# Evaluation of the Limit of Detection in Network Dataset Quality Assessment with PerQoDA

Katarzyna Wasielewska$^{1[0000-0001-8087-790X]}$,
Dominik Soukup$^{2[0000-0002-4737-8735]}$, Tomáš Čejka$^{3[0000-0001-7794-9511]}$, and
José Camacho$^{1[0000-0001-9804-8122]}$

[1] Signal Theory, Networking and Communications Department, University of Granada, Granada, Spain {k.wasielewska,josecamacho}@ugr.es
[2] Faculty of Information Technology, Czech Technical University in Prague, Prague, Czech Republic soukudom@fit.cvut.cz
[3] CESNET a.l.e, Prague, Czech Republic cejkat@cesnet.cz

**Abstract.** Machine learning is recognised as a relevant approach to detect attacks and other anomalies in network traffic. However, there are still no suitable network datasets that would enable effective detection. On the other hand, the preparation of a network dataset is not easy due to privacy reasons but also due to the lack of tools for assessing their quality. In a previous paper, we proposed a new method for data quality assessment based on permutation testing. This paper presents a parallel study on the limits of detection of such an approach. We focus on the problem of network flow classification and use well-known machine learning techniques. The experiments were performed using publicly available network datasets.

**Keywords:** Dataset quality assessment · Permutation testing · Network dataset · Network security · Attack detection · Machine learning · Classification

## 1 Introduction

Network security is a key research area today. Development in the field of the Internet goes hand in hand with increasing threats. Machine learning (ML) systems play a critical role in this field. However, ML techniques suffer of the ”garbage-in-garbage-out” (GIGO) problem, meaning they can only be as good as the data they are trained on [26]. This is a fact that has serious consequences because, despite its high performance, the model may be ineffective when trained on a dataset that does not represent the real environment. This situation can easily happen if the ML model is moved to another network or there is a drift in input data.

Although the network traffic can be easily captured from the network, many papers noticed the lack of the high-quality network datasets [27][9] and the problem of assessing the quality of datasets is overlooked. Scientists and practitioners tend to focus their efforts on optimizing ML models rather than on the quality of the datasets [7], and the research area related to assessing the quality of the dataset is overlooked. Dataset cleaning fixes well-known bugs in data but does not fix the problem of the quality. After removing duplicates, outliers, and errors resulting from technical problems or human activity, there are still problems with the completeness of the dataset, its accuracy, consistency, or uniqueness of the data, and these characteristics still remain difficult to assess [7]. In addition, problems such as class imbalance [25], class overlapping [12], noisy data [15], incorrect labels [10], or non-stationarity [33] are often unnoticed.

In this paper, we focus on the dataset quality assessment problem. In previous work, we presented how to use permutation testing for this task [8]. Our approach allows us to check whether the dataset contains enough information to predict a specific labeling, i.e., assignation of traffic units to the legitime or attack class. We showed that the proposed methodology is able to effectively assess the quality of a network dataset by checking the relationship between the observations and labels. In this article, we highlight the problem of the sensitivity of this method to partial mislabeling, that is, the incorrect assignation of a subset of traffic units to the normal/attack classes, and demonstrate how to use our approach to capture even a small inconsistency.

The advantage of permutation testing [23] [22] is that the permutation tests create a null distribution that allows us to test the statistical significance of the performance results of a given ML classifier or set of classifiers. Moreover, as already shown in [21], permutation testing can be a useful tool to evaluate the impact of noisy data on the model performance.

In this paper, our contributions are as follows:

- We describe challenges for dataset quality assessment which are crucial for the effectiveness of machine learning-based network systems;
- We emphasize that a small problem in the network data may affect ML results;
- We experimentally investigate the limit of detection in our dataset quality assessment method based on permutation testing (PerQoDA) presented in [8][32] and propose the change in our original methodology;
- We show how by permuting (even extremely) a small number of labels, we can detect small mislabeling problems in the dataset. These are important research results because, to the best of our knowledge, there are no methods that can detect mislabeling at such a high level of sensitivity.

The rest of the paper is organized as follows. Section 2 discusses related work in the literature. Section 3 provides an introduction to permutation testing, the details of the permutation-based methodology for assessing the quality of the dataset, and how to interpret the results. Section 4 describes the problem of the limit of detection in our method. Section 5 lists the results of the experiments carried out on real network datasets. Finally, Section 6 concludes the paper and discusses future work.

## 2 Related work

Dataset quality evaluation is key in the analysis and modeling of big data [4], and it is of interest when developing new benchmarking datasets, critical for network security. Evaluating the dataset quality is challenging and must be done prior to any data modeling. While there are metrics that evaluate some important properties of a dataset (accuracy, completeness, consistency, timeliness, and others), these metrics often overlap [16]. Also, these metrics are more focused on the quality of data, and there is a lack of complete and proven methodologies for assessing the quality of datasets from a general perspective [30]. Soukup et al. [28] proposed general dataset quality definitions and an evaluation methodology of dataset quality based on selected performance measures between several versions of the dataset. Statistical methods were used to compare their results. However, no methods for overall evaluation were proposed.

Current research is mainly focused on data cleaning and optimization that can indirectly improve the dataset's quality. Taleb et al. [30] proposed quality evaluation scheme that analyzes the entire dataset and seeks to improve it. More metrics and proposals are part of future work. Another method is crowdsourcing, where experts perform small tasks to address difficult problems. There are many applications of this approach, for example, a query-oriented system for data cleaning with oracle crowds [5] or a technique that improves the quality of labels by repeatedly labeling each sample and creating integrated labels [34]. Another technique is metamorphic testing, originally developed to evaluate software quality and verify relations among a group of test outputs with corresponding test inputs [36]. Auer et al. [3] proposed to use this method to assess the quality of data expressing data as functions and defining metamorphic relations on these functions. Ding et al. [11] showed another application of metamorphic tests that were used to assess the fidelity, diversity, and validity of datasets on a large scale. The authors of [31] proposed a black-box technique that uses metamorphic tests to find mislabeled observations for which there is a probability that the metamorphic transformations will cause an error. Erroneous labels are found using entropy analysis, which leverages information about the output uncertainty. Additional options for dataset optimization are transfer learning [24] and knowledge graphs [6] that were used to detect information gaps and semantic problems. There is also an approach using reinforcement learning [35]. This meta learning framework explores how likely it is that each training sample will be used in training the predictive model.

