data

MDPI

Article

Data Quality Tools to Enhance a Network Anomaly Detection Benchmark †

José Camacho * and Rafael A. Rodríguez-Gómez

Research Centre for Information and Communication Technologies (CITIC-UGR), University of Granada, 18014 Granada, Spain; rodgom@ugr.es

* Correspondence: josecamacho@ugr.es

† This paper is an extended version of our work "Quality In/Quality Out: Data quality more relevant than model choice in anomaly detection with the UGR'16", published in IEEE/IFIP Network Operations and Management Symposium (NOMS 2023).

Abstract: Network traffic datasets are essential for the construction of traffic models, often using machine learning (ML) techniques. Among other applications, these models can be employed to solve complex optimization problems or to identify anomalous behaviors, i.e., behaviors that deviate from the established model. However, the performance of the ML model depends, among other factors, on the quality of the data used to train it. Benchmark datasets, with a profound impact on research findings, are often assumed to be of good quality by default. In this paper, we derive four variants of a benchmark dataset in network anomaly detection (UGR'16, a flow-based real-world traffic dataset designed for anomaly detection), and show that the choice among variants has a larger impact on model performance than the ML technique used to build the model. To analyze this phenomenon, we propose a methodology to investigate the causes of these differences and to assess the quality of the data labeling. Our results underline the importance of paying more attention to data quality assessment in network anomaly detection.

Dataset: https://codas.ugr.es/animalicos/en/results under the entry 'UGR16 Feature data'.

Keywords: Netflow; UGR'16; anomaly detection; data quality

Academic Editors: Panagiotis Karras and Han Woo Park

Received: 21 October 2024
Revised: 21 February 2025
Accepted: 21 February 2025
Published: 25 February 2025

Citation: Camacho, J.; Rodríguez-Gómez, R.A. Data Quality Tools to Enhance a Network Anomaly Detection Benchmark. *Data* **2025**, *10*, 33. https://doi.org/10.3390/data10030033

1. Introduction

Machine learning (ML) has emerged as a cornerstone in the evolution of computer networks, enabling adaptive decisions based on complex, large-scale data. In recent years, the use of ML in networks has expanded beyond basic data analysis to drive innovations in network management, optimization, and security [1]. Traditional network management strategies are struggling to keep up with the exponential growth of connected devices and network complexity. ML-based approaches offer a possible solution by enabling automated, data-driven decision-making.

Network management and optimization represent a key application of ML in networking [2]. ML algorithms are used to monitor and predict network traffic patterns, identify potential problems, and optimize routing and resource allocation. For instance, when integrated with Software-Defined Networking (SDN), ML can facilitate centralized control, enabling seamless adjustments across the network. ML is also valuable in network

security [3], where it helps to detect anomalies by analyzing patterns that may indicate malicious behavior.

However, the effectiveness of ML tools is inherently tied to the quality of the data they are trained on, underscoring the necessity of high-quality datasets [4,5]. While significant attention has been devoted to model optimization and the development of novel ML methods, the assessment of data quality remains underexplored and often overlooked [6,7].

In this paper, we show that the impact of minor data modifications prior to modeling with ML in network anomaly detection can be more relevant than the specific ML method used. These modifications include subtle adjustments in the computation of traffic features, data anonymization practices, and the selection of observations used for model fitting and testing. The data from this case study illustrate that the research community needs to look further into data quality assessment and improvement.

Our main contributions are as follows:

- We derive four variants of a benchmark dataset in network anomaly detection, the UGR'16 dataset [8] (Dataset available online at https://nesg.ugr.es/nesg-ugr16/, acessed on the 20 February 2025), 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 then sort the data corresponding to the four variants of UGR'16 available for the community. We believe this example can be useful for future research on data quality assessment and automatic labeling.
- We apply 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 differences are relevant and/or when they are due to labeling inaccuracies, particularly with respect to unlabeled anomalies.

Ultimately, the primary contribution of this work is the dataset itself. The analysis underscores the importance of selecting a high-quality dataset for machine learning tasks, demonstrating how even minor changes in the data can significantly impact detection results.

The paper is organized as follows. Section 2 introduces the case study under analysis, the preprocessing and data selection steps, and ML methods considered. Section 3 presents the experimental results and Section 4 draws the conclusions.

2. Materials and Methods

In the following subsections, we present the original case study under analysis, the dataset variants introduced in this paper, and the comparison approach and the strategy to explain the results.

2.1. The Original UGR'16 Dataset

The UGR'16 dataset 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 us to model and study the normal behavior of the network and to unveil certain anomalies such as SPAM campaigns. The dataset flows were labeled as either "background", representing legitimate network traffic, or "anomalies", denoting non-legitimate traffic.

Furthermore, an additional data collection took place between July and August 2016, during which controlled attacks were introduced to generate a test dataset for validating anomaly detection algorithms. In this phase, twenty-five virtual machines were set up within one of the ISP's sub-networks: five machines launched attacks on the other twenty. The attacks included *Denial of Service* (DOS), and two types of *port scanning*: SCAN11 (one attacker, one victim) and SCAN44 (four attackers, four victims). Botnet traffic (*NERISBOT-NET*) was also introduced in the capture. These attacks were carried out over a twelve-day period, at various times, following both pre-planned and random schedules, and with real background traffic.

The primary advantage of the UGR'16 dataset is its origin in a real network environment, which allows for the validation of algorithms under realistic conditions. Background traffic in the dataset reflects real-world day/night cycles and weekday/weekend usage patterns, enhancing the authenticity of the data for cybersecurity research. To date, the UGR'16 dataset has been referenced in more than 260 research papers (according to Google Scholar), establishing it as a benchmark in anomaly detection research using real network traffic. The dataset's general features are outlined in Table 1.

