Machine Learning Metrics for Network Datasets Evaluation

Dominik Soukup[1][0000−0002−4737−8735], Daniel Uhříček[2][0000−0001−7339−4803], Daniel Vašata[1][0000−0003−0616−4340], and Tomáš Čejka[3][0000−0001−7794−9511]

[1] Faculty of Information Technology, CTU in Prague, Prague, Czech Republic
{soukudom,vasatdan}@fit.cvut.cz
[2] Faculty of Information Technology, Brno University of Technology, Brno, Czech Republic
iuhricek@fit.vut.cz
[3] CESNET a.l.e., Prague, Czech Republic
cejkat@cesnet.cz

Abstract. High-quality datasets are an essential requirement for leveraging machine learning (ML) in data processing and recently in network security as well. However, the quality of datasets is overlooked or underestimated very often. Having reliable metrics to measure and describe the input dataset enables the feasibility assessment of a dataset. Imperfect datasets may require optimization or updating, e.g., by including more data and merging class labels. Applying ML algorithms will not bring practical value if a dataset does not contain enough information. This work addresses the neglected topics of dataset evaluation and missing metrics. We propose three novel metrics to estimate the quality of an input dataset and help with its improvement or building a new dataset. This paper describes experiments performed on public datasets to show the benefits of the proposed metrics and theoretical definitions for more straightforward interpretation. Additionally, we have implemented and published Python code so that the metrics can be adopted by the worldwide scientific community.

1 Introduction

Network traffic analysis is a key area to ensure the security of our networks and protection against attacks. Due to the growing amount of traffic and increasing portion of encrypted commutation, it is challenging to use traditional methods based on deep packet inspection (DPI) and application data analysis [17]. Machine learning (ML) has promising results in detecting security events even in such a challenging environment and helps to increase the level of network security. Many researchers have already published their solutions to classify network traffic or identify malware in encrypted traffic [1,13,16]. However, used datasets can include errors such as mislabeled, duplicated, and missing data or do not include enough information to provide reliable classification [10,20]. This approach also brings the question of how to ensure that the results of ML solutions will be consistent over time and applicable among different environments. Due to this, it is important to enhance ML automation, e.g., with Active learning (AL) and data drift detection methods. AL benefits from query strategy algorithms to select the most valuable samples for the dataset. Data drift detection can identify changes in ML performance to trigger training. Unfortunately, most of the community is focused on ML performance metrics instead of the dataset, which is necessary for good results [21]. The wrong or bad quality dataset can lead to false positive results that can have a negative impact on network security. Research regarding dataset quality is in the early stage, but the first prototypes are being delivered [12,21].

In this paper, we bring novel metrics that assess dataset quality and suitability for identified network traffic classification tasks. We react to the current state, where the quality of many publicly available datasets is assessed only by the total amount of flows, which can be insufficient. Large datasets are mainly useful for the intensive evaluation of developed ML algorithms, but applying them in training is more computationally demanding. Additionally, even a large stable dataset can become obsolete sooner or later in an evolving production network, and the precision level of ML models is affected, as it is discussed by Brabec et al. [2]. Network traffic is dynamic and requires an up-to-date dataset for reliable results. This requirement is specific to the network domain, and other domains, such as image classification, are not necessarily sensitive to such dynamic behaviour over time. We investigate the current state of the art and identify its limitation. Metrics from Wasielewska et al. [21], Maillor et al. [14], and Lorena et al. [12] bring promising results; however, our evaluation of public networking datasets shows limitations in the metric calculation, normalization, and robustness.

To evaluate datasets, we use three novel metrics that are capable of universally assessing linear and non-linear multi-class tasks. The identified metrics can be further used by AL or data drift methods to more accurately assess the dataset's quality over time to build a relevant dataset for the target use case. The main contributions of this paper are:

- We propose three novel metrics that enhance the current state of the art in evaluating network traffic datasets.
- We provide theoretical definitions and interpretation of the proposed metrics.
- We evaluate the proposed metrics in detail on publicly available datasets to demonstrate described benefits.
- We share our implemented code publicly for further use by the community.

The rest of the article is organized as follows. Sec. 2 introduces related work in the literature. Sec. 3 describes in detail the proposed metric and provides theoretical consideration. Sec. 4 contains experimental setup used datasets and software package used for evaluation. Sec. 5 present findings from experiments carried out on publicly available datasets. Finally, Sec. 6 contains conclusions and discusses future work.