Apruzzese et al. [2] proposed semisupervised methods within the framework of active learning. This is very beneficial to improve the current dataset but it cannot be used to evaluate quality. Moreover, Joyce et al. [17] is focused on the unlabeled part of the dataset. The proposed solution can detect problematic traits of dataset that can lead to over-fitting. However, the quality measure is missing, and domain knowledge is required. Engelen et al. [14] is focused on dataset quality assessment, however, the dataset is analyzed manually based on deep domain knowledge.

In the paper [8], we proposed a permutation-based assessment methodology that allows the analyst to conveniently check whether the information contained in the dataset is rich enough to classify observations precisely. Our method can detect inconsistencies in the relationships between observations and labels in multidimensional datasets. We also proposed a scalar metric allowing us to compare two versions of a dataset (e.g., after some differential preprocessing) in terms of quality [32]. We focus on supervised binary classification problems and estimate dataset quality without input data or hyperparameters optimization.

In this article, we explore the detection limits of our approach and show how to detect small imperfections in large datasets.

## 3 Background

In this section, we describe the permutation testing method, introduce an approach that uses permutation tests to assess the quality of a dataset, and explain how to interpret the results.

### 3.1 Permutation testing

Permutation tests are a form of statistical inference which does not rely on assumptions about the data distribution [23]. Thanks to this approach, we can test if there is a significant relationship between the content of a traffic dataset and its corresponding labeling. For that, we define the so-called null hypothesis that the association of the traffic and the labeling is mild enough so that it could be the result of randomness, and we test whether this hypothesis could be rejected.

Permutation testing relies on random sampling. We repeatedly shuffle (i.e., permute) the selected data and check if the unpermutted (real) data comes from the same population as the resamples. To compute the p-value, we typically take the number of test statistics computed after permutations that are greater than the initial test statistic and divide it by the number of permutations. If the p-value is less than or equal to the selected significance level, we can reject the null hypothesis and accept the alternative hypothesis, which reflects that the relationship between traffic and labels is statistically significant.

### 3.2 Dataset quality assessment based on the permutation testing

In short, our method presented in [8] is to calculate the model performance after each permutation and see how many times that performance was better than the model performance on the original data (true results). If this happens many times, it would mean that our dataset is so random that it does not allow classifiers to learn an accurate classification model. Since we want to assess the quality of the dataset and not the quality of a specific ML classification strategy, our approach is based on a pool of classifiers (from the simplest to complex and from traditional to the state-of-the-art). In our method, we only permute labels and examine the relationship between observations and labels. For each classifier, after $P$ permutations, we obtain $P$ performance results. Then we compare each result with the true performance result and compute a p-value. The obtained p-value table allows us to evaluate the quality of the dataset.

Let $M$ be the model performance[4] calculated from the original dataset and $M^*$ the model performance computed after permutation. The p-value can be defined as follows [1]:

$$\text{p-value} = \frac{\text{No. of } (M^* \geq M) + 1}{\text{Total no. of } M^* + 1} \tag{1}$$

In our method, we set the significance level to 0.01, and we define the null hypothesis as that the association between observations and labels in the dataset is the simple result of chance. This means that if the p-value $> 0.01$, the null hypothesis cannot be rejected. Therefore the dataset has a weak relationship between observations and labels. On the other hand, if the p-value $\leq 0.01$, we can reject the null hypothesis and conclude that the relationship is significantly strong.

We assess the statistical significance of the performance results permuting a selected part of the dataset, not just the whole dataset. We run permutation tests for different label percentages (for example, 50%, 25%, 10%, 5%, 1%). By taking an increasing number of labels into the permutation, we are able to identify different levels of quality in the data, that is, of association between data and labels. Note these set of tests are incremental and as such we did not apply corrections on the significance level (e.g., Bonferroni corrections).

Let $(\mathbf{X}, \mathbf{y})$ be a dataset, where $\mathbf{X}$ is the set of observations and $\mathbf{y}$ is the set of labels. To evaluate the quality of the dataset, we perform the following steps:

1. Train a pool of classifiers using the original dataset $(\mathbf{X}, \mathbf{y})$
2. Evaluate each model using the selected metric
3. Permute selected percentage of the labels $\mathbf{y}$ to get new labels $\mathbf{y_p}$ and new dataset $(\mathbf{X}, \mathbf{y_p})$
4. Train the pool of classifiers on the dataset $(\mathbf{X}, \mathbf{y_p})$
5. Evaluate each model with the selected performance metric
6. For $y_p$ and $y$, compute the correlation coefficient
7. Repeat the steps 3 through 6, $P$ times
8. Calculate p-value according to Eq. (1)
9. Repeat the steps 7 and 8 for each value of percentage

The proposed approach works for both balanced and imbalanced datasets because it is finding trends in permutations [32].

