Demo: towards reproducible evaluations of ML-based IDS using data-driven approaches

Solayman Ayoubi, Sébastien Tixeuil, Gregory Blanc, Houda Jmila

Demo: Towards Reproducible Evaluations of ML-Based IDS Using Data-Driven Approaches

Solayman Ayoubi
solayman.ayoubi@lip6.fr
Sorbonne University, CNRS, LIP6
Paris, France

Sébastien Tixeuil*
sebastien.tixeuil@lip6.fr
Sorbonne University, CNRS, LIP6
Paris, France

Gregory Blanc
greogry.blanc@telecom-sudparis.eu
SAMOVAR, Télécom SudParis
Institut Polytechnique de Paris
Palaiseau, France

Houda Jmila
houda.jmila@cea.fr
Institute LIST, CEA, Paris-Saclay University
Palaiseau, France

Abstract

Network-based Intrusion Detection Systems (NIDS) are crucial in cybersecurity, but evaluation methodologies are outdated and lack standardization, resulting in incomplete and unreliable assessments. To address these issues, we first proposed a comprehensive evaluation framework for Machine Learning-based Intrusion Detection Systems [1]. This framework accounts for the unique aspects, strengths, and weaknesses of ML algorithms. However, the initial proposition lacked practicality, as it presented an abstract methodology without a substantive solution. In this paper, we present a demo of FREIDA a precise and concrete implementation of our framework, featuring an easy-to-use graphical interface. We also outline FREIDA's evaluation methodology and demonstrate its application in evaluating IDS using a dataset from the literature.

CCS Concepts

• **Security and privacy → Intrusion detection systems**; • **General and reference → Evaluation**; • **Computing methodologies → Machine learning**.

Keywords

Intrusion Detection System, Machine learning, Data-driven Evaluation, Evaluation Tool

ACM Reference Format:
Solayman Ayoubi, Sébastien Tixeuil, Gregory Blanc, and Houda Jmila. 2024. Demo: Towards Reproducible Evaluations of ML-Based IDS Using Data-Driven Approaches. In *Proceedings of the 2024 ACM SIGSAC Conference on Computer and Communications Security (CCS '24), October 14–18, 2024, Salt Lake City, UT, USA.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3658644.3691368

*Also with Institut Universitaire de France.

CCS'24, October 14–18, 2024, Salt Lake City, UT, USA.

ACM ISBN 979-8-4007-0636-3/24/10
https://doi.org/10.1145/3658644.3691368

1 Introduction

Over the past two decades, intrusion detection research has shifted from signature-based solutions to machine learning (ML). ML-based techniques have significantly enhanced research in anomaly detection by providing algorithms that can automatically identify unusual patterns leading to a surge in the development of anomaly detection algorithms for NIDS in the literature.

Despite growing interest in evaluating IDS, few researchers have adapted classical evaluation techniques for ML-based IDS. In our previous work [1], we developed a comprehensive evaluation framework for ML-based IDS that includes data manipulation methods to assess various properties. Our prior work was theoretical, lacking practical implementation and a detailed pipeline for various properties. To address this, we now propose a demo of FREIDA, a practical Python implementation of our framework. FREIDA streamlines the evaluation of ML-based IDS models with a user-friendly interface and robust features for comprehensive assessments and provides a property-specific evaluation process. Additionally, the code of FREIDA and the evaluation results presented later are available on GitHub[1].

There is a notable tool in the literature that bears some resemblance to our work. In 2019, Zoppi et al. [7] introduced RELOAD, a framework designed to streamline the evaluation of anomaly detection algorithms. RELOAD simplifies this process through structured data preparation, enabling users to load data from various sources, apply feature selection techniques, choose algorithms for evaluation, and compute performance metrics. However, similar to many existing tools, RELOAD primarily focuses on assessing the effectiveness of IDS and lacks the capability for automated evaluation of other important IDS properties. In contrast, our tool goes further by incorporating data manipulation techniques to facilitate the assessment of these additional properties (effectiveness and robustness), enabling a more comprehensive evaluation of IDS.

The remainder of the paper is structured as follows: Sec. 2 delves into the evaluation methodology of the tool, followed in Sec. 3 by its validation through experimentation. Finally, we conclude in Sec. 4.

[1]https://gitlab.com/asolayman/freida

2 Methodology

In this section, we present our tool's evaluation methodology, demonstrating its use for comprehensive and reproducible evaluations. This tool implements the theoretical framework from [1], inspired by Milenkoski et al. [5], viewing IDS property evaluation as a measurement methodology that emphasizes selecting suitable datasets and metrics for effective assessment.

Now, we describe how to use the tool for state-of-the-art evaluations based on our framework, which details steps for creating suitable datasets for specific property assessments. The configuration interface (see Fig. 1) allows users to set these steps. To ensure easy reproducibility, the configuration can be exported to a file for others to import. Users simply toggle the switch and drag the configuration file into the designated area, it's also possible to selects a seed for the random number generator to maintain consistent results across different runs.