2 Related Work

Usually, the quality of ML-enabled systems is assessed based on the used ML algorithm and its performance metrics. Celdrán et al. [3] propose the RITUAL platform to quantify the trustworthiness of supervised ML algorithms. A similar approach based on ML performance is researched by Koh et al. [8], who suggest a solution based on benchmarking ML models using a collection of predefined datasets. The goal is to test the generalization of ML models over a variety of unseen data from different domains.

These methods are beneficial for ML model verification if we have already known datasets. For unknown datasets, this approach provides limited value. We must analyze if the input dataset provides reasonable structure and information to apply ML techniques. This area is rarely explored, especially in the network traffic datasets domain. Without this, we can have volatile results, leading to many unwanted false positive defections.

Another approach in the literature is leveraging normalization to improve overall dataset quality. The effect of data normalization and dimensional reduction was studied by Obaid et al. [15] in the intrusion detection NSL-KDD dataset. Gonzalez [5] proposed a method to assess the influence of specific data preparation steps on the model performance. Yoon et al. [22] proposed a novel meta-learning framework for data evaluation that is jointly optimized with the target task predictor model. These approaches focus on dataset optimization that modifies the input dataset to improve ML model operation. However, the impact or correlation with dataset quality is not considered.

Some papers examine data quality and what prerequisites should be considered for a good dataset. Chen et al. [4] describe data quality attributes: comprehensiveness, correctness, and variety used to enhance data quality to improve ML results. The list of attributes for data quality assessment varies among authors who use different segmentation. For example, Lee et al. [11] use categories: Intrinsic, Accessibility, Contextual, and Representational. The need to extend data quality to the dataset quality is introduced by Soukup et al. [20]. He suggests definitions of several categories (good/minimal/better dataset) to consider the complete view of the whole dataset. Wasielewska et al. [21] introduces a dataset quality assessment metric based on permutation tests with different permutation levels. This work provides promising results of binary classification.

The area of dataset quality metrics is also analyzed by researchers from other domains. Lorena et al. [12] published a survey of data complexity measures. Based on the realized survey, the author group the data complexity measure into the following groups: (i) feature-based, (ii) linearity, (iii) neighborhood, (iv) network, (v) dimensionality, and (vi) class imbalance. These groups include 22 metrics. The main aim of these metrics is to characterize datasets to help researchers to select learning and preprocessing techniques on an unknown domain; however, evaluation on existing datasets is missing. Also, some measures are defined for binary classification problems only, and any multiclass problem must be first decomposed into multiple binary sub-problems. Several metrics are limited to linear tasks only. While evaluating the implementation of the proposed metrics created by Komorniczak et al. [9], we encountered memory errors for bigger datasets due to the high memory requirements of selected metrics. Maillor et al. [14] aimed to deal with informative metrics for big datasets. The author implements metrics from Lorena et al. in Apache Spark to allow big data processing. Also, the paper proposes two novel metrics focused on dataset density. These metrics are based on fixed classifiers – the nearest neighbors (1NN) and decision tree (DT) which can lead to non-optimal results and are less universal in different classification scenarios.

3 Proposed Dataset Metrics

This section introduces suggested new metrics with their benefits and expected behavior. To the best of our knowledge, there are no metrics that evaluate the quality of network traffic datasets from the same perspective as ours. The defined metrics are evaluated in detail in Sec. 5.

3.1 Metric 1 (M_1) - Dataset Redundancy

The first metric is focused on the level of dataset size redundancy. Using this measure, one can estimate what portion of the original dataset can be randomly removed while keeping the classification performance drop below the certain controlled level. Zero redundancy indicates not enough data for the classification task. For evaluation, we use the pool of classifiers and acceptance level α to generally assess the level of redundancy with a certain probability.

Dataset redundancy was mentioned as one of the requirements by Soukup et al. [20] since many available datasets are unnecessarily large to provide the same quality of results. Maillo et al. [14] introduced a dataset redundancy method based on 1NN density comparison between the original dataset and reduced dataset by 50%. The analysis is done for redundancy levels of 25%, 50%, and 75% without more detailed steps.