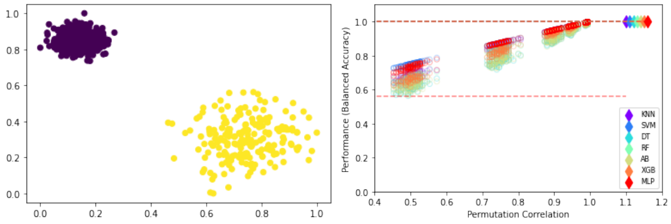
### 3.3 Visualisation and interpretation

After the procedure described in Section 3.2, we get a pool of performance results after permutations and a p-value table. To assess the performance results, we

[4] We can choose any performance metric such as accuracy, precision, recall, etc.

combine a permutation chart and a p-value table. If at least one classifier shows statistical significance in all permutation percentages, we can deem the dataset as *good*. The reason is that we can find at least one ML method that can identify the relationship between data and labels. If no ML method shows significant results at any permutation level, the data should be considered of *bad* quality. Any result in between these two extreme outputs reflect a partial level of quality, which grows with the number of permutation percentages in which we find at least one significant classification model.

An example of the visualisation of the performance results is shown in Fig. 1b. Consider the dataset presented in Fig. 1a. This dataset is of good quality because the classes are well separated, so we expect the ML algorithms to perform very well on this data. In the permutation chart, we can see the true performance results (shown by diamonds) and all the performance results after permutations (shown by circles). Each performance result after permutation is located depending on the correlation between the original labeling and the permuted one [18]. We can notice that the true performance is high (equal to or close to 1) as expected, and the results after each permutation are lower than the true results (what is also expected if the dataset is of good quality). The lowest performance at different percentage levels is marked with a red dashed horizontal line. This can be interpreted as a *baseline of randomness*. This value can sometimes be unexpectedly high and should therefore be observed (in this case, it is around 0.55).



(a) Dataset

(b) Permutation chart

Fig. 1: Dataset (a) and the permutation chart (b)

As can be seen in the p-value table (Tab. 1), all performance results are statistically significant. All classifiers at all permutation levels reject the null hypothesis (the symbol . represents the value $\leq 0.01$). This means a very strong relationship exists between the observations and the labels in the original dataset.

Additionally, in the previous work [32] we proposed a scalar metric for comparing datasets. We defined a *slope* metric that corresponds to the slope of the regression line fitted to the points representing the classifier's performance scores (obtained after permutations) at different permutation levels (see Fig. 1b). Thus, we got one slope per classifier, the largest of which was defined as a measure of the quality of the dataset (in this case, the slope is approximately 0.75).

Table 1: p-value table: p-values less than or equal to the significance level 0.01 are replaced by a dot.

| | 50% | 25% | 10% | 5% | 1% |
|---|---|---|---|---|---|
| KNN | . | . | . | . | . |
| SVM | . | . | . | . | . |
| DT | . | . | . | . | . |
| RF | . | . | . | . | . |
| AB | . | . | . | . | . |
| XGB | . | . | . | . | . |
| MLP | . | . | . | . | . |

## 4 Limits of detection

In the dataset quality assessment method described in Section 3.2, we permute the selected percentage of the labels. However, taking 1% of the labels in a large dataset, we may not notice problems in the relationship between $\mathbf{X}$ and $\mathbf{y}$ in a minor number of instances. Since we want to detect as minor mislabelling problems as possible in the dataset, intuitively, we should permute the smallest possible number of labels. This will allow us to establish the limit of detection (LOD) of our approach. In chemistry, this term is defined as the lowest concentration of an analyte that can be reliably detected with statistical significance (for example, with 95% certainty) using a given analytical procedure [19]. Based on these considerations, this paper examines the LOD of our approach, that is, how well we can detect minor problems in datasets.

In order to investigate the limit of detection in our dataset quality evaluation method, we will perform permutation tests on a very small number of observations (for example, 100, 50, 25, 10, 5, 1), regardless of the size of the dataset. By permuting such a small fraction of the labels, we can evaluate the performance loss (if any) at a high level of detection. This allows us to assess the relevance of very small parts of the dataset and consequently assess the accuracy of the labeling of the entire dataset, i.e., to evaluate its overall quality. It is also worth noting that we will have high correlation coefficient values for a large dataset because we only change a small part of the labels.

In practice, we make one change to the algorithm presented in Section 3.2. We will permute the same small number of labels in each dataset instead of a percentage (in step 3).

The theoretical foundations of the above considerations can be found in our work [32], in which we explained why our method of assessing the quality of a dataset is more sensitive at low permutation percentages.

## 5 Experiments

In this section, we present the results of the experiments with the LOD in the permutation-based dataset quality assessment method. We also present ML techniques and performance metrics that were used in the procedure. We present two case studies conducted on the publicly available real network datasets.

### 5.1 ML algorithms

In our experiments, we used a pool of well-know supervised ML methods: K Nearest Neighbours (KNN), kernel Support Vector Machine (SVM), Decision Tree (DT), Random Forest (RF), AdaBoost (AB), XGBoost (XGB), and multilayer perceptron (MLP). The DT, RF, AB and XGB classifiers had a weight class option set to "balanced". The other hyperparameters were the default. We used the standard stratified 2-fold cross-validation (CV) with shuffling the data before splitting into batches. In other words, datasets were split into two sets (for training and testing) keeping the percentage of samples for each class, the models were then trained on one split and evaluated in the other twice, and the performance results were averaged. Data has been scaled to range $[0, 1]$. We used the Weles tool [29] to automate the generation of results.

