v1 seems to perform better than v2 with higher accuracy, precision and F1-Score values.

A similar result is obtained while comparing the performance of detecting anomalous connections on the two versions of the I-LADS. While I-LADS-v1 has all performance values close to one (meaning that is able to predict all normal and anomalous connections accurately), the performance of I-LADS-v2 is decreased on the precision and F1-score on the detection of normal connections. This is mainly due to the higher false positive rate obtained in the detection of normal connections.

For the case of the I-LADS-v2, since there is no filtered IP address, individual results are unable to be obtained. Only global performance is possible to be drawn for these experiments. Note that although the performance of the I-LADS-v2 is lower than the I-LADS-v1, the model shows a performance higher than 0.9 on the detection of anomalous connections.

4.3.2 Unsupervised Learning Results

Table 3 depicts the results of the unsupervised learning approach against the CIDDS-001 dataset used in the experiments. Note that since no filters are used, no additional tests have been performed using I-LADS-v2, therefore results are the same for supervised and unsupervised approaches in detecting both normal and anomalous connections.

It is important to highlight that the performance of the unsupervised learning model of I-LADS-v2 is higher than the I-LADS-V1, with accuracy, recall, and F1-score values approaching to one. It is important to note that the performance is slightly better for the anomalous detection compared to the normal detection.

When dealing with high volumes of data, it is therefore recommended to use I-LADS-v1 for supervised learning approaches and I-LADS-v2 for unsupervised learning approaches. This is mainly due to the fact that when using no labeled data, the model decreases its performance as the number of connections included in the training dataset increases.

5 CONCLUSION AND FUTURE WORK

We have proposed I-LADS, an Improved Live Anomaly Detection System that uses a deep learning algorithm called Autoencoders to detect network traffic anomalies based on the analysis of multiple NetFlow features (e.g., IP addresses, ports, protocols, packets size, packets speed, etc.). I-LADS comes in two distinct versions: (i) I-LADS-v1, that filters the flow of every IP address; and (ii) I-LADS-v2, that uses no filters for the IP addresses and trains the dataset with a single algorithm in charge of identifying the similarities among the NetFlow features.

The proposed model has been tested against a Flow-based public dataset containing over two million connections using both supervised and unsupervised learning approaches. The algorithm encodes and decodes all features to compute the difference from the input and output and classify connections as normal or anomalous based on this difference.

Table 2: Performance Results Supervised Learning.

Approach	TP	TN	FP	FN	Accuracy	Recall	Precision	F1-Score
V1(Normal)	0.1736	0.8193	0.0038	0.0033	0.9929	0.9816	0.9784	0.9800
V2(Normal)	0.1329	0.8041	0.0629	0.0000	0.9371	1.0000	0.6786	0.8085
V1(Anomalous)	0.8193	0.1736	0.0033	0.0038	0.9929	0.9953	0.9960	0.9957
V2(Anomalous)	0.8041	0.1329	0.0000	0.0629	0.9371	0.9274	1.0000	0.9623

Table 3: Performance Results Unsupervised Learning.

Approach	TP	TN	FP	FN	Accuracy	Recall	Precision	F1-Score
V1(Normal)	0.1766	0.5977	0.0008	0.2249	0.7742	0.4398	0.9954	0.6101
V2(Normal)	0.1329	0.8041	0.0629	0.0000	0.9371	1.0000	0.6786	0.8085
V1(Anomalous)	0.5977	0.1766	0.2249	0.0008	0.7742	0.9986	0.7266	0.8411
V2(Anomalous)	0.8041	0.1329	0.0000	0.0629	0.9371	0.9274	1.0000	0.9623

I-LADS-v1 has shown a better performance on analysis using supervised learning algorithms, whereas I-LADS-v2 shows a better performance on evaluations using unsupervised learning approaches. This is mainly due to the fact that when using no labeled data, the model decreases its performance as the number of connections included in the training dataset increases. It is therefore preferable to use I-LADS-v2 when dealing with high volumes of data in an unsupervised scenario.