Our metric defines redundancy based on the performance metric, e.g., F1 score, from the pool of selected classifiers. Let D denote the dataset and $\alpha \in (0,1)$ be the relative acceptance level of the performance. Moreover, let $\mathcal{C} = \{c_1, \ldots, c_m\}$ be a pool of m classifier models. For $\varphi \in (0,1)$ and for $i = 1, \ldots, k$, where k is some positive number, we denote by $D_{\varphi,i}$ the ith dataset of relative size φ sampled randomly (without replacement) from D. Hence, $D_{\varphi,i}$ is a subset of D that contains $\lfloor \varphi |D| \rfloor$ data points, where $|D|$ is the number of data points in D. $D_{\varphi,i}$ is used for training of the models from the pool $\mathcal{P}$. Moreover, by $T_{\varphi,i}$ we denote the remaining part of D, i.e.

$$T_{\varphi,i} = D \setminus D_{\varphi,i},$$

that is used for testing of the performance of each trained model.

For a given $\varphi \in (0,1)$ each $i = 1, \ldots, k$ and $j = 1, \ldots, m$ one takes the model $c_j \in \mathcal{C}$, trains it using dataset $D_{\varphi,i}$ and evaluates it using dataset $T_{\varphi,i}$. Let us denote the obtained testing performance by $\tau(c_j, D_{\varphi,i})$.

By taking $\min_i \tau(c_j, D_{\varphi,i})$ we observe the minimal performance of a jth model over all random splits of the dataset at a level φ. Now we will compare this value with the lowest acceptable performance level that is given by $\tau_\alpha = \alpha \max_{i,j} \tau(c_j, D_{0.99,i})$, i.e. is relative according to maximal performance measured over all dataset splits and all models from the pool for $\varphi = 0.99$. If there is at least one model from the pool of classifiers $\mathcal{C}$ for which its minimal performance $\min_i \tau(c_j, D_{\varphi,i})$ is above this lowest acceptable performance level, the redundancy of the dataset should be larger than $1 - \varphi$.

This means that the redundancy can be defined as maximum of all possible $1 - \varphi$ satisfying the above condition $\max_j \min_i \tau(c_j, D_{\varphi,i}) \geq \tau_\alpha$. Formally,

$$M_1 = 1 - \inf_{\{\varphi \mid \max_j \min_i \tau(c_j, D_{\varphi,i}) \geq \tau_\alpha\}} \varphi. \tag{1}$$

To implement the search for the infimum we used the half splitting search.

Note that there can be more policies to set the redundancy level where we, for example, do not require to fit above τ_α for all runs of at least one model. For stability purposes, we take a strict policy that insists on acceptable results for all runs from a specific percentage level. The evaluation process of single ML model is depicted in Fig. 1. The metric domain is $[0, 1]$, and it describes the percentage size of the dataset that is redundant.

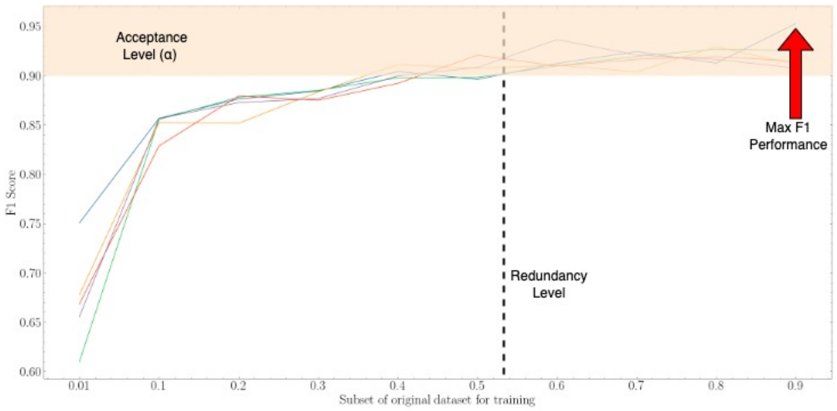
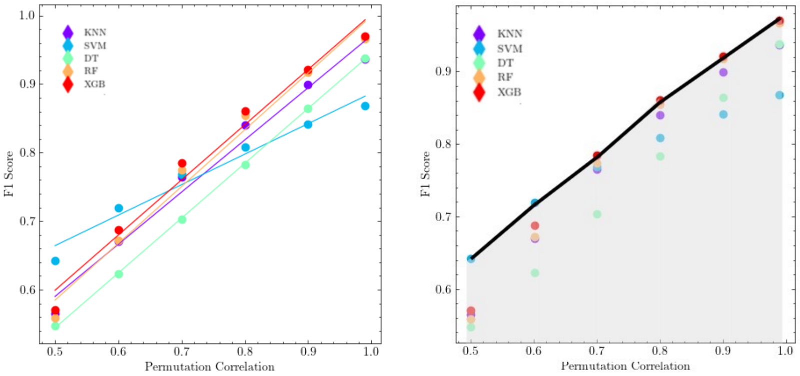


Fig. 1: Visualization of M_1 calculation workflow. The figure depicts single model with five independent runs.