### 5.2 Evaluation metric

Our dataset quality assessment method can be used with different performance metrics [32]. For this paper, we selected a recall metric, which directly reflects the number of detected anomalies, i.e., the percentage of correctly classified positives, and which can arguably be considered especially relevant in cybersecurity research. The recall is defined as follows:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{2}$$

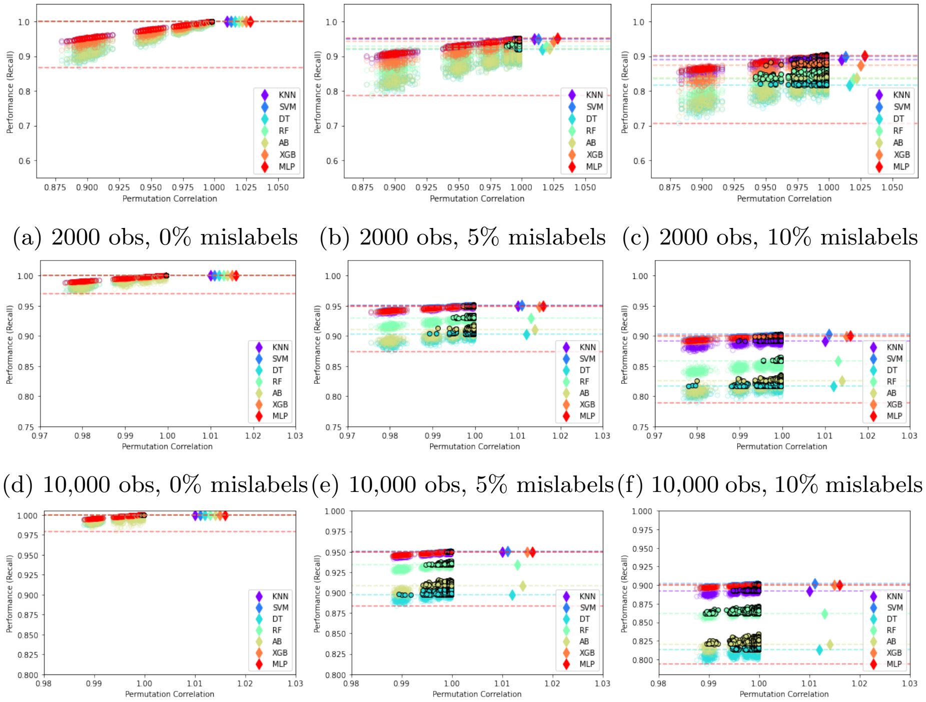
where TP is the number of correctly predicted undesired traffic (True Positives), and FN is the number of anomalous traffic classified as normal traffic (False Negatives).

### 5.3 Case studies

In this section, we present the results of the experiments on real datasets. We used publicly available datasets: inSDN and UGR16. In all experiments, we considered the following fixed number of permuted labels (instead of percentages): 100, 50, 25, 10, 5, and 1 (from each class), and we conducted 200 permutations. We focused on the binary classification problem.

#### Case study 1: inSDN dataset

The inSDN dataset is a publicly available network flow-based dataset that contains 68,424 normal (legitimate) and 275,465 attack observations captured in a Software Defined Network (SDN) environment [13]. The inSDN dataset includes



(a) 2000 obs, 0% mislabels (b) 2000 obs, 5% mislabels (c) 2000 obs, 10% mislabels

(d) 10,000 obs, 0% mislabels (e) 10,000 obs, 5% mislabels (f) 10,000 obs, 10% mislabels

(g) 20,000 obs, 0% mislabels (h) 20,000 obs, 5% mislabels (i) 20,000 obs, 10% mislabels

Fig. 2: Permutation charts for the inSDN datasets

attacks on the Open vSwitch (OVS) machine as well as the server-directed attacks: DoS, probe, brute force attacks (BFA), password-guessing (R2L), web application attacks, and botnet. The inSDN dataset contains 83 traffic features.

In this scenario, we assessed the quality of the dataset against the problem of distinguishing Probe attacks from normal traffic. We created balanced datasets (prevalence[5] = 0.5) with 2000, 10,000, and 20,000 observations. We removed the following features: Timestamp, Flow ID, Src IP, Dst IP, Src Port, Dst Port, and Protocol. To the original datasets, we introduced 0%, 5% and 10% mislabels. Mislabels were injected randomly to both the normal data and attack data (in the same proportions), and were present in the training set and test set. Our goal was to capture the quality difference between original and mislabeled datasets. Using the permutation strategy described in Section 4, we permute a maximum of 5% (100/2000), 1% (100/10,000), and 0.5% (100/20,000) of the labels of the first, second and third dataset, respectively.

The results of the dataset quality assessment with our permutation approach is shown in Fig. 2 and Tab. 2. As expected, we can see that mislabeled datasets are of lower quality than the original ones. All original samples are of a good

[5] percentage of positives in the dataset

Table 2: p-value tables for the inSDN datasets. P-values above significance level 0.01 are marked in red, lower p-values are replaced by dot.