The main limitation identified in our proposed approach is in terms of explainability. We need to make use of techniques to compute the difference between inputs and outputs, making it possible to obtain a diff value for each analyzed feature (e.g., number of bytes per flow). If such a difference is too big, it could lead to understand the reason why a connection has been classified as anomalous.

Note that for our experiments we have used 80% of the samples as the training dataset, and 20% of the samples for testing and predictions. However, the time needed to obtain results using our model, highly depends on the size of the dataset, which in some cases may require several minutes to classify connections. In such a case, although the testing is executed in real time, predictions are expected to be obtained win near-real time.

It is important to highlight that previous versions of the LADS were best suited for the behavior analysis of IoT devices (e.g., smart security systems, monitoring devices, jamming detectors, smart thermostats, etc.) that generates one or few signals, and whose behavior can be easily modelled by the algorithm. Our proposed approach, although also suitable to model more complex devices (e.g., workstations, servers, mobile phones, etc.), it is best suited to model tools with a specific behavior. We expect the model decreases its performance when analyzing devices with heterogeneous behaviours.

Future work will consider to evaluate different attack scenarios, use other features to evaluate traffic behavior and other ML and DL algorithms to compare their performance. In addition, we will use other datasets and different neural network architectures aiming at improving interpretability of results. The ultimate goal is to build customized models for a wide variety of attacks to be tested in different scenarios.

ACKNOWLEDGEMENTS

The research in this paper has been partially supported by the European Commission through the STOP-IT project, (Grant Agreement No. 740610).

REFERENCES

Attak, H., Combalia, M., Gardikis, G., Gastón, B., Jacquin, L., Katsianis, D., Litke, A., Papadakis, N., Papadopoulos, D., Pastor, A., Roig, M., and Segou, O. (2018). Application of distributed computing and machine learning technologies to cybersecurity. In *C& ESAR Conference on Artificial Intelligence and Cybersecurity*.

Baldi, P. (2012). Autoencoders, unsupervised learning, and deep architectures. In *Workshop on Unsupervised and Transfer Learning*, pages 37–50.

Brauckhoff, D. (2010). Network traffic anomaly detection and evaluation. PhD Thesis available at http://e-collection.library.ethz.ch/eserv/eth:1078/eth-1078-02.pdf.

Charte, D., Charte, F., García, S., del Jesus, M. J., and Herrera, F. (2018). A practical tutorial on autoencoders for nonlinear feature fusion: Taxonomy, models, software and guidelines. *Information Fusion*, 44:78–96.

Delplace, A., Hermoso, S., and Anandita, K. (2020). Cyber attack detection thanks to machine learning algorithms. ArXiv preprint, availabble at https://arxiv.org/abs/2001.06309.

Gonzalez-Granadillo, G., Diaz, R., Medeiros, I., Gonzalez-Zarzosa, S., and Machnicki, D. (2019). Lads: A live anomaly detection system based on machine learning methods. In *Conferene on Security and Cryptography, SECRYPT*.

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.

Le, Q. V., Brain, G., and Inc, G. (2015). A tutorial on deep learning part 2: Autoencoders, convolutional neural networks and recurrent neural networks. Available at https://cs.stanford.edu/ quocle/tutorial2.pdf.

Lopez-Martin, M., Carro, B., Sanchez-Esguevillas, A., and Lloret, J. (2017). Kitsune: An ensemble of autoencoders for online network intrusion detection. *Sensors Journal*, 17.

Mirsky, Y., Doitshman, T., Elovici, Y., and Shabtai, A. (2018). Kitsune: An ensemble of autoencoders for online network intrusion detection. *Sensors Journal*.

Nguyen, Q. P., Lim, K. W., Divakaran, D. M., Low, K. H., and Chan, M. C. (2019). Gee: A gradient-based explainable variational autoencoder for network anomaly detection. In *Conference on Communications and Network Security (CNS)*. IEEE.

Ring, M., Wunderlich, S., Gruedl, D., Landes, D., and Hotho, A. (2017). Creation of flow-based data sets for intrusion detection. *Journal of Information Warfare (JIW)*, 16:40–53.

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*.