Quality In / Quality Out: Data quality more relevant than model choice in anomaly detection with the UGR'16

José Camacho
Dept. Signal Theory, Networking and Communication, University of Granada
Granada, Spain
josecamacho@ugr.es

Katarzyna Wasielewska
Dept. Signal Theory, Networking and Communication, University of Granada
Granada, Spain
k.wasielewska@ugr.es

Pablo Espinosa
Dept. Signal Theory, Networking and Communication, University of Granada
Granada, Spain
pabloespinosa@correo.ugr.es

Marta Fuentes-García
COSCYBER
FIDESOL
Granada, Spain
mfuentes@fidesol.org

***Abstract*—Autonomous or self-driving networks are expected to provide a solution to the myriad of extremely demanding new applications in the Future Internet. The key to handle complexity is to perform tasks like network optimization and failure recovery with minimal human supervision. For this purpose, the community relies on the development of new Machine Learning (ML) models and techniques. However, ML can only be as good as the data it is fitted with. Datasets provided to the community as benchmarks for research purposes, which have a relevant impact in research findings and directions, are often assumed to be of good quality by default. In this paper, we show that relatively minor modifications on the same benchmark dataset (UGR'16, a flow-based real-traffic dataset for anomaly detection) cause significantly more impact on model performance than the specific ML technique considered. To understand this finding, we contribute a methodology to investigate the root causes for those differences, and to assess the quality of the data labelling. Our findings illustrate the need to devote more attention into (automatic) data quality assessment and optimization techniques in the context of autonomous networks.**

***Index Terms*—Netflow, UGR'16, anomaly detection, data quality**

I. Introduction

There is an increasing interest in the development of new machine learning (ML) methods to improve the performance of communication networks [1]. ML tools can only be as good as the data they are trained on, reason why we need high-quality datasets [2] [3]. However, while the process of model optimization and the development of new ML methods have received the full attention of the community, techniques to assess data quality are scarce and often ignored [4].

In this paper, we show that the impact of minor data modifications prior to modelling with ML can be indeed more relevant than the specific ML method used. These modifications include mild changes on how traffic features were computed, whether or not data was anonymized, and the set of observations that were considered for model fitting and testing. This case study illustrates that the research community needs to look more into data quality assessment and optimization.

Our main contribibutions are:

- We derive four variants of a benchmark dataset in network anomaly detection, by applying minor differences in the data treatment. We perform anomaly detection using these variants with two very different ML methodologies, finding negligible differences in performance between the ML variants but significant differences among the dataset variants.
- We develop an analysis methodology to investigate the root causes of the performance differences found. Applying this methodology to the case study provides a full understanding of the differences, which allows us to obtain a better picture of when these are relevant and/or when they are due to labelling inaccuracies (in particular, unlabelled anomalies).

The paper is organized as follows. Section II presents the related work. Section III introduces the case study under analysis, the preprocessing and data selection steps, and ML methods considered. Section IV presents the experimental results and Section V draws the conclusions.

II. Related Work

Due to the various methods of collecting and preparing datasets, and the problems associated with these processes (for example, device or human errors), it is necessary to assess the quality of the dataset for ML modeling. We can define quality as the degree to which a dataset fulfills the

This work was supported by the Agencia Estatal de Investigación in Spain, grant No PID2020-113462RB-I00, and the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 893146.

requirements for its intended use. Data quality is a multi-dimensional concept [5], and the following dimensions have been proposed: accuracy, completeness, validity, timeliness, consistency, correctness, uniqueness, reliability, and others. These indicators help understand the data, but while intuitive, they are difficult to measure in practice [6] [7]. Furthermore, the meaning and importance of each dimension and its metric varies from application to application [8]. Several of these indicators have been adapted in the networking area, but there is no general framework to assist in assessing the quality of network datasets [9]. Yet, the assessment of data quality should be a priority for the network community, as a recent survey on intrusion datasets points out [10].

The research on how data preprocessing affects model quality is gaining momentum in the community. The influence of data normalization and dimensionality reduction is studied in [11] in the intrusion detection NSL-KDD dataset, a refined version of the (unfortunately unrealistic and outdated) KDD'99 dataset. Gonzalez [12] proposes a method to assess the influence of specific data preparation steps on the model performance. Lauría and Tayi [13] evaluate the effect of noise in the KDD'99 dataset. Chen et al. [14] perform a very complete analysis to assess both data quality and the choice of ML models in intrusion detection.

There is an inherent connection of data quality and labelling quality. The relevance and challenges of the process of labelling network traffic datasets are emphasized in [15]. Landauer et al. [16] introduce a framework for automatic labelling of datasets to train host intrusion detection systems. Camacho and Wasielewska [4] contribute a method of labels permutation in order to estimate the quality of association of a dataset with a specific labelling.

Our findings in this paper support the need to study the quality of data and labelling in network datasets. Labelling quality is of specially relevance given that a wrong labelling can detriment our perception of model quality and thus the potential conclusions derived from a study. Thus, unlike aforementioned works on data preprocessing, our work introduces a methodology to perform a deep analysis and get a full understanding on how data characteristics affect model performance. This interpretation methodology is connected to the aims of eXplainable Artificial Intelligence (XAI) [17]. A relevant advantage of such methodology is that it can lead to identify labelling errors, rather than accepting the labelling correctness for granted. Furthermore, we do our analysis with a real network dataset, which provides an excellent example to the community of why data and labelling quality should not be disregarded in practical applications.

III. Materials and Methods

In the following sub-sections we present the case study under analysis, the data parsing/preparation, the four dataset variants and the two variants of machine learning considered, the performance measures for evaluation and the strategy to explain the results.

TABLE I
Characteristics of the calibration and the test sets.