| | 2000 obs<br>0% mislabels | | | | | | | 2000 obs<br>5% mislabels | | | | | | | 2000 obs<br>10% mislabels | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 100 | 50 | 25 | 10 | 5 | 2 | 1 | 100 | 50 | 25 | 10 | 5 | 2 | 1 | 100 | 50 | 25 | 10 | 5 | 2 | 1 |
| KNN | . | . | . | . | . | .02 | .02 | . | . | . | . | .01 | .08 | .16 | . | . | . | .05 | .13 | .18 | .14 |
| SVM | . | . | . | . | . | .02 | .02 | . | . | . | . | .02 | .13 | .15 | . | . | . | . | .07 | .25 | .31 |
| DT | . | . | . | . | . | . | . | . | . | .02 | .18 | .21 | .33 | .38 | . | .04 | .26 | .51 | .66 | .62 | .64 |
| RF | . | . | . | . | . | . | . | . | . | .12 | .66 | .81 | .91 | .89 | . | . | .10 | .32 | .51 | .52 | .56 |
| AB | . | . | . | . | . | . | . | . | . | .01 | .13 | .17 | .24 | .32 | . | .06 | .26 | .52 | .70 | .68 | .68 |
| XGB | . | . | . | . | . | . | . | . | . | .42 | .86 | .94 | .95 | .97 | . | . | .05 | .23 | .38 | .43 | .41 |
| MLP | . | . | . | . | . | .03 | .01 | . | . | . | . | .02 | .17 | .21 | . | . | . | .12 | .34 | .74 | .74 |
| | 10,000 obs<br>0% mislabels | | | | | | | 10,000 obs<br>5% mislabels | | | | | | | 10,000 obs<br>10% mislabels | | | | | | |
| | 100 | 50 | 25 | 10 | 5 | 2 | 1 | 100 | 50 | 25 | 10 | 5 | 2 | 1 | 100 | 50 | 25 | 10 | 5 | 2 | 1 |
| KNN | . | . | . | . | . | . | . | . | . | . | .06 | .15 | .28 | .30 | . | . | .06 | .17 | .21 | .22 | .30 |
| SVM | . | . | . | . | . | .02 | .05 | . | . | . | . | .02 | .11 | .10 | . | . | . | . | .03 | .15 | .19 |
| DT | . | . | . | . | . | . | . | . | . | .01 | .05 | .06 | .10 | .12 | .01 | .12 | .35 | .53 | .56 | .53 | .55 |
| RF | . | . | . | . | . | . | . | . | . | . | .08 | .13 | .10 | .15 | . | . | . | .05 | .07 | .05 | .04 |
| AB | . | . | . | . | . | . | . | . | . | .02 | .03 | .07 | .06 | .07 | . | .02 | .08 | .20 | .17 | .14 | .16 |
| XGB | . | . | . | . | . | . | . | . | . | . | .01 | .05 | .11 | .10 | . | . | .01 | .09 | .15 | .17 | .19 |
| MLP | . | . | . | . | .03 | .19 | .21 | . | . | . | . | .03 | .19 | .22 | . | . | . | .02 | .06 | .21 | .26 |
| | 20,000 obs<br>0% mislabels | | | | | | | 20,000 obs<br>5% mislabels | | | | | | | 20,000 obs<br>10% mislabels | | | | | | |
| | 100 | 50 | 25 | 10 | 5 | 2 | 1 | 100 | 50 | 25 | 10 | 5 | 2 | 1 | 100 | 50 | 25 | 10 | 5 | 2 | 1 |
| KNN | . | . | . | . | . | . | . | . | . | .10 | .47 | .60 | .66 | .69 | . | . | .07 | .17 | .20 | .23 | .25 |
| SVM | . | . | . | . | .05 | .22 | .24 | . | . | . | .02 | .09 | .27 | .24 | . | . | . | .02 | .16 | .48 | .48 |
| DT | . | . | . | . | .05 | .26 | .27 | . | .06 | .26 | .46 | .46 | .49 | .56 | .11 | .38 | .55 | .68 | .67 | .68 | .68 |
| RF | . | . | . | . | .18 | .46 | .50 | . | . | .05 | .15 | .15 | .16 | .18 | .04 | .21 | .42 | .55 | .60 | .61 | .61 |
| AB | . | . | . | . | .08 | .30 | .44 | . | .05 | .12 | .28 | .36 | .32 | .39 | .14 | .41 | .54 | .61 | .73 | .68 | .68 |
| XGB | . | . | . | .01 | .30 | .59 | .65 | . | . | . | .02 | .13 | .43 | .33 | . | . | .15 | .63 | .80 | .80 | .81 |
| MLP | . | . | . | .03 | .03 | .57 | .56 | . | . | . | . | .04 | .19 | .24 | . | . | . | .08 | .27 | .67 | .65 |

quality (even in case of the dataset with 20,000 observations and 0% mislabels we have at least one classifier with statistically significant results). However, in the permutation charts depicted in Fig. 2, we see black circles indicating that the performance results after permutations were better than in the original dataset. In the case of mislabeled datasets, we cannot reject the hypothesis that the dataset is random as all classifiers do not have significant results when we permute 1, 2, and 5 labels. Note that if the smallest percentage were 1%, we would not be able to detect relationship problems in datasets with 10,000 and 20,000 observations having 5% mislabels (because for these datasets 1% means 100 and 200 labels, respectively, and for these permutation levels the performance results are statistically significant). Moreover, these datasets would most likely have statistically significant performance scores for 5% permutation level as well, since typically, if the results are statistically significant at some level of permutation, they're also statistically significant if more labels are permuted.

The slope analysis also confirms that the mislabeled datasets are worse than the original dataset (Tab. 3). The original good-quality datasets have the highest slope values, and the datasets with 10% incorrect labels have the lowest slope.

It is worth noting, however, that for all analyzed samples, we can find ML techniques which achieved true performance results above 0.9 (Tab. 4), and, in research practice, without the dataset quality evaluation, they could be considered to be of good quality. In particular, the datasets with 5% mislabels have quite high performance results (even 0.95), and without the analysis with a method like the one we propose, they could be considered as good-quality datasets.

Table 3: The slopes computed for the inSDN consecutive normal observations and Probe/OVS attack data (samples without and with mislabels)

| Dataset | 0% mislabels | | 5% mislabels | | 10% mislabels | |
|---|---|---|---|---|---|---|
| 2000 obs | 0.97279 | DT | 0.78068 | DT | 0.60636 | RF |
| 10,000 obs | 1.04494 | AB | 0.82356 | DT | 0.70278 | AB |
| 20,000 obs | 1.04837 | DT | 0.85276 | DT | 0.63686 | DT |

