Rethinking Synthetic Oversampling for Intrusion Detection: When Similarity Hurts Performance

Maximilian Wolf[1], Dieter Landes[1], Andreas Hotho[2], and Daniel Schlör[2]

[1] Center for Responsible Artificial Intelligence, University of Applied Sciences and Arts Coburg, Coburg, Germany {maximilian.wolf,dieter.landes}@hs-coburg.de
[2] Center for Artificial Intelligence and Data Science, University of Würzburg, Würzburg, Germany
{hotho,schloer}@informatik.uni-wuerzburg.de

Abstract. Machine learning-based network intrusion detection systems (NIDS) require representative training data to detect evolving cyber-attacks. In order to identify specific attack types, multi-class models promise to be a suitable approach, but imbalanced training datasets make it difficult to train multi-class models to reliably detect specific cyber-attack types. Although prior work claims that resampling improves the classification performance, the literature is missing a comprehensive evaluation of classical and generative resampling methods, alongside a detailed assessment of synthetic data quality.
Our study fills that gap by assessing how different resampling strategies affect synthetic data quality and classifier performance. We evaluate 42 resampling combinations on seven classification models trained on five datasets and show differences in the quality of the synthetic data. Interestingly, synthetic data that is less similar to the original data often improve the classification results.

Keywords: Imbalanced learning · Intrusion detection · Attack classification · Generative model.

1 Introduction

Motivation The detection and prevention of a cybercrime is subject to constant adaption due to the continuous development of new threats. Based on the threat report of ENISA [6], company infrastructures are threatened by various forms of attacks, such as data theft, ransomware, and threats to system availability, including denial-of-service attacks and malware. Network intrusion detection systems (NIDS) analyze a company's network traffic to identify malicious activities. The application of machine learning methods for the analysis and extraction of network traffic patterns is a viable option, as highlighted in a comprehensive review of existing approaches by Salman et al. [27].

Multi-class classifiers can distinguish attack types like port scans or brute-force attacks, which allows a tailored response to the specific threat. However, multi-class models are often sensitive to imbalances in class distribution. The

infrequent nature of attack events results in unbalanced class distributions when collecting such data. Consequently, multiple ways of handling imbalanced class distributions have been proposed, e.g., classification algorithms specifically for imbalanced learning [14], weighting schemes [30], cost-sensitive learning [4] or resampling [8], which is particularly versatile as it is model-agnostic. Resampling techniques, specifically under- and oversampling, balance the class distribution by drawing samples from the majority classes to match the number of entries in the minority class or vice versa. To further augment the training data, synthetic resampling methods, such as SMOTE [2], balance the dataset by generating synthetic data. These common approaches might introduce issues in high-dimensional and complex datasets, such as boundary bias [7] and issues related to data-quality [10, 1] as a linear feature interpolation may not generate valid datapoints that accurately represent real-world scenarios, potentially leading to reduced classifier performance. In particular, some works claim that generative models outperform classical oversampling techniques in terms of augmentation performance [12, 21]. In contrast, [13] reports mixed results in a systematic evaluation of classical resampling techniques in the NetFlow domain and emphasizes that none of the tested techniques reliably improves classification performance and can even decrease classifier performance.

Research Gaps Despite the existence of resampling (mostly not comparable) benchmarks, the quality of synthetic data and its impact on classification improvement have not been thoroughly assessed, leaving open questions whether certain oversampling strategies produce higher-quality data that leads to greater improvements in classifier performance.

Approach To address these gaps, our objective is to systematically evaluate the effectiveness of modern generative models to improve class balance through data augmentation. Therefore, we compare modern generative models with classical (synthetic) oversampling techniques, both independently and in combination with undersampling methods (overall 42 combinations), which are tested on seven multi-class classifier models on 5 public NIDS datasets. Given this setup, we analyze the correlation between data quality and performance gains on the trained classification models. The overall setup is visualized in Fig. 1.

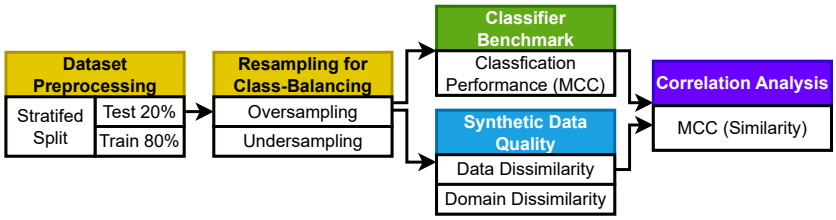

Fig. 1. Experimental Setup

2 Related Work

Various approaches for synthetic NetFlow generation and synthetic balancing can be found in literature.

Liu et al. [20] apply a conditional Variational Autoencoder for oversampling. In their experiments, they use the CIC-IDS-2018 dataset exclusively and incorporate two neural network models, a 1D-CNN and a GRU Network for attack classification. Their evaluation features a comparison of the classification performance of the models that are trained on the original data and the oversampled data. Although their work highlights the effect of class balancing in the NIDS domain, their results cannot be generalized since their findings rely on a single dataset and only two classification models. With respect to synthetic data quality, the work provides sample distributions per class after oversampling; but lacks a detailed analysis of the synthetic data itself.