You can select one or more properties to assess, with two available: effectiveness and robustness. The effectiveness property includes two scenarios: *Baseline*, which evaluates standard performance using a basic dataset, and *Open-World*, which tests the model's ability to handle unknown attacks by removing samples from one attack class in the training set. For robustness, the tool offers three attack types to evaluate IDS performance against adversarial threats: FGSM [3] (a white-box attack), ZOO [2] (a black-box attack), and Data Poisoning Attack (where training data is compromised). Assessing the IDS against these diverse attack types provides a comprehensive understanding of its robustness.

Another selector lets you choose the models to evaluate. The web interface offers a limited list, but if you use the tool as a Python library it supports Scikit-learn[2] and Keras[3] models. If the robustness property is selected, you can choose attacks for evaluation using ART[4] to assess the model's performance against various adversarial threats. With the selected properties and models, you can import either an already split dataset or a single dataset to be split by the framework, selecting the split method (random or imbalanced) and specifying the split ratio.

Once the dataset is loaded, the tool displays a dataframe (see Fig. 2) to assist in filling out the next set of parameters. Four new parameters become available: the *Label Column*, which specifies the location of the labels; the *Multi-label Column*, identifying the column with multi-class labels for Open-World evaluations; the *Class to Remove*, which specifies which class to exclude in the Open-World scenario; and the *Drop Column*, which allows you to remove unnecessary columns, such as identifiers, before evaluation.

Our tool, building on the steps defined in [1], includes a Dataset Construction step with two configurable substeps: Dataset Preprocessing, which offers StandardScaler, MinMaxScaler, and RobustScaler methods, and Feature Selection, allowing for Lasso or PCA. The subsequent Dataset Evaluation and Dataset Refinement steps, although often overlooked, are crucial; our tool currently evaluates null values and uses SimpleImputer for handling missing data. While not yet comprehensive, we plan to enhance these steps to ensure higher data integrity and more reliable model evaluations.

[2]https://scikit-learn.org/stable/
[3]https://keras.io/
[4]https://github.com/Trusted-AI/adversarial-robustness-toolbox

Finally, once all fields are filled, click the *Run Evaluation* button to display the results in the corresponding section (see Fig. 3). You can download all evaluation artifacts, including the configuration file, JSON files with metrics, and heatmaps, using the *Download Evaluation Artifacts* button, enabling thorough analysis and sharing of your results.

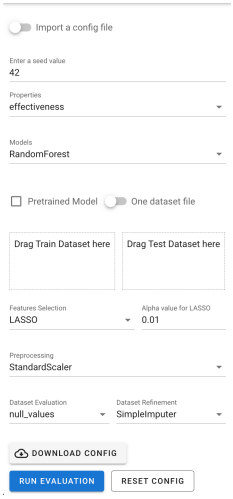


Figure 1: Configuration GUI

3 Experiments

In this section, we analyze and compare several IDS to evaluate their robustness and effectiveness using our tool. We tested various machine learning classifiers against different adversarial attacks created with the ZOO and FGSM algorithms, as well as a simple poisoning attack. These assessments followed the described methodology, utilizing 10% of the UNSW-NB15 [6] dataset with a multi-class configuration that is randomly selected but reproducible with a fixed seed. Our tool ensures that all results are reproducible.

The tool generates a heatmap for each property as output (see Fig. 4 for an example), showing the IDS' performance range on the baseline dataset. In this experiment, we assessed several classifiers for IDS: AdaBoost, Bagging, Decision Tree, Gradient Boosting, Logistic Regression, Random Forest, and SVC. These models, widely used in the literature, including [4], were implemented using the Scikit-learn library.

In the *Baseline* scenario, Random Forest achieved the highest performance, indicating strong overall effectiveness compared to others, while SVC and Gradient Boosting also performed well, with SVC showing the best discriminative ability. Bagging showed moderate performance, and the Decision Tree had the lowest accuracy due to overfitting. In the *Open-world* setting, where we choose to exclude one class during the training, and in our experiment, the — Exploits — class, Random Forest again demonstrated strong performance, with Bagging and Decision Tree performing relatively well. However, AdaBoost struggled without the Exploits class, and while Gradient Boosting and Logistic Regression had similar accuracies, Logistic Regression showed better discriminative ability in this scenario. Full results are available in the GitHub repo.

