Methodology for the Detection of Contaminated Training Datasets for Machine Learning-Based Network Intrusion-Detection Systems

Joaquín Gaspar Medina-Arco   Roberto Magán-Carrión   Rafael A. Rodríguez-Gómez   Pedro García-Teodoro
Network Engineering & Security Group (NESG)
Research Centre for Information and Communication Technologies (CITIC-UGR), University of Granada, Granada, Spain.
jgasparmedina@correo.ugr.es, {rmagan, rodgom, pgteodor}@ugr.es

***Abstract*—With the significant increase in cyber-attacks and attempts to gain unauthorised access to systems and information, Network Intrusion-Detection Systems (NIDSs) have become essential detection tools. Anomaly-based systems use machine learning techniques to distinguish between normal and anomalous traffic. They do this by using training datasets that have been previously gathered and labelled, allowing them to learn to detect anomalies in future data. However, such datasets can be accidentally or deliberately contaminated, compromising the performance of NIDSs. This paper addresses the mislabelling problem of real network traffic datasets by introducing a novel methodology that (i) allows analysing the quality of a network traffic dataset by identifying possible hidden or unidentified anomalies and (ii) selects the ideal subset of data to optimise the performance of the anomaly detection model even in the presence of hidden attacks erroneously labelled as normal network traffic.**

***Index Terms*—anomaly detection, NIDS, deep learning, autoencoders, methodology, real network datasets, data quality**

Contribution type: *Research already published*

I. Introduction

Network Intrusion-Detection Systems (NIDSs) are critical for cybersecurity, analyzing network traffic to identify potential attacks and vulnerabilities. They can be categorized based on architecture (host-based, network-based or collaborative) and detection techniques (signature-based, Stateful Protocol Analysis-based or anomaly detection-based). Signature-based NIDSs match network sequences with known attack patterns, while Stateful Protocol Analysis-based NIDSs analyze protocol interactions. Finally, anomaly-detection-based NIDSs detect abnormal traffic behavior which notably differs from normal network traffic.

Various machine learning techniques are employed for anomaly detection in NIDSs. These techniques can be supervised, semi-supervised, or unsupervised, depending on the approach. Dataset labeling accuracy is crucial for training AI models effectively, whether in supervised or unsupervised learning.

Real, synthetic, or composite datasets are used for training NIDS models. Synthetic datasets provide controlled traffic samples but may not fully represent real-world patterns, while real datasets capture actual network traffic but may lack sufficient anomalous samples. In addition, composite datasets combine real and synthetic data to introduce attack patterns.

Ensuring dataset labeling accuracy is crucial to avoid mislabeled data poisoning AI models. This is exemplified by the UGR'16 traffic dataset [1], where unlabelled anomalies affected detection performance [2]. In this work, a methodology is proposed to identify hidden anomalies in real traffic datasets, enhancing labeling reliability and optimizing dataset size for AI model efficiency. Altogether, contributes to build robust NIDS against intentionally or not poisoned datasets. This methodology integrates Kitsune [3], a widely-used NIDS, for demonstrating its efficacy.

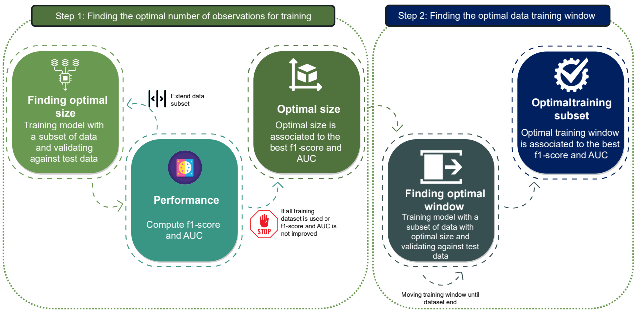
Step 1: Finding the optimal number of observations for training
Step 2: Finding the optimal data training window
Finding optimal size
Training model with a subset of data and validating against test data
Extend data subset
Optimal size
Optimal size is associated to the best f1-score and AUC
Performance
Compute f1-score and AUC
STOP
If all training dataset is used or f1-score and AUC is not improved
Finding optimal window
Training model with a subset of data with optimal size and validating against test data
Moving training window until dataset end
Optimaltraining subset
Optimal training window is associated to the best f1-score and AUC

Figure 1. Proposed methodology workflow.

II. Proposed Methodology

Figure 1 represents the proposed methodology, which will be detailed below. For testing it, the UGR'16 dataset [1] has been used. The UGR'16 dataset was created by the University of Granada in 2016 as a result of capturing the real network traffic of a medium-sized ISP between March and June 2016. Subsequently, during the months of July and August, different attacks such as DoS, botnet, or port scanning were deliberately generated on the same ISP to capture all the traffic so that this subset could be used as a test.

Firstly, it is essential to estimate the optimal number of observations in the entire training dataset that is ideal for training an AI model. The next step is to find the subset of training data composed of this optimal number of observations that maximizes the model's performance while enabling the potential discovery of hidden anomalies within the dataset. To achieve this, metrics are established to measure the quality of each data subset, and stopping mechanisms for the proposed process are determined based on these metrics.