Feature	Calibration	Test
Capture start	10:47h 03/18/2016	13:38h 07/27/2016
Capture end	18:27h 06/26/2016	09:27h 08/29/2016
Attacks start	N/A	00:00h 07/28/2016
Attacks end	N/A	12:00h 08/09/2016
Number of files	17	6
Size (compressed)	181GB	55GB
# Connections	$\approx$ 13,000M	$\approx$ 3,900M

A. The UGR'16 Dataset

The UGR'16 dataset [18][1] was captured from a real network of a tier 3 Internet Server Provider (ISP). The data collection was carried out with Netflow between March and June of 2016 under Normal Operation Conditions (NOCs), meaning that the network was used normally by the ISP clients. This allowed to model and study the normal behavior of the network, and to unveil certain anomalies such as SPAM campaigns. The flows of the dataset were labelled indicating if they were ”background” (regarded as legitimate flows), or ”anomalies” (identified as non-legitimate flows).

In addition, another capture was made between July and August of 2016, including some controlled attacks that were launched to obtain a test dataset for validation of anomaly detection algorithms. To do this, twenty five virtual machines were deployed within one of the ISP sub-networks. Five of these machines attacked the other twenty. The type of attacks were *Denial of Service* (DOS), *port scanning* in two modalities: either from one attacking machine to one victim machine (SCAN11) or from four attacking machines to four victim machines (SCAN12), and *botnet traffic* (NERISBOTNET). These attacks were launched during twelve days in different periods of time, following either planned or random scheduling, and with real background traffic.

This dataset has the main benefit that data are collected from a real network and allow us to validate algorithms in a realistic manner, where background traffic follows day/night and weekday/weekend patterns. As of today, the UGR'16 has been referenced in more than 150 research papers (according to Google Scholar) and it can be considered a benchmark in the research of anomaly detection in real traffic data for cybersecurity. The general characteristics of the dataset are provided in Table I.

B. Data parsing

A custom step of the ML workflow, referred to as feature engineering, is to transform raw data information into quantitative variables or features. This is a tough task due to the unstructured nature of several system log formats and network traces, which makes it difficult to parse the information in an automated manner. Moreover, selecting which network features are suitable for analysis is not trivial. Traffic data

[1] Dataset available online at https://nesg.ugr.es/nesg-UGR'16/

TABLE II
UGR'16 DATASET VARIANTS.

Label	Training	Type of flows	Anonymized flows
UGR'16v1	March to June	Unidirectional	No
UGR'16v2	March to May	Unidirectional	No
UGR'16v3	March to May	Bidirectional	Yes
UGR'16v4	March to May	Unidirectional	Yes

is ordered in time, but characteristics such as groups of IP addresses, destination ports and size of the packets in the network should be considered to maintain a high degree of observability in the analysis.

The pioneering work of Lakhina et al. [19] in anomaly detection with multivariate techniques (in particular with Principal Component Analysis, PCA) approached feature engineering by defining variables as counts of packets and bytes, thus directly obtaining quantitative variables from Netflow records. Camacho et al. [20] extended this definition to the *feature-as-a-counter* (FaaC) approach, in which the variables represent counters for the number of times a particular traffic feature takes place in a time window. This makes it possible to obtain quantitative variables of very different nature, *e.g.*, variables for traffic volume within a particular range of IPs or ports. Moreover, the window size acts as a configurable sampling interval, reducing the initial data size significantly and simplifying the data analysis.

We make use of the FaaC approach in this work. Using this approach, we perform anomaly detection at 1 minute intervals rather than at flow level. A total of 134 features are extracted per interval. The process of feature extraction is based on two steps: i) binary files are transformed to flow-level csv files with the nfdump tool, and ii) csv files are transformed to feature vectors with the FCParser [21]. In our case, using parallelization with 16 CPUs, the features of a daytime were extracted in approximately 3h, and the complete dataset can be transformed in app. 15 days of processing. Given that flows are aggregated at 1 minute intervals, test observations are categorized as normal when only background traffic is present, and as anomalous when attack flows are included with background traffic. For more details on the FaaC approach, please refer to reference [21].

C. Dataset variants

In the context of this paper, we considered four variants of the UGR'16, described in Table II:

- In the first variant (UGR'16v1), the original (non-anonymized) Netflow logs for the entire NOC period (from March to June) were employed. This corresponds to the same data used in previous works [21].
- Subsequently to this contribution, it was found [22] that the training data corresponding to June included real anomalies that hamper the ability of detection of the botnet attack in the test set. Leveraging this finding, we consider a second version (UGR'16v2) in which the training data corresponds only to the period from March to May.
- In both previous versions (UGR'16v1 and UGR'16v2), unidirectional Netflow flows were considered. Unidirectional flows may complicate the interpretation of the results. For this reason, we decided to repeat the feature generation process using bidirectional flows (in nfdump), in this case considering the anonymized flows available online. This is the third version of the dataset (UGR'16v3), and it shares with the second version that June is not included in the training data.
- Finally, and to distinguish the influence of anonymization from the use of bidirectional or unidirectional flows, we considered a last version (UGR'16v4) equivalent to version 3 but with unidirectional flows.

All previous versions are based on the use the same approach for feature engineering described in previous subsection: FaaC. Please note that the vast majority of the literature that makes use of UGR'16 is outside the research group at UGR. Thus, most research has been performed from anonymized data, and therefore is intuitively closer to versions 3 and 4.

The consideration of the previously described four versions of UGR'16 allows us to determine the impact of some data preprocessing steps on the model quality for anomaly detection, in particular:

- The selection of the set of training data (by comparing performance results between UGR'16v1 and UGR'16v2).
- The effect of bi- or uni-directional flows (by comparing performance results between UGR'16v3 and UGR'16v4).
- The effect of anonymization (by comparing performance results between UGR'16v2 and UGR'16v4).