Table 1. Characteristics of the calibration and the test sets.

Feature	Calibration	Test
Capture start	10:47 h 03/18/2016	13:38 h 07/27/2016
Capture end	18:27 h 06/26/2016	09:27 h 08/29/2016
Attacks start	N/A	00:00 h 07/28/2016
Attacks end	N/A	12:00 h 08/09/2016
Number of files	17	6
Size (compressed)	181 GB	55 GB
# Connections	≈13,000 M	≈3900 M

2.2. Building New Versions of UGR'16

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 complex 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 are 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. [9] 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, and thus, quantitative variables were obtained directly from Netflow records. Camacho et al. [10] 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, significantly reducing the initial data size and simplifying the data analysis.

Using the FaaC approach, the dataset is divided into 1-minute intervals, and a total of 134 features are obtained per interval. The feature extraction process consists of two main steps: (i) converting binary files into flow-level CSV files using the nfdump tool, and (ii) transforming these CSV files into feature vectors with the FCParser tool [11].

In our case, using parallelization with 16 CPUs, the daytime features were extracted in approximately 3h, and the complete dataset was transformed within approximately 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 [11].

The data for the new versions of UGR'16 can be accessed in the following different data formats (DFs):

DF.1 Raw registers in nfcapd format (a binary format used in the nfdump package): We downloaded the data from the original repository, accessed via the Network Engineering and Security Group (NESG) website[1], and fixed the original nfcapd files, since they use a deprecated format which can no longer be interpreted with nfdump. The IPs were anonymized, as discussed in the original paper [8]. The data in the new nfcapd (nfdump readable) version are stored in weekly files and organized in folders by month. The total amount of storage required for the data is 341 GB. The raw registers are accessible through the Data Analysis as a Service (DAaaS) deployed at https://codas.ugr.es/animalicos/es/daaas (accessed on the 20 February 2025).

DF.2 Feature data per time interval using the feature-as-as-counter approach, with one file in csv format per time interval of 1 min: The data are obtained in two steps. First, nfcapd files are transformed into csv format using the nfdump tool. These intermediate files are not stored due to the volume of the resulting data. Subsequently, the FCParser is applied over intermediate files, generating output files (one per minute) with a single row of 144 counters each. There are two versions of the UGR16 dataset in this format, which consider bidirectional and unidirectional flows in nfdump, respectively. The data volume is 1,5GB for the two versions, which reflects the compression capability of the parsing operation. The data generation of these files can also be reproduced at the DAaaS.

DF.3 Feature data for complete datasets, with as many rows as time intervals and a selection of 134 counters as columns: The feature data are provided with labels for eight attack classes: the artificial attacks (DOS, SCAN11, SCAN44 and NERISBOTNET) and some real activity that has already been observed and labeled in the original data [8] (BLACKLIST, UDPSCAN, SSHSCAN and SPAM). Data can be found in csv, excel, and matlab format. The data are available for download at https://codas.ugr.es/animalicos/en/results under the entry 'UGR16 Feature data' (accessed on the 20 February 2025). This is the easiest format to use for anyone interested in the development of new ML models from the data. We provide four variants of the feature data, which are described in Table 2:

- **UGR'16v1.** In this version, the original (non-anonymized) Netflow logs from the entire NOC period (March to June) were used to generate the feature data, but the resulting data were completely anonymized. This dataset corresponds to the same data that were utilized in previous studies [11] and in most works that make use of UGR'16.
- **UGR'16v2.** Fuentes [12] discovered that the training data from June in the previous version contain real anomalies, which hinder the detection of botnet attacks in the test set. To address this, we created a second version by simply discarding the observations from June in UGR'16v1.
- **UGR'16v3.** In the first two versions, only unidirectional Netflow flows were considered, which made interpreting the results more challenging. Therefore, we generated a third version using the bidirectional flows from DF.2. Like UGR'16v2, this version also excludes June from the training data.

– **UGR'16v4.** Finally, to differentiate the effect of anonymization from the use of bidirectional or unidirectional flows, we developed a fourth version, which is equivalent to UGR'16v3 but uses unidirectional flows from DF.2 instead.

All previous versions used the same previously described approach to feature engineering (FaaC).

Table 2. 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

2.3. Comparison

By considering the four different versions of UGR'16 described previously, we can assess the impact of various data preprocessing steps on the quality of anomaly detection models. Specifically, our strategy allows us to evaluate the following factors:

- The impact of training data selection, by comparing the performance between UGR'16v1 (which includes June) and UGR'16v2 (which excludes June).
- The effect of using bidirectional versus unidirectional flows, by comparing performance between UGR'16v3 (bidirectional flows) and UGR'16v4 (unidirectional flows).
- The effect of data anonymization, by comparing the results between UGR'16v2 (generated from non-anonymized flows) and UGR'16v4 (from anonymized flows).

To assess the impact of data preprocessing on anomaly detection performance, as compared to the influence of different ML methods, we employ two distinct approaches: Multivariate Statistical Network Monitoring (MSNM) [13] and a radial basis function (RBF) kernel-based one-class support vector machine (OCSVM) [14,15], which is a commonly used kernel choice. MSNM represents a linear approach well suited to handling the highly multivariate nature of FaaC features, while OCSVM, as a nonlinear method, has the advantage of capturing complex, nonlinear behavior in normal traffic patterns. Thus, both methods have very different features that could, in principle, affect performance in a significant way.

To evaluate anomaly detection performance across the various data and model variants, we calculate the false-positive rate (FPR) and true-positive rate (TPR) using the labeled test dataset and plot receiver operating characteristic (ROC) curves to show the TPR as a function of the FPR at different anomaly detection thresholds. This method is particularly suitable for network security contexts, where balancing true positives and false positives is essential [16,17]. We compare the ROC curves using the area under the curve (AUC) metric, which quantifies the performance of the anomaly detector; ideally, the AUC should approach 1, while an AUC around 0.5 indicates random performance.