3.2 Metric 2 (M_2) - Dataset Association Quality

The second metric evaluates the level of association between labels and respective data. Especially for the public datasets, we don't know how the dataset was collected and if it's meaningful to apply ML algorithm on such a dataset.



(a) Original permutation slope visualization which selects DT as the top performing ML model, however, XGB is more suitable

(b) Maximized Area Under Curve (AUC) accross the whole pool of classifiers

Fig. 2: Comparison of original permutation slope metric with the proposed M_2 metric

The level of association is estimated based on permutation tests which are interpreted by this novel metric. As a result, we get an estimate of how strong is the connection between data and related labels.

Permutation tests [18] are known area in state of the art. Wasielewska et al. [21] introduced permutation tests using different permutation percentage levels to estimate the relationship between labels and data with a certain sensitivity. During our evaluation on public datasets, we identified incorrect behavior of permutation slope that is used for the interpretation of results. In several cases, it selected the wrong ML model as a top performing, as depicted in Fig. 2a, and the normalization of received values is not defined. We follow up on this work and enhance this limitation. For each permutation level from 0.5 to 0.99 the best performing ML classifier from the pool of classifiers is selected, which yields a piecewise-linear curve f in the domain $[0.5, 0.99]$. Then the Area Under Curve (AUC) is calculated and normalized.

This gives us the best possible association of labels and data among all selected models in the pool of classifiers. Moreover, we extended this metric to support both binary and multiclass classifications. A comparison of permutation slope and the proposed metric calculation is depicted in Fig. 2.

To define the metric formally, let as assume that γ indicates the permutation level and for $i = 1, \ldots, k$ we denote by $P_{\gamma,i}$ the ith sample given as the original dataset D where the fraction of γ of labels was randomly permuted. On this sample dataset one uses cross-validation with ℓ folds to evaluate each model c_j from the pool $\mathcal{C}$ of m classification models. Let us denote by $\bar{\tau}(c_j, P_{\gamma,i})$ the obtained performance.

The function f at permutation level γ is defined by

$$f(\gamma) = \max_j \frac{1}{k} \sum_{i=1}^{k} \bar{\tau}(c_j, P_{\gamma,i}).$$

The area under curve (AUC) is then given by

$$\mathrm{AUC} = \int_{0.5}^{1} f(\gamma) \mathrm{d}\gamma.$$

Finally, the metric M_2 is given by the normalized value of the AUC:

$$M_2 = \frac{0.5 - \mathrm{AUC}}{0.25 \cdot \max_\gamma f(\gamma)}. \tag{2}$$

In calculations the function f is taken as a piecewise linear function evaluated only at breaking points $\gamma = 0.5, 0.6, \ldots, 0.9, 0.99$. The metric is shown in Fig. 2b.

The normalization is done by subtracting the AUC from 0.5 which effectively means taking the Area Above Curve. Then it is divided by the highest score of the best ML model, which effectively changes the range of possible values of f to $[0, 1]$. Then the divison by 0.25 is to make it relative to the largest value possible after the previous transformations.

The metric domain is $[0, 1]$ and it corresponds to the level of association between data and labels in the dataset. Generally, the permutation slope and its sensitivity are dependent on the imbalanced ratio and selected performance metric [21]. Therefore, this metric cannot be used to easily compare the results of different datasets. However, the focus of this paper is on a single dataset.

3.3 Metric 3 (M_3) - Dataset Class Similarity

The third metric looks at the dataset classes. We propose a method to estimate how instances of different classes are similar to each other. In other words, how complex the classification task generally is on the input dataset. The metric measures relative class similarity using autoencoders and their respective reconstruction error. Calculated relative similarity lays out direct indicators of how prone are other machine learning models to misclassifications.