D. Anomaly Detection Techniques

To compare the influence of data preprocessing methods in the anomaly detection performance against the influence of the specific ML methods used, we consider two very different tools: the Multivariate Statistical Network Monitoring (MSNM) [23] and the one-class support vector machine (OCSVM) [24], [25] based on radial basis functions (RBF), the most extended kernel choice. The former is a linear multivariate approach, and therefore it is specially suited to handle the highly multivariate nature of the FaaC features. The latter is a non-linear tool, and therefore has the advantage to model non-linear behaviour in the model of normal traffic. Thus, both methods have very different features that could, in principle, affect performance in a significant way.

E. Performance evaluation

To test the anomaly detection performance with the different data and model variants, we compute the false positive rate (FPR) and true positive rate (TPR) in the labeled test set, and in turn the Receiver Operating Characteristic (ROC) curves, that show the evolution of the TPR versus the FPR for different values of the anomaly detection threshold. We selected this option since in the context of network security, maintaining the balance between TP and FP is relevant in practice [26], [27]. A practical way to compare several ROC curves is with the Area Under the Curve (AUC), a scalar that quantifies the quality of the anomaly detector. An anomaly detector should present an AUC as close to 1 as possible, while an AUC around 0.5 corresponds to a random classifier.

F. Strategy for explanation of the results

We will use the Univariate-Squared (U-Squared) statistic [28] to shade light on the model performance differences when using different dataset versions. The U-Squared has shown to have superior diagnosis ability than other multivariate diagnosis tools and it has two main advantages: it is extremely simple and it is model agnostic[2].

To diagnose a certain anomaly type, represented by a set of observations $\mathbf{x}_n$ for $n \in \{1, ..., N\}$, we compute the vectors of sample means $\boldsymbol{\mu}$ and standard deviations $\boldsymbol{\sigma}$ of a reference dataset composed of (ideally) only non-anomalous observations, where $\mathbf{x}_n$, $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ are row vectors of length the number of features. In our case, this reference is represented by any of the versions of the UGR'16. Then, for each anomalous observation $\mathbf{x}_n$, the U-Squared follows:

$$\mathbf{d}_n^2 = ((\mathbf{x}_n - \boldsymbol{\mu})/\boldsymbol{\sigma}) \cdot |(\mathbf{x}_n - \boldsymbol{\mu})/\boldsymbol{\sigma}|^T \qquad (1)$$

The accumulated U-Squared for the set of anomalous observations simply follows:

$$\mathbf{d}^2 = \sum_n \mathbf{d}_n^2 \qquad (2)$$

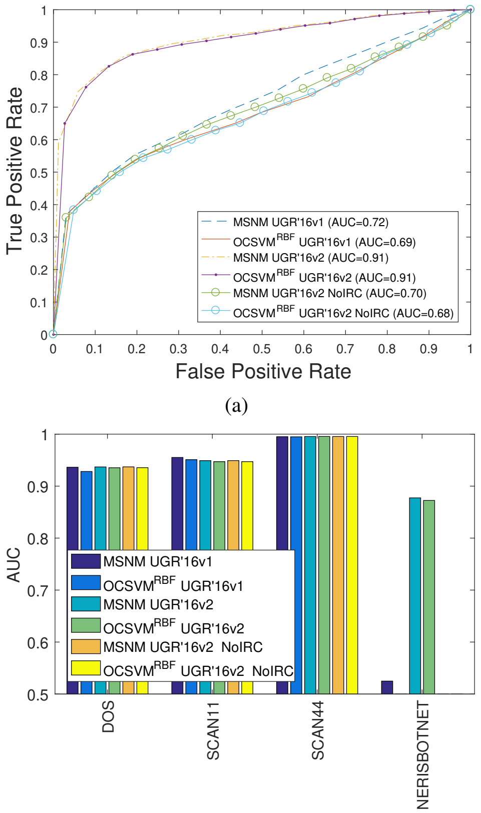
where Vector $\mathbf{d}^2$ is also of length the number of features, and can be conveniently visualized using a bar plot. In this bar plot, high magnitude bars (either positive or negative) highlight the main differences of the considered attack from the reference. Positive (negative) bars mean that the attack show significant higher (lower) values for the specific features than the reference.

The U-Squared statistic, like other diagnosis solutions [29], provides a discriminative pattern for the attack in comparison to the reference. This pattern can be further studied to determine whether the reference dataset is of good quality to train anomaly detection models able to detect the attack or not. From the U-Square we can identify a subset of features in this pattern of detection, and then we can proceed using statistical means to analyze whether those features have good detection capability for the attack. We will show that this approach can provide a full understanding of the performance differences between dataset variants in our case study.

IV. Experiments and Results

A. Influence of the set of observations

Fig. 1 shows the comparison of the two anomaly detectors (MSNM and OCSVM) when trained with the datasets UGR'16v1 and UGR'16v2, and with a sub-version of UGR'16v2 (UGR'16v2 NoIRC) that will be discussed later. Fig. 1(a) presents the general ROC curves, obtained for the four types of attacks, and Fig. 1(b) represents the AUCs per attack type. Performance differences between the two anomaly detectors are minor in all cases. However, there is a huge difference with respect to including June in the training data (UGR'16v1) or not including it (UGR'16v2). This difference can be mapped to one specific attack type, the NERISBOTNET. We hypothesize that this difference is mainly caused by the anomaly detected in the background traffic of June, related to suspicious activity through a MIRC channel [22].