2.4. Strategy for Explanation of the Results

We will use the Univariate-Squared (U-Squared) statistic [18] to interpret the differences in anomaly detection performance when using different dataset versions for model training. U-Squared has superior diagnostic ability compared to other multivariate (multifeature) diagnosis tools and it has two main advantages: it is extremely simple and it is model-agnostic[2].

The U-Squared statistic, like other diagnosis solutions [19], provides a discriminative pattern for the attack in comparison to a reference. In our case, this reference is represented by any of the versions of the UGR'16. To diagnose a certain anomaly type, represented by a set of observations $\mathbf{x}_n$ for $n \in \{1, ..., N\}$ that contain such anomalies, we compute the vectors of sample means μ and standard deviations σ of the reference dataset, where $\mathbf{x}_n$, $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ are row vectors of length the number of features. Then, for each anomalous observation $\mathbf{x}_n$, the U-Squared is calculated as follows:

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

where the symbol $\oslash$ represents the Hadamard (element-by-element) division, and $|||$ represents the absolute value. The accumulated U-Squared for the set of anomalous observations is written as follows:

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

where vector $\mathbf{d}^2$ also represents the length and 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 ways in which the considered attack differs from the reference. Positive (negative) bars mean that the attack shows significantly higher (or lower) values than the reference for specific features.

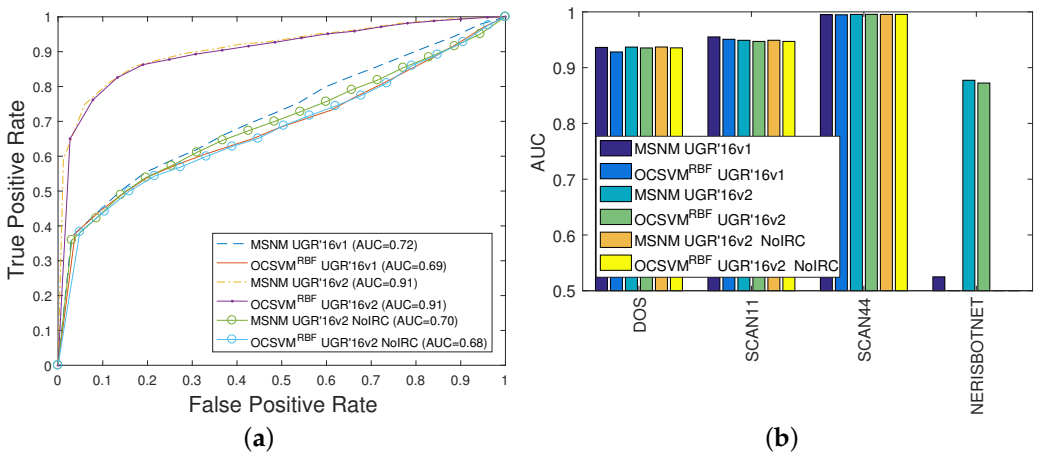
The U-Squared statistic helps us to identify the feature subset that is most suitable for a specific model and attack type. This is then analyzed statistically to evaluate its effectiveness in detecting the anomaly. We will demonstrate that this approach can provide a simple but complete interpretation of the performance differences between dataset variants in our case study.

3. Experiments and Results

3.1. Influence of the Set of Observations

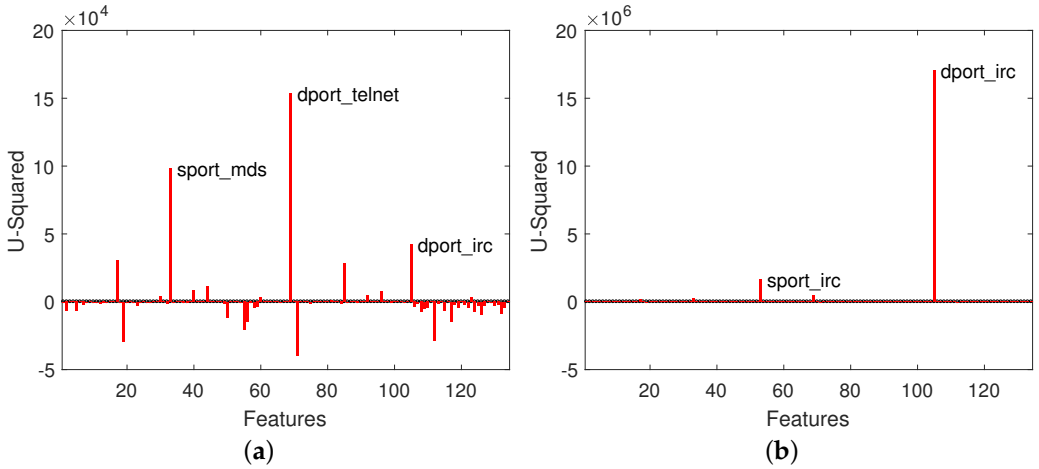
Figure 1 shows a comparison of the two anomaly detectors (MSNM and OCSVM) when trained with the datasets UGR'16v1 and UGR'16v2, and with a sub-version of UGR'16v2 (UGR'16v2 NoIRC) that will be discussed later. Figure 1a presents the general ROC curves, obtained for the four types of artificial attacks, and Figure 1b represents the AUCs per attack type. Performance differences between the two anomaly detectors are minor in all cases. However, a significant difference emerges when June is included in the training data (UGR'16v1) versus when it is excluded (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, which is related to suspicious activity through an MIRC channel [12].