A. Step 1 - Finding the Best Size for the Training Window

The proposed method employs iterative training to determine the ideal training window size. Each iteration progressively increases the number of records used, starting from the first record of the dataset subset for training. After training in each iteration, the resulting model is tested on the designated test dataset, evaluating F1-score and AUC indicators. The optimal training window size is determined by the iteration yielding the highest F1-score and AUC values. An early stopping condition is implemented in this step.

B. Step 2 - Finding the Optimal Data Training Window

After determining the optimal training data window size, the proposed approach involves a repeated training process, shifting the data window through the entire training dataset during each iteration. At the end of each cycle, the resulting model is evaluated using the test dataset, assessing performance indicators such as F1-score and AUC.

C. Hidden Anomaly Detection

The search for the optimal training window in NIDS model training yields a tool for detecting labeling errors or unidentified anomalies in the dataset. Under normal conditions, similar-sized subsets of correctly labeled data should yield comparable results in F1-score and AUC. However, if unlabelled data are used, resulting F1-score and AUC may significantly degrade due to the model learning incorrect patterns. Analyzing the evolution of these metrics can pinpoint dataset sections with potential labeling issues, aiding specialists in identifying dataset quality problems.

D. Results

Figure 2 illustrates the search for the optimal window size. During the initial iterations, the performance metrics oscillate, which may be attributed to the lack of training data that hindered the model's ability to generalize correlations. By iteration 8, stabilization occurs with over 8 days of normal traffic, suggesting improved generalization. After 20 days, a qualitative leap is observed in all metrics, indicating consolidated generalization. However, performance plateaus by iteration 40, suggesting maximum generalization. Subsequently, performance significantly drops around May 3, 2016 (iteration 41), likely due to infrequent normal traffic patterns or undetected attacks, leading to process termination based on early stop criteria.

Figure 3 depicts the iterative process of determining the optimal training window, highlighting the initial 40 days of the UGR'16 dataset as most effective. Despite stable AUC values, a notable F1-score decrease occurs in iteration 18, indicating the model's struggle in predicting anomalies, likely due to misclassifications or undetected anomalies, particularly evident from April 16 to May 29, 2016 (iteration 29). Subsequent iterations show limited improvement, suggesting dataset insufficiency beyond June. Futher analysis on F1-score behaviour focused on botnet attacks, suggests the presence of undetected botnet activity in the training set by May 19, 2016.

III. Conclusions and Future Works

The paper introduces a methodology for detecting contaminated data in network traffic datasets used to train ML-based network anomaly detection NIDSs. It aims to determine the optimal training dataset subset size, select the subset maximizing NIDS model performance, and identify labeling issues or polluted data. Testing the UGR'16 dataset with Kitsune using this methodology revealed an improved subset, detected previously undetected botnet attacks in May, and confirmed labeling errors in June. Future work will extend this methodology to other datasets and NIDSs, potentially using metaheuristics like PSO, and explore its applicability against poisoning-type adversary attacks to enhance robustness.

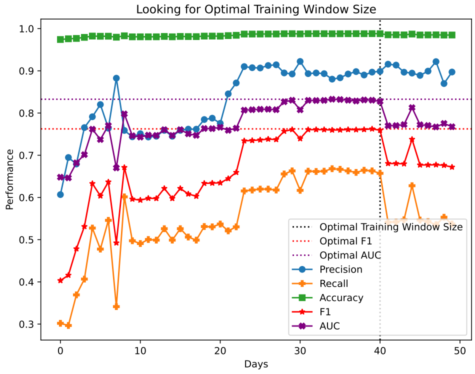

Figure 2. Performance metrics for iterations looking for the training window size.

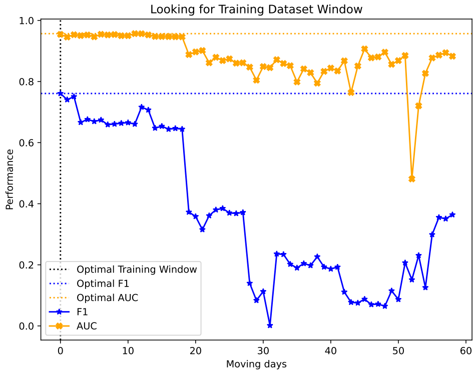

Figure 3. Performance metrics for iterations looking for the training window size.

Acknowledgements

This publication results from the project NetSEA-GPT (C-ING-300-UGR23) funded by Consejería de Universidad, Investigación e Innovación and the European Union through the ERDF Andalusia Program 2021-2027. Additionally, it has also been partially funded by MICIN/AEI/10.13039/501100011033 under grants PID2020-113462RB-I00 and PID2020-114495RB-I00.

References

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

[2] J. G. Medina-Arco, R. Magán-Carrión, and R. A. Rodríguez-Gómez, "Exploring Hidden Anomalies in UGR'16 Network Dataset with Kitsune," in *Flexible Query Answering Systems*, ser. Lecture Notes in Computer Science, H. L. Larsen, M. J. Martin-Bautista, M. D. Ruiz, T. Andreasen, G. Bordogna, and G. De Tré, Eds. Cham: Springer Nature Switzerland, 2023, pp. 194–205.

[3] Y. Mirsky, T. Doitshman, Y. Elovici, and A. Shabtai, "Kitsune: An Ensemble of Autoencoders for Online Network Intrusion Detection," 2018, arXiv:1802.09089 [cs].