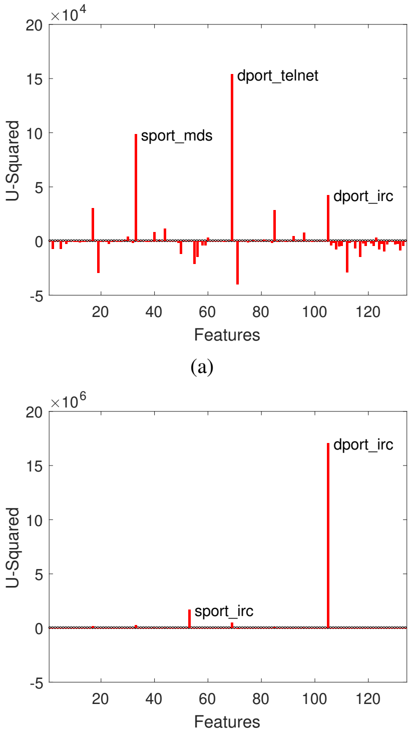
(a)

(b)

Fig. 1. ROC curve (a) and attack-type based AUC results (b) for the data parsed from original unidirectional flows in UGR'16v1 and UGR'16v2, and for a variant of the latter with no IRC features (UGR'16v2 NoIRC).

To check our hypothesis, we compute the U-Squared statistic for the observations in the test set that contain flows of the NERISBOTNET attack, and using as a reference UGR'16v1 and UGR'16v2, respectively. This is shown in Fig. 2. When using UGR'16v1 as a reference (Fig. 2(a)), we find that the NERISBOTNET attack is mainly characterized by an excess in 3 out of the 134 features: $sport\_mds$, $dport\_telnet$ and $dport\_irc$. This suggests that the number of flows with source port MDS, with destination port TELNET and with destination port IRC are generally higher in observations where NERISBOTNET attacks are taking place. However, when we use UGR'16v2 as a reference (Fig. 2(b)), the NERISBOTNET

[2] While the U-Squared is theoretically model agnostic, it is consistent with any linear multivariate model with squared detection statistics, like MSNM.



(a)

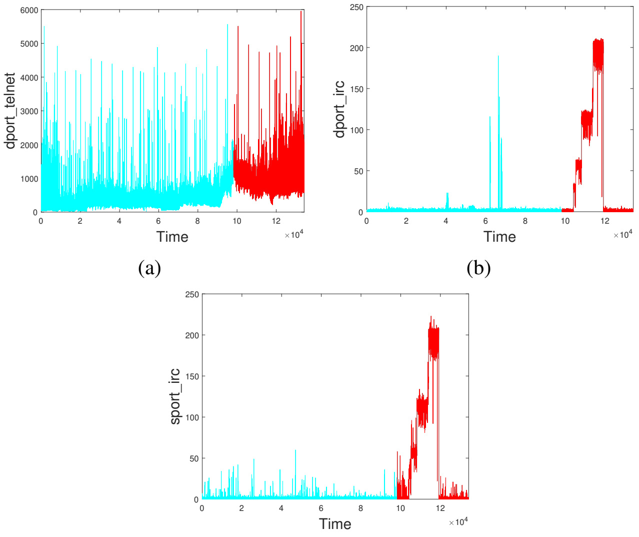
(b)

Fig. 2. Comparison of U-Squared statistics for the NERISBOTNET attack using as a reference UGR'16v1 (a) and UGR'16v2 (b).

attack is mainly characterized by the amount of flows to or from the IRC port[3]. This difference between the U-Squared patterns found with the two reference datasets implies that ML models trained from them will have different means to detect the NERISBOTNET attack. These differences affect performance, as seen in the AUC results.

To further investigate the reason behind the performance differences when using UGR'16v1 and UGR'16v2 as a reference, we represented in Fig. 3 the time series of the training data from March to June for a set of selected features, previously highlighted by the U-Squared. All features present a change of tendency in June, which is specially clear in the case of IRC features. The latter show the suspicious activity in the MIRC channel found in [22]. When June is included in the reference (UGR'16v1), we are telling the anomaly detection models that this type of behaviour is normal, and that future similar events should not be flagged as an anomaly. This is the reason why, when using UGR'16v1 as a reference, the IRC activity is not the most relevant feature to characterize the NERISBOTNET attack (Fig. 2(a)).

Fig. 4 presents boxplots to compare the distribution, in the test set, of the normal vs the NERISBOTNET observations in the same selected features of Fig. 3. We also include the result of a t-test to check whether there is statistical evidence that the NERISBOTNET attack does present higher content in the corresponding feature. Feature *dport_telnet*, highlighted when UGR'16v1 is the reference, does not show statistical significant differences between normal and NERISBOTNET observations. Clearly, including the anomaly in June as "normal data" makes the detectors to incorporate this type of activity in the normality model, and therefore prevents them to detect it in future traffic. Therefore, this feature (and in general UGR'16v1) will allow a low detection ability of the attack. However, all IRC features do show statistical significant differences. Therefore, we can conclude that models that use UGR'16v2 as a reference will detect the presence of NERISBOTNET attacks as significant changes in the IRC features, and will yield a high detection ability. This conclusion is

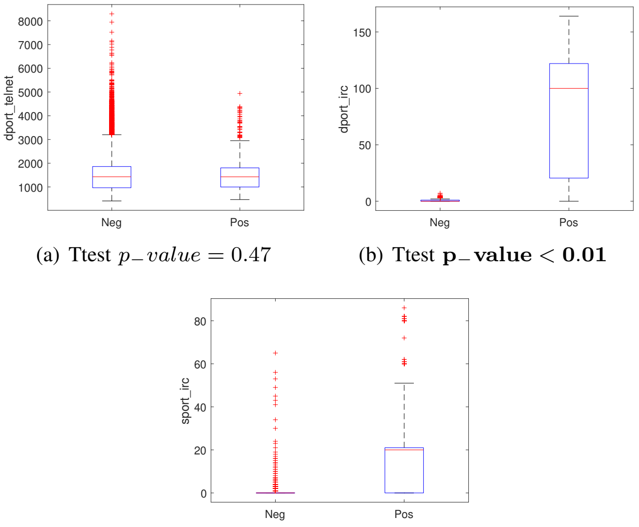


(a)

(b)

(c)

Fig. 3. Time series from March to May (blue light color) and June (red dark color) for features: dport_telnet (a), dport_irc (b) and sport_irc (c).