To check our hypothesis, we compute the U-Squared statistics for the observations in the test set that contains flows of the NERISBOTNET attack, and use UGR'16v1 and UGR'16v2 as references, respectively. This is shown in Figure 2. When using UGR'16v1 as a reference (Figure 2a), 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 (Figure 2b), the NERISBOTNET attack is mainly characterized by the amount of flows to or from the IRC port[3]. The difference in U-Squared patterns between the two reference datasets suggests that ML models trained on them will employ different methods to detect the NERISBOTNET attack. These differences affect performance, as seen in the AUC results.



(**a**) (**b**)

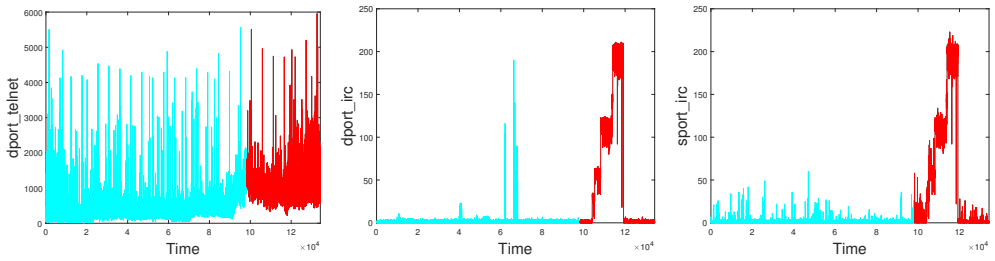
Figure 1. ROC curve (**a**) and attack-type-based AUC results (**b**) for the data parsed from original unidirectional flows in UGR'16v1 and UGR'16v2, and for a variant of the latter with no IRC features (UGR'16v2 NoIRC).



(**a**) (**b**)

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

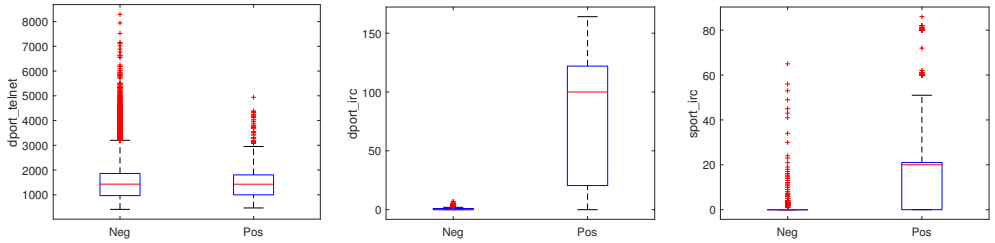
To further investigate the reason behind the performance differences when using UGR'16v1 and UGR'16v2 as a reference, in Figure 3, we present the time series of the training data from March to June for a set of selected features, previously highlighted by the U-Squared. All features present a change in tendency in June, which is especially clear in the case of IRC features. The latter show the suspicious activity in the MIRC channel found in [12]. When June is included in the reference (UGR'16v1), it informs the anomaly detection models that this type of behavior 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 for characterizing the NERISBOTNET attack (Figure 2a).



(**a**) (**b**) (**c**)

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

Figure 4 provides a set of boxplots comparing the distributions of normal observations and NERISBOTNET observations in the test set, focusing on the same selected features of Figure 3. Additionally, we include the outcomes of a t-test to evaluate whether there is statistical evidence to support the increased presence of NERISBOTNET activity in the corresponding feature. Notably, the feature *dport_telnet*, which was emphasized in the original reference of UGR'16v1, does not exhibit statistically significant differences between normal and NERISBOTNET observations. This lack of distinction is attributed to the inclusion of the June anomaly as part of the "normal data", which leads detectors to incorporate such activity into the normality model. Consequently, this inclusion limits the detectors' ability to identify it as anomalous in future traffic. As a result, this feature (and, by extension, UGR'16v1) leads to lower detection capabilities for the attack. However, all IRC features show statistically 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 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 Figure 1 under the label "UGR'16v2 NoIRC". Additionally, a further inspection of raw flows using nfdump revealed extensive usage of IRC port 6667 in the NERISBOTNET traffic, corroborating our previous observations.



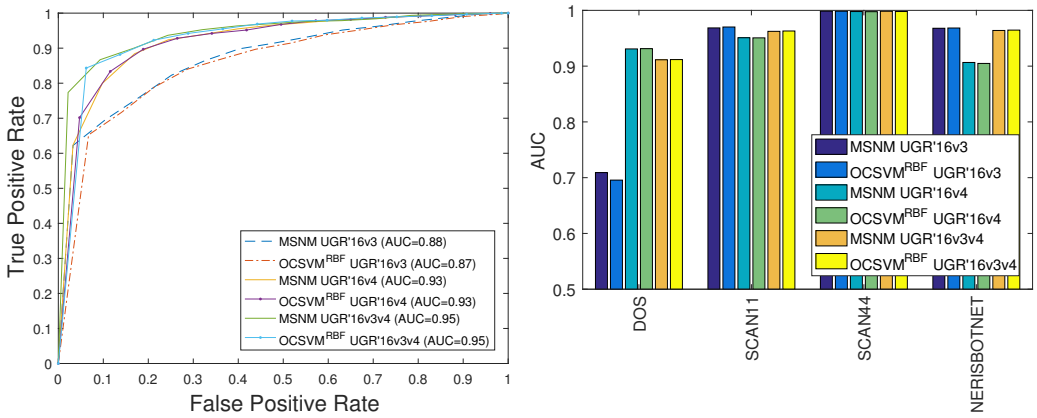
(**a**) Ttest $p\_value = 0.47$ (**b**) Ttest $\mathbf{p\_value < 0.01}$ (**c**) Ttest $\mathbf{p\_value < 0.01}$

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

This example underscores the importance of conducting a thorough assessment of data quality in anomaly detection tasks, especially for unsupervised identification of unusual patterns, a topic that has received limited attention but is crucial for effective performance in ML applied to computer networks. In this real world example, careful selection of relevant observations and features had a far more significant impact on detection results than the choice of ML method.