Methods of representation learning, and specifically autoencoders, have already been successfully applied to network data. For example, Zhang et al. [23] proposed a framework for network data feature extraction using autoencoders and successfully evaluated extracted features based on their clustering performance. Hwang et al. [6] applied autoencoders to image datasets comparison and examined how reconstruction error can be used to predict inter-dataset similarity. Their solution introduces multiple autoencoders, each trained on a separate dataset and used to calculate reconstruction error on rest. Moreover, they state that the method has the potential for inter-class similarity within the same dataset. Based on their results, authors discussed that such or similar approach might be beneficial for inter-class similarity, but, in their words, more experiments are needed. Our metric follows up on their work, showing how a similar approach would be applied to network and security datasets. We leverage the fact that security datasets are often imbalanced and have some majority (or benign class), and instead of training n autoencoders for each class, we train only one autoencoder on the benign class giving us information about how much other classes could be confused with the majority class. Later in our experiments, we see that this method gives consistent results and corresponds well both with our prior knowledge about tested datasets and with other evaluated metrics.

The metric is calculated in two phases – initialization phase and the evaluation phase. In the initialization phase, an autoencoder model is trained on one of the classes as a base class (most commonly a majority class or a benign class). The instances of the base class are split in a ratio of 8:2 to training and validation datasets. During our experiments, we encountered that the relative similarity (calculated relative to the base class) is consistent even when the underlying model is randomly reinitialized differently and has different numbers of hidden layers and their sizes.

In the evaluation phase, the trained autoencoder is then used to calculate the reconstruction error for each of the classes – both in absolute values and relative values to the class used for training. In our reports, we either state all values (absolute and relative) for non-base classes or the weighted average of the relative error to describe the dataset as a whole. Formally, the described process can be stated as follows. We denote individual classes as C_i and base class as B. Reconstruction error for instance x is denoted as $\mathrm{err}_{\mathcal{A}}(x)$. We define mean absolute error (MAE) for each of the classes based on the reconstruction error from the trained autoencoder (Eq. 3). MAE is used to define mean relative error (MRE) for each of the classes (Eq. 4). Finally, metric M_3 is defined as a weighted average over MRE values of non-base classes (Eq. 5).

$$\mathrm{MAE}_{C_i} = \frac{1}{|C_i|} \sum_{x \in C_i} \mathrm{err}_{\mathcal{A}}(x) \tag{3}$$

$$\mathrm{MRE}_{C_i} = \frac{\mathrm{MAE}_{C_i}}{\mathrm{MAE}_B} \tag{4}$$

$$M_3 = \frac{1}{\sum\limits_{C_i, C_i \neq B} |C_i|} \sum_{C_i, C_i \neq B} |C_i| \cdot \mathrm{MRE}_{C_i} \tag{5}$$

Since metric M_3 represents average relative reconstruction error over all instances of non-base classes, the domain are positive numbers – multiples of reconstruction error on base class. Corresponding with our experiments in Sec. 4, $M_3 <= 1$ means that autoencoder performs similarly for all the classes and from this point of view, classes are similar, or even some classes might seem as a subset of the base class in the feature space. On the other hand, if $M_3 > 1 + K$, classes are different in the feature space and might be easier to separate from the base class.

Table 1: Summary of selected datasets describtion for defined experiments

Dataset	Samples	Features	Classes
DoH [7]	20,000	24	2
TLS [13]	125,000	44	5
CICIDS2017 [19]	71,599	78	6
CICIDS2017 Fixed [10]	63,321	86	6

4 Experiments setup

This paper introduces novel metrics to evaluate dataset suitability for the target use case. We are considering binary and multi-classification tasks that are involved in selected experiments. In this section, we describe our experiments and selected datasets for evaluation of proposed metrics.

4.1 Datasets

The applicability of the proposed solution is demonstrated on three publicly available network datasets – DoH – Real-World [7], CESNET-TLS22 [13], and CICIDS2017 [19]. DNS over HTTPS (DoH) is an encrypted communication protocol with a domain name resolver that increases the privacy of internet users. Jerabek et al. [7] published this dataset from a large ISP network with labels for binary classification. Luxemburk et al. [13] recently collected a dataset from the national network CESNET2. It includes 191 labels of network services using TLS traffic. Sharafaldin et al. [19] published the CICIDS2017 dataset that included malware samples with classes of benign and seven common attacks. This dataset was investigated by Lanvin et al. [10], who identified several errors in this dataset and published a new version with the necessary fixes.

To make easier development and proper evaluation of our metric, we selected subsamples of input datasets. The description of datasets for the provided experiments is summarized in Tab. 1. Analysis of findings from experiments is described on Sec. 5.

4.2 Experiments

After the introduction of the proposed metrics, we provide a list of experiments to demonstrate the value on real datasets. The results of these experiments are described in Sec. 5.