(a) Ttest $p\_value = 0.47$

(b) Ttest $\mathbf{p\_value < 0.01}$

(c) Ttest $\mathbf{p\_value < 0.01}$

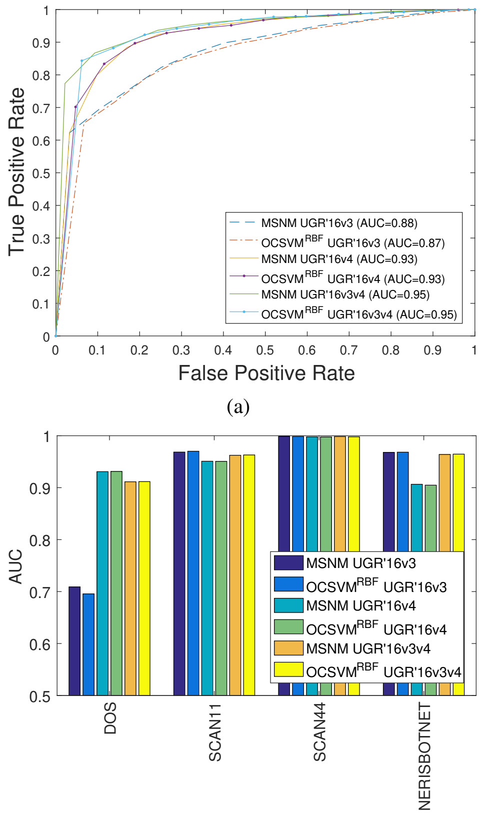
Fig. 4. Boxplots of selected features in background traffic (Negative) versus NERISBOTNET traffic (Positive).

[3] Recall both UGR'16v1 and UGR'16v2 use uni-direction flows. This means that the flows in the direction from the server to the client identify the server port as the source of the communication.

further supported by the fact that if we take UGR'16v2 as a reference, but we delete the IRC features $sport\_irc$ and $dport\_irc$ from the data, the detection of NERISBOTNET is poor, as illustrated in Fig. 1 with the results associated to the label "UGR'16v2 NoIRC". Finally, we also inspected the raw flows with nfdump, and found a massive use of IRC port 6667 in the NERISBOTNET attacks, which is consistent with our observations.

This example supports the claim that anomaly detection requires careful data quality assessment in terms of unsupervised identification of suspicious patterns in data, which has deserved little attention in the community but can be principal in the context of autonomous networks. In this real example, the proper selection of observations (and features) was by far more relevant than the choice of the ML method employed.

B. Bidirectional vs Unidirectional flows

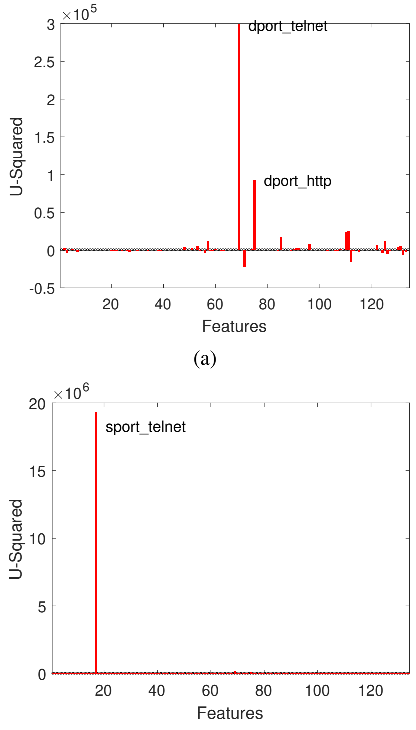


(a)

(b)

Fig. 5. ROC curve (a) and attack-type based AUC results (b) for the data parsed from anonymized bidirectional (UGR'16v3) and unidirectional (UGR'16v4) flows, and a combination of both (UGR'16v3v4).

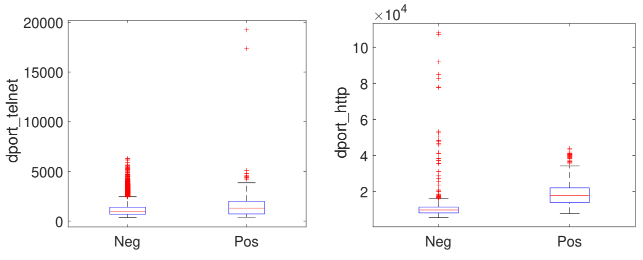
Fig. 5 presents the performance results of the anomaly detectors in UGR'16v3 and UGR'16v4, and a combination of both datasets that will be discussed later. In all situations, the differences between the two detectors, MSNM and OCSVM, is again negligible. Performance differences are observed between the use of bidirectional and unidirectional flows, in favour of the latter. In this case, this difference is mainly mapped to the DOS attacks. Therefore, like in the previous comparison, relatively minor decisions on data preparation (in this case whether or not use an nfdump flag during flows parsing) impact more in the performance than the choice of the ML tool. Fig. 5(b) also shows that bidirectional flows are indeed slightly better in the detection of NERISBOTNET, what suggests that the best detection performance in this case is attack specific.



(a)

(b)

Fig. 6. Comparison of U-Squared statistics for the DOS attack using as a reference UGR'16v3 (a) and UGR'16v4 (b).



(a) Ttest $\mathbf{p\_value} < \mathbf{0.01}$

(b) Ttest $\mathbf{p\_value} < \mathbf{0.01}$

Fig. 7. Boxplots of selected features in background traffic (Negative) versus DOS traffic (Positive) in UGR'16v3.