3.2. Bidirectional vs. Unidirectional Flows

Figure 5 displays the performance results of anomaly detectors on datasets UGR'16v3 and UGR'16v4, as well as on a combination of both datasets that will be discussed later. Across all tests, the performance difference between the two detectors, MSNM and OCSVM, is negligible. However, there is a noticeable performance difference depending on whether bidirectional or unidirectional flows are used, with unidirectional flows generally performing better. This performance disparity is particularly evident when it comes to detecting DOS attacks. Therefore, as in the previous analysis, even minor decisions in data preparation, such as whether to apply an nfdump flag when parsing flows, can have a more substantial impact on detection performance than the choice of the ML tool itself. Figure 5b also indicates that bidirectional flows perform slightly better when it comes to detecting NERISBOTNET attacks, implying that the most effective training dataset may vary depending on the specific attack being targeted.



(**a**) (**b**)

Figure 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).

To shed some light into the observed differences in the detection of DOS attacks, we computed the U-Squared for the observations with DOS attacks using UGR'16v3 and UGR'16v4 as references (Figure 6). Again, we found different patterns of characterization depending on the reference dataset. When bidirectional flows were used, the DOS attacks were characterized by flows with destination ports HTTP and TELNET. Statistically significant differences between normal observations and those containing DOS attacks confirmed this characterization (Figure 7). However, upon examining the raw flows labeled as DOS attacks using nfdump, we found that these flows only involve destination port HTTP. The correlation between DOS attacks and TELNET activity is confirmed in Figure 8. As shown in the Figure, every time there is a DOS attack, we can see an increase in both HTTP activity (due to the attacking flows themselves) and TELNET activity (which is not part of the flows that are labeled as attacks). We believe this TELNET activity may have been erroneously introduced by the research group during the UGR'16 dataset generation.

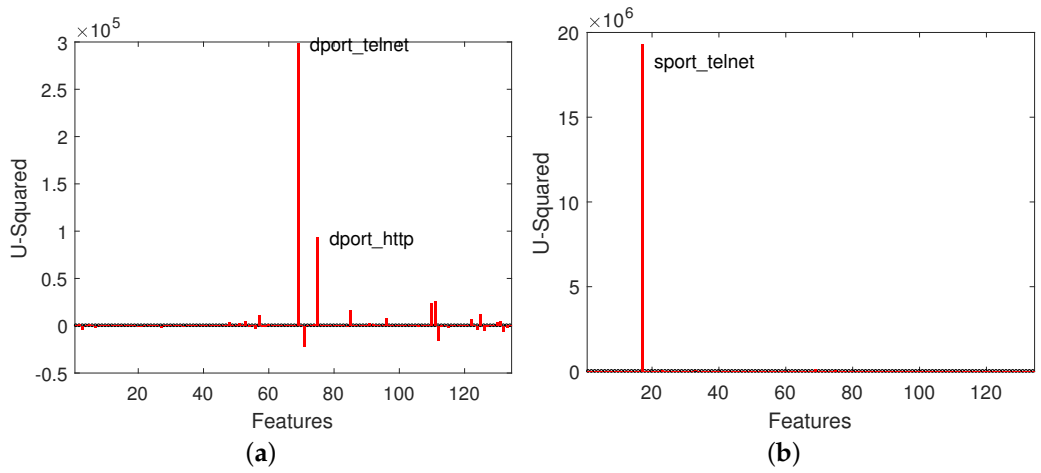


Figure 6. Comparison of U-Squared statistics for the DOS attack using UGR'16v3 (**a**) and UGR'16v4 (**b**) as the references.

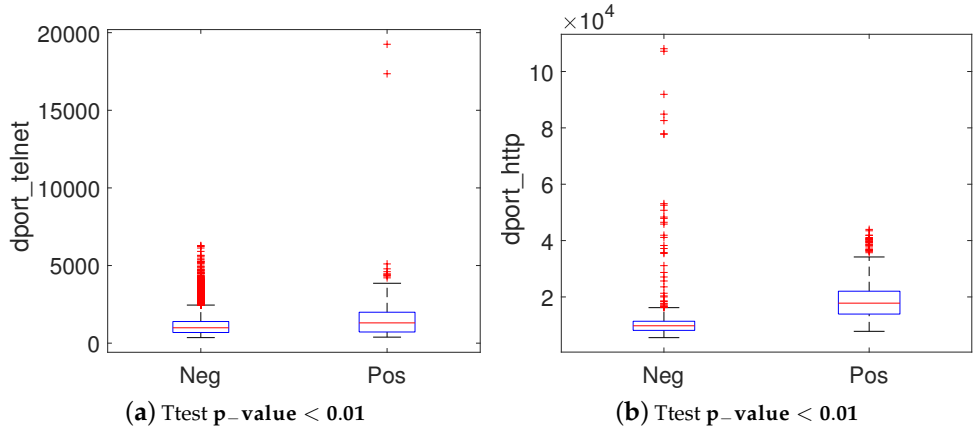


Figure 7. Boxplots of selected features in background traffic (negative) versus DOS traffic (positive) in UGR'16v3.