Case Study 1: Evaluation for binary classification In this case study, we take the DoH dataset for several tests. In the first test, we test sensitivity to a reduced size of the dataset. We started with the original dataset, which set the baseline, and then we randomly removed the defined portion. The second test investigates the impact of mislabels, which we created by randomly switching specific percentages of labels in a balanced way. The last test analyses the sensitivity of dataset imbalance. The imbalance is created by setting the different ratio between negative (non-doh) and positive (doh) classes, however, the total size is always the same.

Case Study 2: Evaluation for multi-class classification The second case study contains a multi-class CESNET-TLS22 dataset. The main aim is to validate metrics application on a multi-class dataset that has not been considered in several related papers and related metrics. Moreover, the classification of TLS encrypted traffic is more complex than the previous binary classification for DoH.

Case Study 3: Evaluation of bad and corrected dataset The final case study is focused on the CICIDS2017 dataset that contains known errors identified by Lanvin et al. [19]. His paper describes each error with respective evaluation in detail. However, a comparison between completely fixed and original datasets is missing. This case study evaluates the CICIDS2017 dataset before and after the application of suggested fixes to verify findings from other researchers who used this dataset.

4.3 Software package

All introduced metrics are implemented in the public Github repository[4]. The implementation is done in Python and in Jupyter Notebook, which is popular in the community and allows easy usage of proposed methods. Part of the notebook is an enhanced visualization of the proposed metric for further analysis of their behavior. Together with the novel metric, we included other known dataset metrics to allow a complete valuation of the input dataset.

5 Evaluation of Introduced Metrics

In this section, we describe the received results from defined experiments. For all metric, we use consistent configuration to provide reliable verification. Parameters of input datasets are described in each experiment.

5.1 Case Study 1: Evaluation for binary classification

In the first case study, we analyze the proposed metrics on binary classification tasks for DoH traffic. For the first test, we take 10,000 samples from DoH — Real-World and calculate all metrics to set the baseline for the reduction of samples. We reduced 10, 30, and 50% of the samples to see the impact of the introduced metric. The results are summarized in Tab. 2a. The original input dataset gets M_1 score of 30.8% with sensitivity parameter α equal to 1%. When we remove

[4] `https://github.com/soukudom/NDVM`

	M_1	M_2	M_3	XGB F1
Dataset 0%	0.308	0.437	0.732	0.975
Dataset 10%	0.257	0.429	0.769	0.974
Dataset 30%	0.0	0.442	0.767	0.974
Dataset 50%	0.0	0.435	0.799	0.964

(a) (Test 1) Consistency on the reduced dataset. The percentage represents a removed part of the dataset. Removed records are balanced over both classes. The last column represents F1 score for XGB classifier.

	M_1	M_2	M_3	XGB F1
Mis. 0%	0.308	0.437	0.732	0.975
Mis. 10%	0.0	0.397	0.729	0.865
Mis. 20%	0.0	0.324	0.806	0.865
Mis. 30%	0.0	0.132	0.810	0.764

(b) (Test 2) Sesitivity to mislabels. The percentage represents amount of created mislabels in the whole dataset in a balanced way. The last column represents F1 score for XGB classifier.

	M_1	M_2	M_3	XGB F1
Imbl. 50/50%	0.492	0.429	0.731	0.982
Imbl. 60/40%	0.446	0.413	0.744	0.982
Imbl. 70/30%	0.140	0.371	0.759	0.982

Table 3: (Test 3) Dependency on imbalanced ratio. The percentage represents ratio between negative (non-DoH) and positive (DoH) class. The last column represents F1 score for XGB classifier.

defined portions of samples, we can see a decreasing trend for this metric. For a reduction 30%, we can see M_1 score of 0, and other metrics, including the F1 score, are consistently at the same level with the original dataset since this is the threshold value for M_1. For a reduction of 50%, we can see the continuous trend for M_3 and F1 score. However, the change is not significant since there is still value in the dataset which is reflected by M_2.

The second test is based on the same input dataset as in the first test. Tab. 2b summarizes the impact of mislabels we added to each class in a balanced way. We can see a clear trend in all metrics. Especially, M_2 shows the decrease of relevant association between class labels and data.

In the last test, we increased the dataset size to 20,000 samples so that we have enough samples in each class for different imbalanced ratios, as we know the minimal size from Test 1. Results are included in Tab. 3. Since the dataset is twice bigger as in previous experiments, we can see an increase of M_1. The F1 score is consistent over all versions, but the dataset quality is consistently lower for all metrics. This dependency is aligned with work from Brabec et al. [2].