To shade some light into the observed differences in the detection of DOS attacks, we computed the U-Squared for the observations including DOS attacks using UGR'16v3 and UGR'16v4 as references (Fig. 6). Again, we find different patterns of characterization depending on the reference dataset. Using bidirectional flows, the DOS attacks are characterized by flows with destination ports HTTP and TELNET. Statistical significant differences between test normal observations and those containing DOS attacks confirm this characterization (Fig. 7). However, when we look into the raw flows labelled as DOS attacks with nfdump, these flows only show destination port HTTP. The correlation between DOS attacks and TELNET activity is confirmed in Fig. 8. The Figure shows that every time there is a DOS attack, we can see an increase of both HTTP activity (due to the attacking flows) but also of TELNET activity (which is not in the flows labelled as attacks). We believe this TELNET activity was induced by the research group during the UGR'16 dataset generation. Clearly, from the perspective of anomaly detection, identifying this TELNET activity as part of the attack is indeed correct.

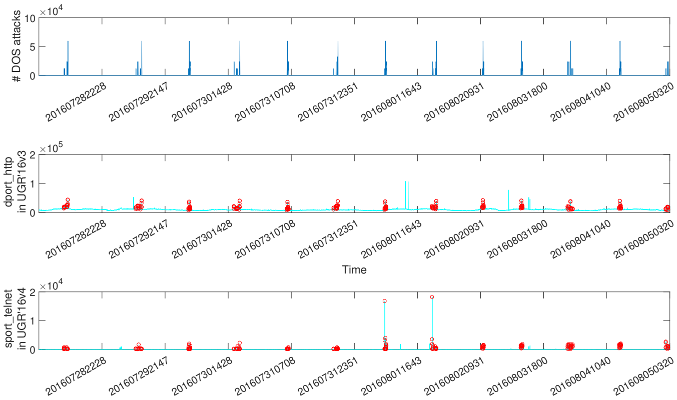
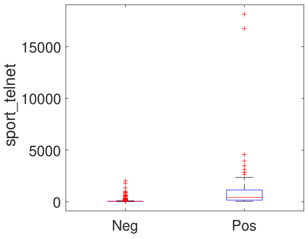


Fig. 8. Time series of the DOS Attacks (top), of feature dport_http in UGR'16v3 (middle) and of feature sport_telnet in UGR'16v4 (bottom).

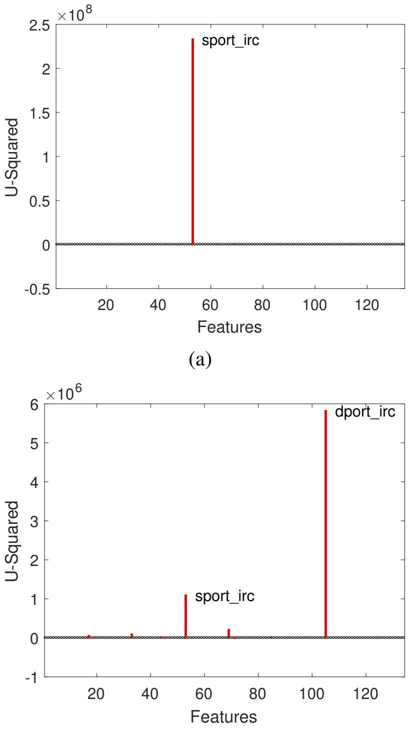
When we use unidirection flows (UGR'16v4), the DOS attacks are only characterized by the activity in the TELNET source port (Fig. 6(b)). This activity represents the flows that go from the TELNET server to the client. Fig. 9 shows this characterization is statistically significant but also of high quality: the activity of TELNET source port in normal observations is almost null. This is the explanation for the higher performance of anomaly detection models when using unidirectional flows in DOS attacks. When instead we employ bidirectional flows, both client-server and server-client flows are combined in a way that the detection ability is reduced, since the resulting pattern in background traffic is not so negligible (Fig 7).



Ttest $\mathbf{p\_value} < \mathbf{0.01}$

Fig. 9. Boxplot of sport_telnet in background traffic (Negative) versus DOS traffic (Positive) in UGR'16v4.

We repeated the U-Squared analysis for the observations including NERISBOTNET attacks (Fig. 10). For this attack, unlike in the DOS attacks, the bidirectional flows provide a better detection performance. Using as a reference UGR'16v3, the U-Squared points to 'sport_irc' as the main feature for the attack[4]. If otherwise UGR'16v4 is used, we get both 'sport_irc' and 'dport_irc' as relevant. While all aforementioned features, regardless the reference, yield statistical significant results (not shown), according to the AUC values in Fig. 5, using bidirectional flows is more effective in this case.



(a)

(b)

Fig. 10. Comparison of U-Squared statistics for the NERISBOTNET attack using as a reference UGR'16v3 (a) and UGR'16v4 (b).

Given that the convenience on the use of unidirectional or birectional flows is attack specific, we can always combine both set of features in a single dataset with double number (268) of features. We name such dataset UGR'16v3v4. When we do so, the performance is optimized in general terms, as shown in Fig. 5.

C. *Anonymization*

UGR'16v4 represents the anonymized version of UGR'16v2. Performance results for UGR'16v4 are similar to those in UGR'16v2 (compare Figs. 1 and 5), with a relatively minor improvement in the botnet detection by the former.

D. *Assessing the Test Labelling*

We can use the same general interpretation approach for those background observations that obtain a high anomaly

[4] Inspecting the raw bidirectional flows with nfdump, the attacks are communications in which the server part is IRC and the client port uses a lower number than the server port. For this reason, when parsing bidirectional flows, nfdump mistakes IRC as the client (source) port. When parsing unidirectional flows, we see a separated amount of communications in both directions.

score when using a reference dataset. As an example, we show in Fig. 11 the anomaly scores for the MSNM model trained from UGR'16v4, highlighting with circles the location of the labelled attacks. We also highlight with dots in the plot those background observations that obtain an anomaly score above 100. We will focus on an interval with 13 consecutive of this type of observations, starting at '201608040948'.