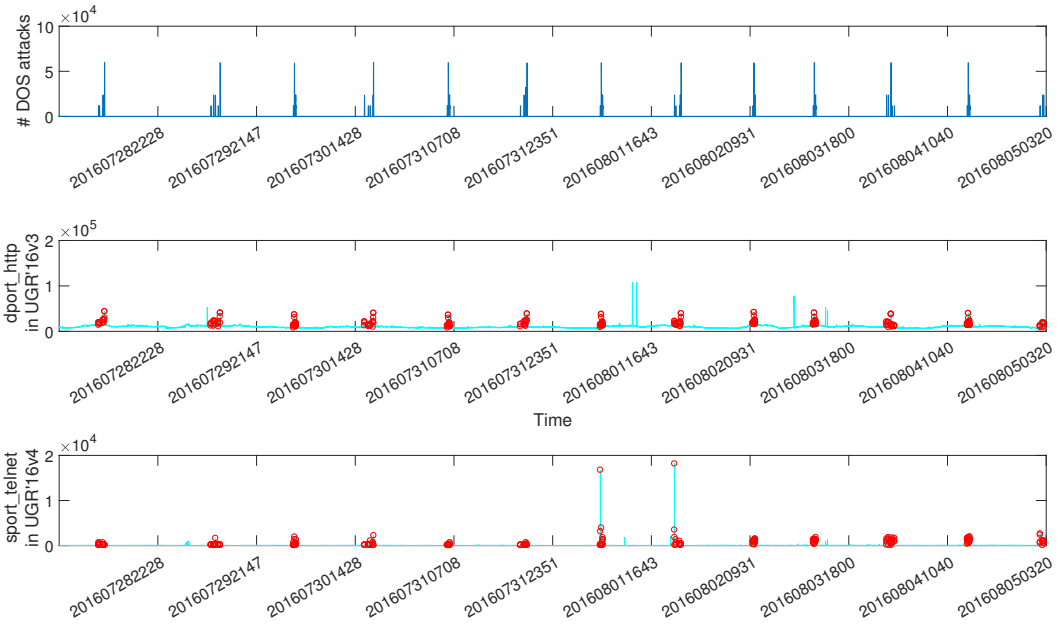


Figure 8. Time series of the DOS Attacks (**top**), of feature dport_http in UGR'16v3 (**middle**) and of feature sport_telnet in UGR'16v4 (**bottom**).

When we use unidirectional flows (UGR'16v4), the DOS attacks are only characterized by the activity in the TELNET source port (Figure 6b). This activity represents the flows from the TELNET server to the client. Figure 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 result explains the improved performance of anomaly detection models when using unidirectional flows for DOS attacks. When we employ bidirectional flows instead, both client–server and server–client flows are combined in such a way that the detection ability is reduced, since the resulting pattern in background traffic is less negligible (Figure 7).

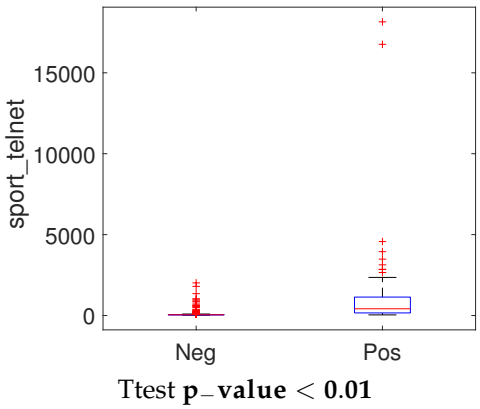


Figure 9. Boxplot of sport_telnet in background traffic (negative) versus DOS traffic (positive) in UGR'16v4.

We repeated the U-Squared analysis for the observations including NERISBOTNET attacks (Figure 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 of the attack[4]. If otherwise UGR'16v4 is used, both 'sport_irc' and 'dport_irc' are deemed relevant. While all aforementioned features, regardless of the reference, yield statistically significant results, according to the AUC values in Figure 5, using bidirectional flows is more effective in this case.

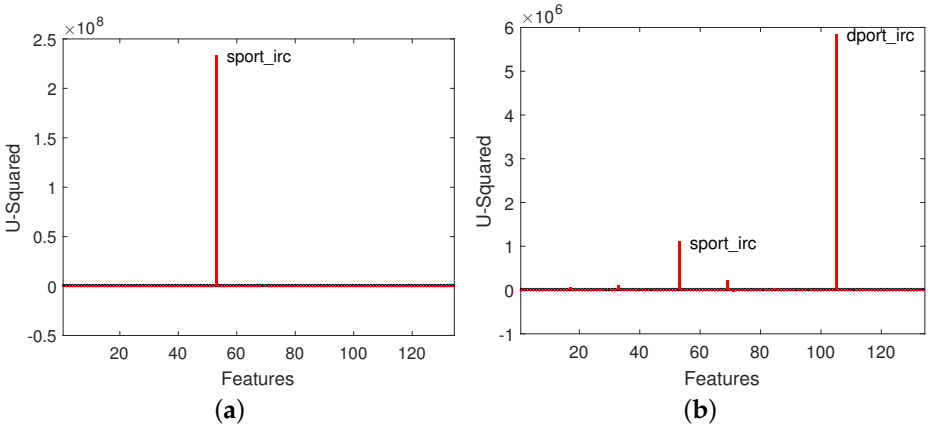


Figure 10. Comparison of U-Squared statistics for the NERISBOTNET attack using as a reference UGR'16v3 (**a**) and UGR'16v4 (**b**).

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

3.3. Anonymization

UGR'16v4 represents the anonymized version of UGR'16v2. The performance results for UGR'16v4 are slightly better than those achieved for UGR'16v2 (compare Figures 1 and 5). However, it should be noted that in the original versions of UGR'16 (UGR'16v1 and UGR'16v2), the real anomalies that were detected (e.g., SPAM) [8] were discarded at the flow level prior to the parsing step, while in new versions (UGR'16v3 and UGR'16v4), corresponding 1-minute observations were omitted after the parsing step. If we remove the corresponding observations from UGR'16v2, the AUC results are actually better than for UGR'16v4, once again demonstrating that understanding the impact of data preprocessing on the final quality is instrumental for a sound interpretation of the results.