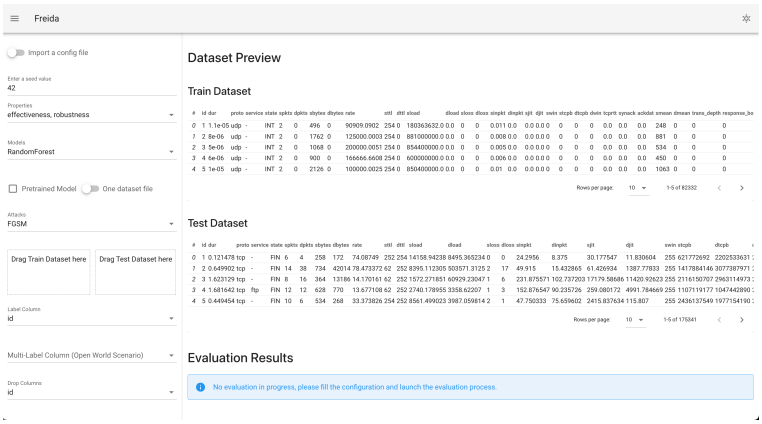
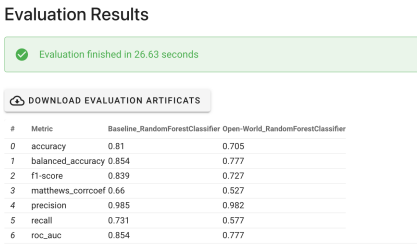


Figure 2: Complete GUI



#	Metric	Baseline_RandomForestClassifier	Open-World_RandomForestClassifier
0	accuracy	0.81	0.705
1	balanced_accuracy	0.854	0.777
2	f1-score	0.839	0.727
3	matthews_corrcoef	0.66	0.527
4	precision	0.985	0.982
5	recall	0.731	0.577
6	roc_auc	0.854	0.777

Figure 3: Evaluation results

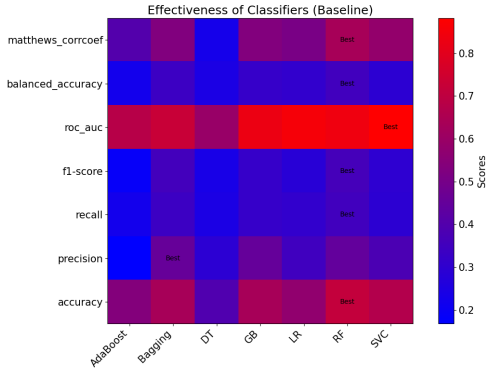


Figure 4: Evaluation results of the Baseline effectiveness

Under the FGSM adversarial attack, the SVC demonstrated the highest robustness, followed by Gradient Boosting and Random Forest, which showed moderate resilience. Bagging and Decision Tree exhibited high vulnerability. In contrast, under the ZOO attack, AdaBoost performed best, indicating its relative robustness, while Random Forest and Bagging showed decent performance. Logistic Regression was significantly susceptible, and SVC showed varied performance across attacks. Under the Poisoning scenario, all classifiers exhibited uniformly poor performance, reflecting the severe impact of poisoned data on their ability to distinguish between classes.

Overall, Random Forest and SVC were the most robust models across various scenarios, while Decision Tree and Logistic Regression showed greater sensitivity to data distribution changes and adversarial attacks.

4 Conclusion

In this paper, we presented a demo of FREIDA, a new tool addressing gaps in evaluation methodologies for ML-based IDS, often overlooked in terms of unbiased, transparent, and reproducible evaluations. We showcased FREIDA's web interface and its configuration process, demonstrating its applicability in evaluating IDS for detection effectiveness and robustness to adversarial attacks. Our Python implementation is a valuable resource for researchers, though features like explainability and data representation analysis from our original framework [1] are yet to be integrated. FREIDA's implementation opens new research avenues for evaluating and developing robust IDS solutions.

References

[1] Solayman Ayoubi, Gregory Blanc, Houda Jmila, Thomas Silverston, and Sébastien Tixeuil. 2022. Data-driven evaluation of intrusion detectors: a methodological framework. In *Foundations and Practice of Security*. Springer Nature Switzerland, 142–157. ISBN: 978-3-031-30122-3.

[2] Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. 2017. Zoo: zeroth order optimization based black-box attacks to deep neural networks without training substitute models. *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*.

[3] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *CoRR*, abs/1412.6572.

[4] Houda Jmila and Mohamed Ibn Khedher. 2022. Adversarial machine learning for network intrusion detection: a comparative study. *Computer Networks*, 214, 109073:1–109073:14.

[5] Aleksandar Milenkoski, Marco Vieira, Samuel Kounev, Alberto Avritzer, and Bryan D Payne. 2015. Evaluating computer intrusion detection systems: a survey of common practices. 48, 1, 1–41. Publisher: ACM New York, NY, USA.

[6] Nour Moustafa and Jill Slay. 2015. Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set). In *2015 Military Communications and Information Systems Conference (MilCIS)*, 1–6.

[7] Tommaso Zoppi, Andrea Ceccarelli, and Andrea Bondavalli. 2019. Evaluation of anomaly detection algorithms made easy with reload. In *2019 IEEE 30th International Symposium on Software Reliability Engineering (ISSRE)*, 446–455. DOI: 10.1109/ISSRE.2019.00051.