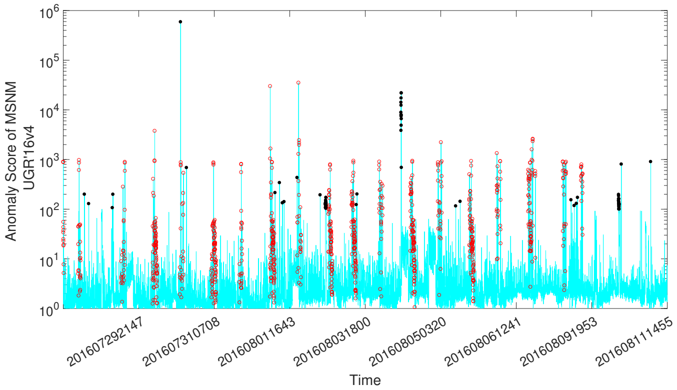


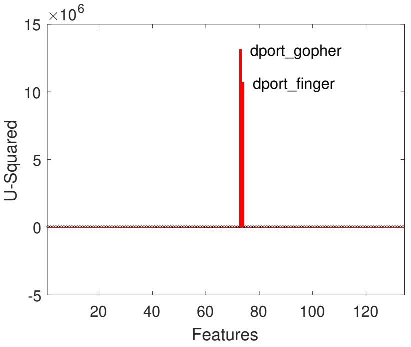
Fig. 11. Time series of the Attacks (top) and of the anomaly score by MSNM in UGR'16v4 (bottom).

Inspecting this period with the U-Squared statistic (Fig. 12) and UGR'16v4 as a reference, we found that the pattern of anomaly was associated to the destination port of the gopher and finger protocols. Comparing the rest of background traffic with this period in those specific features, we found a clear and statistically significant excess on the use of the protocols in the period (Fig. 13). Inspecting the raw flows of the anomaly with nfdump, we found 1 device performing subtle scanning for open ports in the network. Clearly, this corresponds to a malicious activity and, as such, the labelling was incorrect in the period under investigation. We found similar results in other analyzed periods. Note that the accuracy of the labelling has a profound impact on our interpretation of the results when using ROC/AUC values. To some extent, this is a similar problem to the one treated in section IV.A with the anomaly in June, which was mislabelled as 'normal' background traffic. In this case, however, mislabelling in the test dataset affects the reliability of the ROC/AUC.

Finally, it should be noted that the scanning activity found was not limited to the protocols or the time period detected by MSNM. This shows a limitation of the flow-based detection, in that malicious activity can be hidden in the background traffic and can only be detected thanks to multivariate patterns (e.g., the pattern of NERISBOTNET in unidirectional flows) and/or when the pattern in background traffic is almost null.

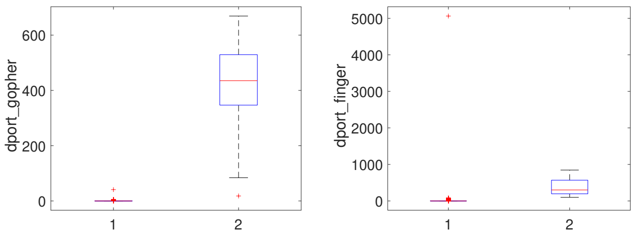
V. Conclusions

In this paper, we present a number of experiments that assess the quality of anomaly detection in a real network dataset, the UGR'16, which can be regarded as a benchmark in the network literature. The experiments are intended to understand the impact in anomaly detection performance of customary data preprocessing steps and of different anomaly detection models. The motivation of these experiments is that a wide part of the literature on this topic is focused on exploring and optimizing modelling variants, while data preprocessing and data quality assessment is regarded as a minor topic, that does not deserve so much research attention. The case study under analysis, however, show that data preprocessing has a major influence on the performance result. Given that this case study represents a benchmark for research and a realistic situation, our conclusion is that the community should look more into (automatic) data quality assessment and improvement techniques. This conclusion, in the authors view, is of special relevance in the context of autonomous networks, where the data workflow, including steps like data gathering, preprocessing and modelling, is expected to have little or none human supervision.



(a)

Fig. 12. Comparison of U-Squared statistics for the anomalous period detected in UGR'16v4.



(a) Ttest $\mathbf{p\_value < 0.01}$ (b) Ttest $\mathbf{p\_value < 0.01}$

Fig. 13. Boxplot of dport_gopher (a) and dport_finger (b) in background traffic (Negative) versus the detected period (Positive) in UGR'16v4.

As part of our analysis, we contribute an approach to investigate the reasons behind disparate performance results when using dataset variants. In this approach we employ the Univariate-Squared statistic, to identify the pattern of a given anomaly, and the statistical/visualization assessment of this pattern with t-tests, boxplots and time series visualizations. Analysis like the one performed in this case study can be useful to determine the dataset of optimal quality for anomaly detection among a set of variants considered, and to understand the reason behind this optimality.

References

[1] P. Kalmbach, J. Zerwas, P. Babarczi, A. Blenk, W. Kellerer, and S. Schmid, “Empowering self-driving networks,” in *Proceedings of the afternoon workshop on self-driving networks*, 2018, pp. 8–14.

[2] L. Caviglione, M. Choraś, I. Corona, A. Janicki, W. Mazurczyk, M. Pawlicki, and K. Wasielewska, “Tight arms race: Overview of current malware threats and trends in their detection,” *IEEE Access*, vol. 9, pp. 5371–5396, 2021.

[3] I. Sarker, A. S. M. Kayes, S. Badsha, H. Alqahtani, P. Watters, and A. Ng, “Cybersecurity data science: an overview from machine learning perspective,” *Journal of Big Data*, vol. 7, 07 2020.

[4] J. Camacho and K. Wasielewska, “Dataset quality assessment in autonomous networks with permutation testing,” in *NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium*. IEEE, 2022, pp. 1–4.

[5] R. Y. Wang and D. M. Strong, “Beyond accuracy: What data quality means to data consumers,” vol. 12, no. 4, p. 5–33, 1996.