3.4. Assessing the Test Labeling for False Negatives

We can use the same general interpretation approach for background observations that obtain a high anomaly score when using a reference dataset. As an example, in Figure 11, we present the anomaly scores for the MSNM model trained using UGR'16v4, with circles highlighting the location of the labeled attacks. We use dots to highlight the background observations that obtain an anomaly score above 100, which are false-negative candidates (that is, they are labeled as normal observations when in reality they may be affected by some form of attack). We will focus on an interval with 13 consecutive instances of this type of observation, starting at '201608040948'.

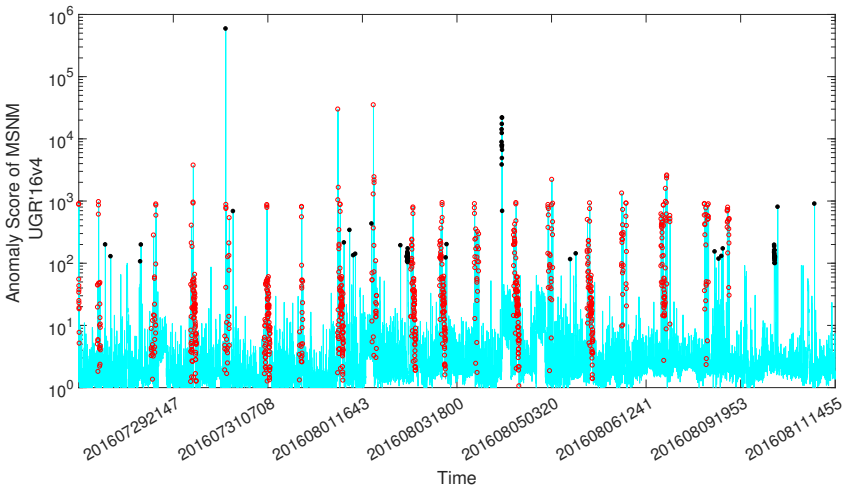


Figure 11. Time series of the attacks (**top**) and of the anomaly score by MSNM in UGR'16v4 (**bottom**).

Inspecting this period with the U-Squared statistic (Figure 12) and UGR'16v4 as a reference, we found that the pattern of anomaly was associated with 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 with regard to the use of the protocols in the period (Figure 13). Inspecting the raw flows of the anomaly with nfdump, we found one device that was subtly scanning for open ports in the network. Clearly, this corresponds to malicious activity and, as such, the labeling is incorrect and the observations are indeed false negatives. We found similar results in other analyzed periods. It is important to note that the accuracy of labeling significantly impacts the interpretation of results when using ROC/AUC values. To some extent, this is a similar problem to the one associated with the anomaly in June, which was mislabeled as 'normal' background traffic. In this case, however, mislabeling in the test dataset affects the reliability of the ROC/AUC.

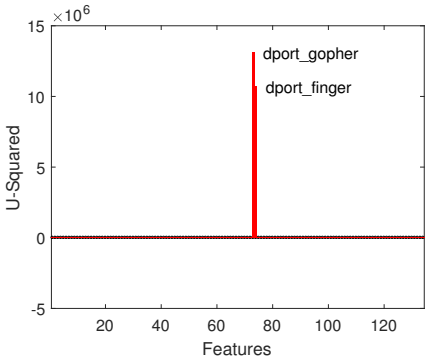


Figure 12. Comparison of U-Squared statistics for the anomalous period detected in UGR'16v4.

Notice that mislabeling is a general problem when dealing with real data, particularly in massive network datasets. Expecting a real dataset to be perfectly labeled can be generally regarded as a naive assumption. We need tools to identify this situation, particularly when it is dramatic, and a good understanding of its consequences (loss of reliability of ROC results) is required.

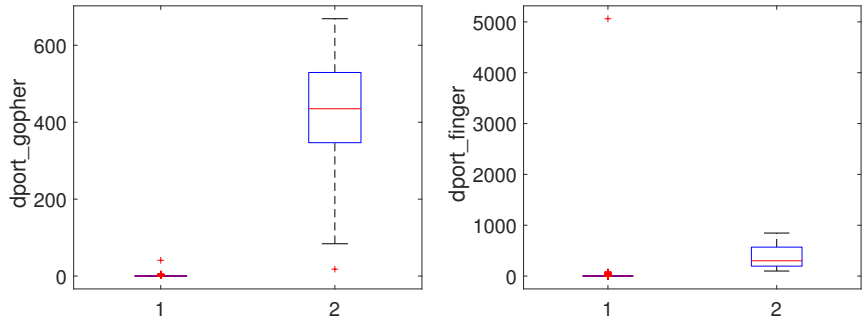


Figure 13. Boxplot of dport_gopher (**a**) and dport_finger (**b**) in background traffic (negative) versus the detected period (positive) in UGR'16v4.

4. Conclusions

In this paper, we present four derived variants of the UGR'16 dataset, a real-world network dataset widely recognized as a benchmark in the field of network anomaly detection. Our work primarily focuses on these dataset variants, aiming to evaluate the impact of customary data preprocessing steps on anomaly detection performance, as well as the role of different anomaly detection models. The motivation of these experiments is that a vast amount of the literature on this topic is focused on exploring and improving modeling variants, while data preprocessing and data quality assessment are regarded minor topics, which do not deserve as much research attention. However, the data from this case study demonstrates that data preprocessing can significantly influence performance outcomes, sometimes even more so than the choice of detection model. As this case study serves as both a benchmark for research and a realistic scenario, we conclude that the community should focus more on (automatic) data quality assessment.