5.2 Case Study 2: Evaluation for multi-class classification

In the second case study, we take the CESNET-TLS22 dataset, which contains two types of labels - Category and TLS_SNI. TLS_SNI is a subset of the higher-level Category label. From the original dataset, we created two datasets for this experiment with different label type. We take 5 classes from Category label (Antivirus, Videoconferencing, Streaming media, Analytics & Telemetry, File sharing) and 5 classes with TLS_SNI labels[5], each with 25,000 records.

[5] login.microsoftonline.com, settings-win.data.microsoft.com, outlook.office365.com, api.github.com, v10.events.data.microsoft.com

	M_1	M_2	M_3	XGB F1
TLS (Category)	0.614	0.558	1.489	0.969
TLS (TLS_SNI)	0.967	0.526	4.093	0.998

Table 4: Summary of results for multi-class classification. The last column represents F1 score for XGB classifier.

	M_1	M_2	M_3	XGB F1
CICIDS2017	0.767	0.520	9.956	0.946
CICIDS2017 Fixed	0.783	0.492	6.296	0.975

Table 5: Summary of comparison between original and fixed version of CICIDS2017 dataset. The last column represents F1 score for XGB classifier.

The results of the evaluation are shown in Tab. 4. Even though we have the same dataset used for different use cases, we can see different behavior. Based on the F1 score and M_2 metric, we can see good classification performance and high accuracy of assigned labels. M_3 shows higher classification similarity for TLS_SNI since 4 classes are related to Microsoft and are very similar in comparison with class api.github.com. M_1 indicates higher redundancy for TLS_SNI classification.

5.3 Case Study 3: Evaluation of error and corrected dataset

The last case study evaluates recent findings in the CICIDS2017 dataset from Lanvin et al. [10]. The author identifies several errors in terms of duplicated, misordered data and incoherent timestamps. We compare the original and fixed versions of the CICIDS2017 dataset. We analyze Wednesday traffic with all available labels (BENIGN, DoS Hulk, DoS GoldenEye, DoS slowloris, DoS Slowhttptest, Heartbleed) and limit the size of each class to 5000 samples. Results are summarized in Tab. 5. Even though the fixed version of the dataset contains less amount of samples due to duplicated data, the M_1 and M_2 metrics are almost consistent due to valid corrections of the original CICIDS2017 dataset. M_3 is influenced by the sensitivity of Heartbleed class which has only 11 samples.

As the author briefly said, this dataset has a similar ML classification score. The main reason is due to the wrong collection process, which does not necessarily must cause classification failure. Since our metrics are working with a single dataset, we can confirm findings from Lanvin et al. [10] that datasets are providing similar results. The difference and relationship between two datasets or correct dataset development is a separate challenge that should be investigated.

6 Conclusion

In this paper, we have proposed three novel metrics to estimate the quality of datasets that are usually used to train and evaluate ML models used in network security. The introduced metrics are robust, since we combine several ML models with statistical methods. Contrary to other existing published works, our proposed metrics can be used as a universal assessment procedure to evaluate linear and non-linear tasks over datasets for binary or even multi-class classification.

Experiments on three different case studies tested all the metrics and explained their added value, sensitivity, and interpretation. According to the results, we can see high redundancy and reliability of the labels in all tested datasets, which can be considered good for target classification.

After intentional modifications of the datasets for test purposes (mislabels, samples reduction and imbalance), we showed successful identification of such events/corruptions by our metrics. Surprisingly and most importantly, these evident flaws in the modified datasets were not observable by any change in F1 score, which is commonly used in literature to evaluate ML models. We believe this highlights the urgent need to include additional dataset evaluation techniques as a useful practice in scientific research as well as production deployment of ML technologies. Especially, this is essential during the work with datasets, i.e., creation or optimization either manually or with some automation technology such as Active Learning. It is worth repeating that imperfect datasets used for training can lead to poor performance of the machine learning models that are becoming popular for network security area, such as traffic recognition, detection of security threats, or detection of suspicious behavior of devices.

The main focus of this paper was to evaluate a single dataset. However, during our experiments, we identified dataset relationship comparison as a challenge that should be researched in future work.