[6] F. Li, S. Nastic, and S. Dustdar, “Data quality observation in pervasive environments,” in *2012 IEEE 15th International Conference on Computational Science and Engineering*, 2012, pp. 602–609.

[7] B. Heinrich, D. Hristova, M. Klier, A. Schiller, and M. Szubartowicz, “Requirements for data quality metrics,” *Journal of Data and Information Quality*, vol. 9, no. 2, 2018.

[8] F. Sidi, P. H. Shariat Panahy, L. S. Affendey, M. A. Jabar, H. Ibrahim, and A. Mustapha, “Data quality: A survey of data quality dimensions,” in *2012 International Conference on Information Retrieval Knowledge Management*, 2012, pp. 300–304.

[9] R. M. Verma, V. Zeng, and H. Faridi, “Data quality for security challenges: Case studies of phishing, malware and intrusion detection datasets,” in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS ’19. New York, NY, USA: Association for Computing Machinery, 2019, p. 2605–2607. [Online]. Available: https://doi.org/10.1145/3319535.3363267

[10] A. Kenyon, L. Deka, and D. Elizondo, “Are public intrusion datasets fit for purpose characterising the state of the art in intrusion event datasets,” *Computers & Security*, vol. 99, p. 102022, Dec. 2020. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0167404820302959

[11] H. S. Obaid, S. A. Dheyab, and S. S. Sabry, “The impact of data pre-processing techniques and dimensionality reduction on the accuracy of machine learning,” in *2019 9th Annual Information Technology, Electromechanical Engineering and Microelectronics Conference (IEMECON)*. IEEE, 2019, pp. 279–283.

[12] C. V. Gonzalez Zelaya, “Towards Explaining the Effects of Data Preprocessing on Machine Learning,” in *2019 IEEE 35th International Conference on Data Engineering (ICDE)*. Macao, Macao: IEEE, Apr. 2019, pp. 2086–2090. [Online]. Available: https://ieeexplore.ieee.org/document/8731532/

[13] E. J. M. Lauría and G. K. Tayi, “A COMPARATIVE STUDY OF DATA MINING ALGORITHMS FOR NETWORK INTRUSION DETECTION IN THE PRESENCE OF POOR QUALITY DATA,” p. 12.

[14] H. Chen, J. Chen, and J. Ding, “Data Evaluation and Enhancement for Quality Improvement of Machine Learning,” *IEEE Transactions on Reliability*, vol. 70, no. 2, pp. 831–847, Jun. 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9417095/

[15] J. L. Guerra, C. Catania, and E. Veas, “Datasets are not enough: Challenges in labeling network traffic,” *Computers & Security*, vol. 120, p. 102810, Sep. 2022. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0167404822002048

[16] M. Landauer, M. Frank, F. Skopik, W. Hotwagner, M. Wurzenberger, and A. Rauber, “A framework for automatic labeling of log datasets from model-driven testbeds for hids evaluation,” in *Proceedings of the 2022 ACM Workshop on Secure and Trustworthy Cyber-Physical Systems*, 2022, pp. 77–86.

[17] A. Barredo Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. Garcia, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, and F. Herrera, “Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI,” *Information Fusion*, vol. 58, pp. 82–115, Jun. 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1566253519308103

[18] G. Maciá-Fernández, J. Camacho, R. Magán-Carrión, P. García-Teodoro, and R. Therón, “UGR‘16: A new dataset for the evaluation of cyclostationarity-based network IDSs,” *Computers & Security*, vol. 73, pp. 411–424, 2018.

[19] A. Lakhina, M. Crovella, and C. Diot, “Mining anomalies using traffic feature distributions,” *ACM SIGCOMM Computer Communication Review*, vol. 35, no. 4, p. 217, 2005.

[20] J. Camacho, G. Maciá-Fernández, J. Díaz-Verdejo, and P. García-Teodoro, “Tackling the big data 4 vs for anomaly detection,” *Proceedings of the IEEE INFOCOM*, no. 1, pp. 500–505, 2014.

[21] J. Camacho, J. M. García-Giménez, N. M. Fuentes-García, and G. Maciá-Fernández, “Multivariate big data analysis for intrusion detection: 5 steps from the haystack to the needle,” *Computers Security*, vol. 87, p. 101603, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167404818307909

[22] N. M. Fuentes García *et al.*, “Multivariate statistical network monitoring for network security based on principal component analysis,” 2021.

[23] J. Camacho, A. Pérez-Villegas, P. García-Teodoro, and G. Maciá-Fernández, “PCA-based multivariate statistical network monitoring for anomaly detection,” *Computers & Security*, vol. 59, pp. 118–137, June 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167404816300116

[24] B. Schölkopf, A. J. Smola, R. C. Williamson, and P. L. Bartlett, “New Support Vector Algorithms,” *Neural computation*, vol. 12, no. 5, pp. 1207–1245, 2000.

[25] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, “Estimating the Support of a High-Dimensional Distribution,” *Neural Computation*, vol. 13, no. 7, pp. 1443–1471, 2001. [Online]. Available: http://www.mitpressjournals.org/doi/abs/10.1162/089976601750264965

[26] T. Alpcan and T. Başar, *Network security: A decision and game-theoretic approach*. Cambridge University Press, 2010.

[27] M. Collins and M. S. Collins, *Network security through data analysis: building situational awareness*. ” O’Reilly Media, Inc.”, 2014.

[28] M. Fuentes-García, G. Maciá-Fernández, and J. Camacho, “Evaluation of diagnosis methods in pca-based multivariate statistical process control,” *Chemometrics and Intelligent Laboratory Systems*, vol. 172, pp. 194 – 210, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0169743917302046

[29] J. Camacho, “Observation-based missing data methods for exploratory data analysis to unveil the connection between observations and variables in latent subspace models,” *Journal of Chemometrics*, vol. 25, no. 11, pp. 592–600, 2011.