Furthermore, we introduce an approach to investigate the reasons behind disparate performance results when using different dataset variants in a network anomaly detection context. 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. Analyses like the one performed in this case study can be useful for identifying the dataset with the highest quality for anomaly

detection among a set of variants considered and to understand the reasons behind its superior performance.

Author Contributions: Conceptualization, J.C. and R.A.R.-G.; methodology, J.C. and R.A.R.-G.; software, J.C.; investigation, J.C.; resources, J.C.; data curation, J.C.; writing—original draft preparation, J.C. and R.A.R.-G.; visualization, J.C.; project administration, J.C.; funding acquisition, J.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Agencia Estatal de Investigación in Spain, MCIN/AEI/10.13039/501100011033, grant No. PID2020-113462RB-I00.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data is available as described in Section 2.2.

Conflicts of Interest: The authors declare no conflicts of interest.

Notes

[1] https://nesg.ugr.es/nesg-ugr16/, acessed on the 22 February 2025.

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

[3] Recall both UGR'16v1 and UGR'16v2 use unidirection 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.

[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.

References

1. Kalmbach, P.; Zerwas, J.; Babarczi, P.; Blenk, A.; Kellerer, W.; Schmid, S. Empowering self-driving networks. In Proceedings of the Afternoon Workshop on Self-Driving Networks, Budapest, Hungary, 24 August 2018; pp. 8–14.
2. Hussain, F.; Hassan, S.A.; Hussain, R.; Hossain, E. Machine Learning for Resource Management in Cellular and IoT Networks: Potentials, Current Solutions, and Open Challenges. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 1251–1275. [CrossRef]
3. Chou, D.; Jiang, M. A Survey on Data-driven Network Intrusion Detection. *ACM Comput. Surv.* **2021**, *54*, 1–36. [CrossRef]
4. Caviglione, L.; Choraś, M.; Corona, I.; Janicki, A.; Mazurczyk, W.; Pawlicki, M.; Wasielewska, K. Tight Arms Race: Overview of Current Malware Threats and Trends in Their Detection. *IEEE Access* **2021**, *9*, 5371–5396. [CrossRef]
5. Sarker, I.; Kayes, A.S.M.; Badsha, S.; Alqahtani, H.; Watters, P.; Ng, A. Cybersecurity data science: An overview from machine learning perspective. *J. Big Data* **2020**, *7*, 41. [CrossRef]
6. Camacho Páez, J.; Wasielewska, K.; Espinosa, P.; Fuentes García, M. Quality In/Quality Out: Data quality more relevant than model choice in anomaly detection with the UGR'16. In Proceedings of the NOMS 2023-2023 IEEE/IFIP Network Operations and Management Symposium, Miami, FL, USA, 8–12 May 2023.
7. Camacho, J.; Wasielewska, K. Dataset Quality Assessment in Autonomous Networks with Permutation Testing. In Proceedings of the NOMS 2022–2022 IEEE/IFIP Network Operations and Management Symposium, Budapest, Hungary, 25–29 April 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 1–4.
8. Maciá-Fernández, G.; Camacho, J.; Magán-Carrión, R.; García-Teodoro, P.; Therón, R. UGR'16: A new dataset for the evaluation of cyclostationarity-based network IDSs. *Comput. Secur.* **2018**, *73*, 411–424. [CrossRef]
9. Lakhina, A.; Crovella, M.; Diot, C. Mining anomalies using traffic feature distributions. *ACM SIGCOMM Comput. Commun. Rev.* **2005**, *35*, 217. [CrossRef]
10. Camacho, J.; Maciá-Fernández, G.; Díaz-Verdejo, J.; García-Teodoro, P. Tackling the big data 4 vs for anomaly detection. In Proceedings of the 2014 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Toronto, ON, Canada, 27 April–2 May 2014; pp. 500–505. [CrossRef]
11. Camacho, J.; García-Giménez, J.M.; Fuentes-García, N.M.; Maciá-Fernández, G. Multivariate Big Data Analysis for intrusion detection: 5 steps from the haystack to the needle. *Comput. Secur.* **2019**, *87*, 101603. [CrossRef]
12. Fuentes García, N.M. Multivariate Statistical Network Monitoring for Network Security Based on Principal Component Analysis. Ph.D. Thesis, Universidad de Granada, Granada, Spain, 2020.

13. Camacho, J.; Pérez-Villegas, A.; García-Teodoro, P.; Maciá-Fernández, G. PCA-based Multivariate Statistical Network Monitoring for anomaly detection. *Comput. Secur.* **2016**, *59*, 118–137. [CrossRef]
14. Schölkopf, B.; Smola, A.J.; Williamson, R.C.; Bartlett, P.L. New Support Vector Algorithms. *Neural Comput.* **2000**, *12*, 1207–1245. [CrossRef] [PubMed]
15. Schölkopf, B.; Platt, J.C.; Shawe-Taylor, J.; Smola, A.J.; Williamson, R.C. Estimating the Support of a High-Dimensional Distribution. *Neural Comput.* **2001**, *13*, 1443–1471. [CrossRef] [PubMed]
16. Alpcan, T.; Başar, T. *Network Security: A Decision and Game-Theoretic Approach*; Cambridge University Press: Cambridge, UK, 2010.
17. Collins, M.; Collins, M.S. *Network Security Through Data Analysis: Building Situational Awareness*; O'Reilly Media, Inc.: Sebastopol, CA, USA, 2014.
18. Fuentes-García, M.; Maciá-Fernández, G.; Camacho, J. Evaluation of diagnosis methods in PCA-based Multivariate Statistical Process Control. *Chemom. Intell. Lab. Syst.* **2018**, *172*, 194–210. [CrossRef]
19. Camacho, J. Observation-based missing data methods for exploratory data analysis to unveil the connection between observations and variables in latent subspace models. *J. Chemom.* **2011**, *25*, 592–600. [CrossRef]