Acknowledgement

This research was funded by the Ministry of the Interior of the Czech Republic, grant No. VJ02010024: Flow-Based Encrypted Traffic Analysis, and by the MEYS of the Czech Republic, grant No. LM2023054: e-Infrastructure, and by internal grant of CTU in Prague, grant No. SGS23/207/OHK3/3T/18.

References

1. Blake Anderson and David McGrew. Machine learning for encrypted malware traffic classification: Accounting for noisy labels and non-stationarity. In *23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2017.
2. Jan Brabec et al. On model evaluation under non-constant class imbalance. In *Computational Science (ICCS)*, 2020.
3. Alberto H. Celdrán et al. RITUAL: a Platform Quantifying the Trustworthiness of Supervised Machine Learning. In *18th International Conference on Network and Service Management (CNSM)*, 2022.
4. Haihua Chen et al. Data curation and quality assurance for machine learning-based cyber intrusion detection, 2021.
5. Carlos V. G. Zelaya. Towards explaining the effects of data preprocessing on machine learning. In *35th International Conference on Data Engineering*, 2019.

6. Inseok Hwang et al. Simex: Express prediction of inter-dataset similarity by a fleet of autoencoders. *arXiv preprint arXiv:2001.04893*, 2020.
7. Kamil Jeřábek, Karel Hynek, Tomáš Čejka, and Ondřej Ryšavý. Collection of datasets with dns over https traffic. *Data in Brief*, 42:108310, 2022.
8. Pang Wei Koh et al. WILDS: A Benchmark of in-the-Wild Distribution Shifts. In *Proceedings of the 38th International Conference on Machine Learning*, 2021.
9. Joanna Komorniczak and Pawel Ksieniewicz. problexity — an open-source python library for binary classification problem complexity assessment. *arXiv preprint arXiv:2207.06709*, 2022.
10. Maxime Lanvin et al. Errors in the CICIDS2017 dataset and the significant differences in detection performances it makes. Preprint at `https://hal.science/hal-03775466`, 2023.
11. Yang W. Lee et al. AIMQ: a methodology for information quality assessment. *Information & Management*, 2002.
12. Ana C. Lorena, Luís P. F. Garcia, Jens Lehmann, Marcilio C. P. Souto, and Tin Kam Ho. How complex is your classification problem? a survey on measuring classification complexity. *ACM Comput. Surv.*, 52(5), sep 2019.
13. Jan Luxemburk and Tomáš Čejka. Fine-grained tls services classification with reject option. *Computer Networks*, 220:109467, 2023.
14. Jesus Maillo, Isaac Triguero, and Francisco Herrera. Redundancy and complexity metrics for big data classification: Towards smart data. *IEEE Access*, 8:87918–87928, 2020.
15. Hadeel S. Obaid et al. The impact of data pre-processing techniques and dimensionality reduction on the accuracy of machine learning. In *9th Annual Information Technology, Electromechanical Engineering and Microelectronics Conference (IEMECON)*, 2019.
16. Eva Papadogiannaki and Sotiris Ioannidis. A survey on encrypted network traffic analysis applications, techniques, and countermeasures. *ACM Computing Surveys*, 54(6), 2021.
17. Feargus Pendlebury et al. Tesseract: Eliminating experimental bias in malware classification across space and time. In *Proceedings of the 28th USENIX Conference on Security Symposium*, USA, 2019.
18. Fortunato Pesarin and Luigi Salmaso. A review and some new results on permutation testing for multivariate problems. *Statistics and Computing*, 22(2):639–646, 2012.
19. Iman Sharafaldin, Arash Habibi Lashkari, and Ali A. Ghorbani. Toward generating a new intrusion detection dataset and intrusion traffic characterization. In *International Conference on Information Systems Security and Privacy*, 2018.
20. Dominik Soukup et al. Towards evaluating quality of datasets for network traffic domain. In *17th International Conference on Network and Service Management (CNSM)*, 2021.
21. Katarzyna Wasielewska et al. Dataset Quality Assessment with Permutation Testing Showcased on Network Traffic Datasets. Preprint at `http://dx.doi.org/10.36227/techrxiv.20145539.v1`, 2022.
22. Jinsung Yoon, Sercan Arik, and Tomas Pfister. Data valuation using reinforcement learning. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 10842–10851. PMLR, 13–18 Jul 2020.
23. Yongzheng Zhang, Shuyuan Zhao, and Yafei Sang. Towards unknown traffic identification using deep auto-encoder and constrained clustering. In *Computational Science – ICCS*, 2019.