Table 4: inSDN datasets - true performance results (recall)

| | 2000 obs<br>0% mislabels | 2000 obs<br>5% mislabels | 2000 obs<br>10% mislabels |
|---|---|---|---|
| KNN | 1.0 | 0.951 | 0.899 |
| SVM | 1.0 | 0.95 | 0.9 |
| DT | 1.0 | 0.907 | 0.814 |
| RF | 1.0 | 0.921 | 0.852 |
| AB | 1.0 | 0.908 | 0.816 |
| XGB | 1.0 | 0.926 | 0.881 |
| MLP | 1.0 | 0.95 | 0.9 |
| | 10,000 obs<br>0% mislabels | 10,000 obs<br>5% mislabels | 10,000 obs<br>10% mislabels |
| KNN | 1.0 | 0.948 | 0.891 |
| SVM | 1.0 | 0.951 | 0.902 |
| DT | 1.0 | 0.901 | 0.814 |
| RF | 1.0 | 0.93 | 0.854 |
| AB | 1.0 | 0.909 | 0.822 |
| XGB | 1.0 | 0.949 | 0.902 |
| MLP | 1.0 | 0.95 | 0.9 |
| | 20,000 obs<br>0% mislabels | 20,000 obs<br>5% mislabels | 20,000 obs<br>10% mislabels |
| KNN | 1.0 | 0.948 | 0.893 |
| SVM | 1.0 | 0.95 | 0.902 |
| DT | 1.0 | 0.9 | 0.809 |
| RF | 1.0 | 0.936 | 0.864 |
| AB | 1.0 | 0.91 | 0.82 |
| XGB | 1.0 | 0.95 | 0.9 |
| MLP | 1.0 | 0.95 | 0.901 |

### Case study 2: UGR16 dataset

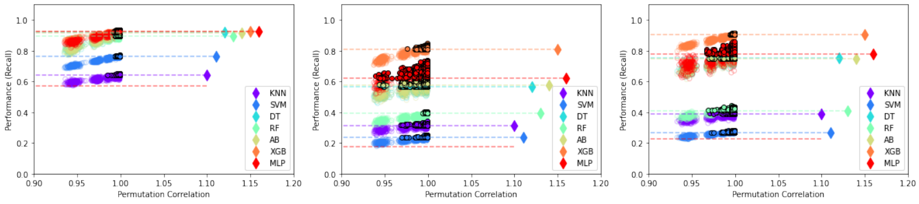
Another publicly available dataset we assessed was the UGR16 dataset [20]. This dataset contains Netflow flows taken from a real Tier 3 ISP network composed of virtualized and hosted services of many companies and clients. The network traffic was captured on the border routers, so this dataset contains all the incoming and outgoing traffic from the ISP. The UGR16 dataset contains 142 features and includes attack traffic (DoS, port scanning, and botnet) against fake victims generated by 25 virtual machines that were deployed within the network.

We tested three versions of this dataset depending on whether the flows in the dataset were unidirectional or bidirectional and whether the traffic was anonymized during parsing or not (Tab. 5). The original dataset (V1) contains unidirectional flows that were obtained from Netflow using the nfdump tool without using the -B option which creates bidirectional flows and maintain proper ordering. After the V1 dataset was anonymized, we created two additional datasets: V2 with the -B option enabled and V3 without this option. Additionally, V2 and V3 datasets were devoid of features identifying Internet Relay Chat (IRC) flows (Src IRC Port, Dst IRC Port) that were seen to have a deep impact in the detection of the botnet in the test data. After parsing, the datasets consisted of 12,960 observations containing flows aggregated at one-minute intervals (1006 observations with attack data). These datasets were highly imbalanced with prevalence = 0.078.

Table 5: UGR16 dataset versions

| Dataset | Direction | -B option | Anonymization | IRC |
|---|---|---|---|---|
| V1 | unidirectional | – | – | ✓ |
| V2 | bidirectional | ✓ | ✓ | – |
| V3 | unidirectional | – | ✓ | – |

The results of the UGR16 dataset quality assessment with our permutation-based approach are shown in Fig. 3, Tab. 6, and Tab. 7. As you can see, the original dataset (V1) is not perfect, and the high quality of the labeling is questionable. All ML techniques do not produce significant results for 1, 2, 5, and 10 permuted labels. Additionally, enabling option -B (V2) resulted in deterioration of the quality of the dataset (the RF algorithm is an exception, although it also has statistically insignificant results). It is also worth noting that anonymization lowered the quality of the dataset which is surprising and should be investigated in the future in more detail. True performance results are presented in Tab. 8.



(a) V1 (b) V2 (c) V3

Fig. 3: Permutation charts for the UGR16 datasets

Table 6: p-value tables for the UGR16 datasets. P-values above significance level 0.01 are marked in red, lower p-values are replaced by dot.

| | V1 | | | | | | | V2 | | | | | | | V3 | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 100 | 50 | 25 | 10 | 5 | 2 | 1 | 100 | 50 | 25 | 10 | 5 | 2 | 1 | 100 | 50 | 25 | 10 | 5 | 2 | 1 |
| KNN | . | . | .01 | .07 | .12 | .13 | .17 | . | .07 | .30 | .40 | .50 | .56 | .54 | . | .04 | .15 | .28 | .35 | .46 | .41 |
| SVM | . | . | . | .15 | .36 | .53 | .58 | . | .02 | .04 | .12 | .18 | .17 | .22 | . | .01 | .22 | .62 | .81 | .86 | .82 |
| DT | . | . | . | .05 | .18 | .32 | .37 | .03 | .26 | .51 | .63 | .66 | .70 | .72 | . | . | .12 | .29 | .42 | .53 | .52 |
| RF | . | . | . | .08 | .20 | .24 | .24 | . | . | . | .02 | .04 | .10 | .04 | . | .03 | .35 | .64 | .68 | .77 | .82 |
| AB | . | . | . | .05 | .21 | .33 | .35 | .03 | .09 | .32 | .41 | .43 | .54 | .52 | . | .04 | .22 | .51 | .54 | .65 | .65 |
| XGB | . | . | . | .16 | .48 | .70 | .64 | . | . | .17 | .57 | .68 | .78 | .83 | . | . | . | .08 | .20 | .22 | .29 |
| MLP | . | . | . | .07 | .15 | .30 | .31 | .13 | .55 | .87 | .96 | .96 | .99 | .98 | . | .13 | .41 | .53 | .58 | .63 | .58 |

Table 7: The slopes computed for the UGR16 datasets

| | V1 | | V2 | | V3 | |
|---|---|---|---|---|---|---|
| Slope | 1.88207 | AB | 1.49998 | MLP | 1.48212 | MLP |

Table 8: UGR16 datasets - true performance results (recall)

| | V1 | V2 | V3 |
|---|---|---|---|
| KNN | 0.641 | 0.315 | 0.391 |
| SVM | 0.762 | 0.236 | 0.268 |
| DT | 0.917 | 0.565 | 0.753 |
| RF | 0.896 | 0.395 | 0.411 |
| AB | 0.916 | 0.575 | 0.746 |
| XGB | 0.926 | 0.807 | 0.903 |
| MLP | 0.923 | 0.62 | 0.778 |

## 6 Conclusions

Machine learning techniques require high-quality datasets. An effective method for assessing the quality of a dataset helps understand how the quality of the dataset affects the performance results and can be instrumental to solve problems related to the degradation of the model performance after the move to production. We believe that the dataset quality has to be addressed and assessed prior to any ML application.

In our previous papers [8][32], we presented an effective method for the dataset quality assessment based on the permutation testing. The technique is based on well-known ML classifiers. In this paper, we investigated the limits of detection of this methodology, that is, how sensitive is our method to small quality problems in the dataset. For that purpose, we investigated deep permutations, that is, permutations of very small parts of the datasets. The theoretical basis and the conducted experiments prove that the method is effective. It is worth adding, however, that our method allows for the evaluation of a dataset, but does not solve the problem of building a high-quality dataset.

In future work, we aim to define the general slope metric more appropriate for assessing every dataset, which will include the solution of the detection limit. Also, we would like to leverage available metadata to describe Root Cause Analysis (RCA) of quality decrease. Moreover, we plan to improve the implementation of the proposed method to allow higher adoption in the community.

## Acknowledgment

This work is partially funded by the European Union's Horizon 2020 research, innovation programme under the Marie Skłodowska-Curie grant agreement No 893146, by the Agencia Estatal de Investigación in Spain, grant No PID2020-113462RB-I00, and by the Ministry of Interior of the Czech Republic (Flow-Based Encrypted Traffic Analysis) under grant number VJ02010024. The authors would like to thank Szymon Wojciechowski for his support on the Weles tool.

## References

1. Anderson, M.J.: Permutational Multivariate Analysis of Variance (PERMANOVA), pp. 1–15. John Wiley  Sons, Ltd (2017). https://doi.org/10.1002/9781118445112.stat07841
2. Apruzzese, G., Laskov, P., Tastemirova, A.: Sok: The impact of unlabelled data in cyberthreat detection (05 2022). https://doi.org/10.48550/arXiv.2205.08944
3. Auer, F., Felderer, M.: Addressing data quality problems with metamorphic data relations. In: IEEE/ACM 4th International Workshop on Metamorphic Testing (MET). pp. 76–83 (2019). https://doi.org/10.1109/MET.2019.00019
4. Batarseh, F.A., Freeman, L., Huang, C.: A survey on artificial intelligence assurance. Journal of Big Data **8**(1) (2021). https://doi.org/10.1186/s40537-021-00445-7
5. Bergman, M., Milo, T., Novgorodov, S., Tan, W.C.: Query-oriented data cleaning with oracles. In: Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data. p. 1199–1214. SIGMOD '15, Association for Computing Machinery, New York, NY, USA (2015). https://doi.org/10.1145/2723372.2737786
6. Bhatt, S., Sheth, A., Shalin, V., Zhao, J.: Knowledge graph semantic enhancement of input data for improving ai. IEEE Internet Computing **24**(2), 66–72 (2020). https://doi.org/10.1109/MIC.2020.2979620
7. Caiafa, C.F., Zhe, S., Toshihisa, T., Pere, M.P., Solé-Casals, J.: Machine learning methods with noisy, incomplete or small datasets. Applied Sciences **11**(9) (2021). https://doi.org/https://doi.org/10.3390/app11094132
8. Camacho, J., Wasielewska, K.: Dataset quality assessment in autonomous networks with permutation testing. In: IEEE/IFIP Network Operations and Management Symposium (NOMS). pp. 1–4 (2022). https://doi.org/10.1109/NOMS54207.2022.9789767
9. 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 **9**, 5371–5396 (2021). https://doi.org/10.1109/ACCESS.2020.3048319
10. Cordeiro, F.R., Carneiro, G.: A survey on deep learning with noisy labels: How to train your model when you cannot trust on the annotations? In: 33rd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI). pp. 9–16 (2020). https://doi.org/10.1109/SIBGRAPI51738.2020.00010

11. Ding, J., Li, X.: An approach for validating quality of datasets for machine learning. In: 2018 IEEE International Conference on Big Data (Big Data). pp. 2795–2803 (2018). https://doi.org/10.1109/BigData.2018.8622640
12. Dudjak, M., Martinovic, G.: An empirical study of data intrinsic characteristics that make learning from imbalanced data difficult. Expert Syst. Appl. **182**, 115297 (2021)
13. Elsayed, M.S., Le-Khac, N.A., Jurcut, A.D.: Insdn: A novel SDN intrusion dataset. IEEE Access **8**, 165263–165284 (2020). https://doi.org/10.1109/ACCESS.2020.3022633
14. Engelen, G., Rimmer, V., Joosen, W.: Troubleshooting an intrusion detection dataset: the cicids2017 case study. In: 2021 IEEE Security and Privacy Workshops (SPW) (2021). https://doi.org/10.1109/SPW53761.2021.00009
15. Gupta, S., Gupta, A.: Dealing with noise problem in machine learning datasets: A systematic review. Procedia Computer Science **161**, 466–474 (2019). https://doi.org/10.1016/j.procs.2019.11.146, 5th Information Systems International Conference, Surabaya, Indonesia
16. Ibrahim, M., Helmy, Y., Elzanfaly, D.: Data quality dimensions, metrics, and improvement techniques. Future Computing and Informatics Journal **6**, 25–44 (07 2021). https://doi.org/10.54623/fue.fcij.6.1.3
17. Joyce, R.J., Raff, E., Nicholas, C.: A framework for cluster and classifier evaluation in the absence of reference labels. AISec '21, Association for Computing Machinery, New York, NY, USA (2021). https://doi.org/10.1145/3474369.3486867, https://doi.org/10.1145/3474369.3486867
18. Lindgren, F., Hansen, B., Karcher, W., Sjöström, M., Eriksson, L.: Model validation by permutation tests: Applications to variable selection. Journal of Chemometrics **10** (1996)
19. MacDougall, D., Crummett, W.B.: Guidelines for data acquisition and data quality evaluation in environmental chemistry. Analytical Chemistry **52**(14), 2242–2249 (1980). https://doi.org/10.1021/ac50064a004
20. 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. Computers Security **73**, 411–424 (2018). https://doi.org/10.1016/j.cose.2017.11.004
21. Ojala, M., Garriga, G.: Permutation tests for studying classifier performance. Journal of Machine Learning Research **11**, 1833–1863 (06 2010)
22. Pesarin, F., Salmaso, L.: The permutation testing approach: a review. Statistica **70**(4), 481–509 (2010)
23. Pesarin, F., Salmaso, L.: A review and some new results on permutation testing for multivariate problems. Statistics and Computing **22**(2), 639–646 (2012)
24. Pin, K., Kim, J.Y., Chang, J.H., Nam, Y.: Quality evaluation of fundus images using transfer learning. In: International Conference on Computational Science and Computational Intelligence (CSCI). pp. 742–744 (2020). https://doi.org/10.1109/CSCI51800.2020.00139
25. Sahu, A., Mao, Z., Davis, K., Goulart, A.E.: Data processing and model selection for machine learning-based network intrusion detection. In: 2020 IEEE International Workshop Technical Committee on Communications Quality and Reliability (CQR). pp. 1–6 (2020). https://doi.org/10.1109/CQR47547.2020.9101394
26. Sarker, I., Kayes, A.S.M., Badsha, S., Alqahtani, H., Watters, P., Ng, A.: Cybersecurity data science: an overview from machine learning perspective. Journal of Big Data **7** (07 2020). https://doi.org/10.1186/s40537-020-00318-5

27. Shaukat, K., Luo, S., Varadharajan, V., Hameed, I.A., Xu, M.: A survey on machine learning techniques for cyber security in the last decade. IEEE Access **8**, 222310–222354 (2020). https://doi.org/10.1109/ACCESS.2020.3041951
28. Soukup, D., Tisovčík, P., Hynek, K., Čejka, T.: Towards evaluating quality of datasets for network traffic domain. In: 17th International Conference on Network and Service Management (CNSM). pp. 264–268 (2021). https://doi.org/10.23919/CNSM52442.2021.9615601
29. Stapor, K., Ksieniewicz, P., García, S., Woźniak, M.: How to design the fair experimental classifier evaluation. Applied Soft Computing **104**, 107219 (2021). https://doi.org/10.1016/j.asoc.2021.107219
30. Taleb, I., El Kassabi, H., Serhani, M., Dssouli, R., Bouhaddioui, C.: Big data quality: A quality dimensions evaluation (07 2016). https://doi.org/10.1109/UIC-ATC-ScalCom-CBDCom-IoP-SmartWorld.2016.0122
31. Udeshi, S., Jiang, X., Chattopadhyay, S.: Callisto: Entropy-based test generation and data quality assessment for machine learning systems. In: IEEE 13th International Conference on Software Testing, Validation and Verification (ICST). pp. 448–453. IEEE Computer Society, Los Alamitos, CA, USA (10 2020). https://doi.org/10.1109/ICST46399.2020.00060
32. Wasielewska, K., Soukup, D., Čejka, T., Camacho, J.: Dataset quality assessment with permutation testing showcased on network traffic datasets (6 2022). https://doi.org/10.36227/techrxiv.20145539.v1
33. Webb, G.I., Lee, L.K., Goethals, B., Petitjean, F.: Analyzing concept drift and shift from sample data. Data Mining and Knowledge Discovery **32**, 1179–1199 (9 2018). https://doi.org/10.1007/s10618-018-0554-1
34. Wu, M., Yin, X., Li, Q., Zhang, J., Feng, X., Cao, Q., Shen, H.: Learning deep networks with crowdsourcing for relevance evaluation. EURASIP Journal on Wireless Communications and Networking **82**, 1687–1499 (2020). https://doi.org/10.1186/s13638-020-01697-2
35. Yoon, J., Arik, S., Pfister, T.: Data valuation using reinforcement learning. In: III, H.D., Singh, A. (eds.) Proceedings of the 37th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 119, pp. 10842–10851. PMLR (7 2020), https://proceedings.mlr.press/v119/yoon20a.html
36. Zhou, Z.Q., Xiang, S., Chen, T.Y.: Metamorphic testing for software quality assessment: A study of search engines. IEEE Transactions on Software Engineering **42**(3), 264–284 (2016). https://doi.org/10.1109/TSE.2015.2478001