Jiang et al. [17] use a diffusion-based model called NetDiffusion to generate packet-based network data in the pcap format. In their experiments, they use a self-collected dataset consisting of video streaming, video conferencing, and social media traffic. In their findings, they report that the diffusion model generates high-fidelity traffic with a high conformity to protocol specifications. Although their work focused on the generation of pcap traffic, their model is based on a diffusion model, which can be adapted for NetFlow data. We therefore adapt this model and include it in our comparison.

Another study [13] examines the effects of classical oversampling combined with undersampling, albeit exclusively on the UNSW-NB15 dataset and without the application of deep learning models for synthetic data generation. They report that there is no clear winning resampling combination that reliably improves classification performance. The quality of the synthetic data is assessed visually using t-SNE plots of NetFlows, allowing for a subjective comparison between real and synthetic datapoints.

Another comparison of resampling approaches [33] systematically evaluates classical oversampling as well as deep learning models of synthetic oversampling. Multiple classifiers are tested on three datasets, namely BoT-IoT, ToN-IoT and UNSW-NB15. The work demonstrates the unreliable nature of synthetic oversampling but does not elaborate on the influence of the synthetic data.

In most studies, the quality of synthetic data is evaluated indirectly through the augmentation task, except for the syntactical, tool-based evaluation by [17], which is exclusively packet-based, and the visual inspection using t-SNE by [13]. Prior work did not assess synthetic data quality from classical and deep learning resampling approaches, leaving unclear whether low-quality data correlates with performance decreases for oversampling. This raises questions about how data quality impacts multi-class model performance.

Moreover, our work includes additional experiments (more datasets and specialized NIDS classifiers like E-Graph-Sage) and evaluations have been included in our work, in which we explore the impact of synthetic data quality on oversampling performance, utilizing NetFlow-specific data quality measures for evaluation.

3 Experiments

Next, we describe the experimental setup of this work. The pseudocode in Algorithm 1 gives an overview of the experimental procedure. Following this, we introduce the datasets and describe the tested resampling approaches, classification models and metrics for evaluation.

Algorithm 1 Pseudocode of the experimental procedure

```
D ← {d_1, ..., d_5}                                              ▷ datasets
R ← {r_1, ..., r_42}                                             ▷ resampling strategies
C ← {c_1, ..., c_7}                                              ▷ classifiers
H^c ← {H_1^c, ..., H_n^c}                                        ▷ classifier hyperparameters
for all d in D do
    {d_train, d_eval} ← SplitTrainTest(d)
    for all r in R do
        RESULTS_c ← {}                                           ▷ classifier results
        d_r ← f_resample(r, d_train)                             ▷ resample train set
        for all c in C do
            for all h in H^c do
                classifier ← f_classifier(c, h, d_r)             ▷ Train Classifier
                predictions_eval ← classifier(d_eval)            ▷ predictions from test set
                metrics_eval ← ClassficationMetrics(d_eval, predictions_eval)
                Append(RESULTS_c, metrics_eval)
            end for
        end for
        if r is oversampling then
            quality_d ← EvalSynQuality(d_r)
            {ρ_p, ρ_k, ρ_s} ← CalcCorrelations(RESULTS_c, quality_d)   ▷ Pearson, Kendall,
Spearman
        end if
    end for
end for
```

Methods for Resampling An imbalanced data distribution refers to significant differences in the number of samples across classes, which can cause classifiers to exhibit poor discriminatory power, particularly concerning minority classes [15]. This section names the approaches for balancing the multi-class distribution of training datasets. The tested resampling approaches are separated into the categories oversampling (OS) and undersampling (US) and are listed in Table 1. In our experiments, we test the oversampling in combination with undersampling, resulting in 42 tested combinations (including cases where no oversampling or undersampling is applied). Some of the undersampling algorithms are specialized for cleaning the decision boundary between two classes, like NCR; therefore, we test the oversampling alongside the cleaning algorithms to include their potential for improvement of the tested classification models.

Benchmark Datasets The NF-BoT-IoT, NF-ToN-IoT and NF-UNSW-NB15 datasets applied in our experiments are originally based on packet-based datasets, which have been transformed via nProbe [24] by [28] to ensure cross dataset comparability. The NF-CIDDS-001 and NF-CIDDS-002 datasets are transformed from unidirectional NetFlow format into bidirectional format for compatibility. All datasets exhibit a heavily imbalanced class distribution, especially in the multi-class setting. In the following, a brief description of each

Table 1. Listing of the resampling approaches in our experiments

Category	Algorithm	Reference
OS	Random Oversampling (ROS)	[15]
OS	Synthetic Minority Over-sampling TEchnique (SMOTE)	[2]
OS	Adaptive synthetic oversampling (ADASYN)	[16]
OS	Conditional-Variational-Autoencoder (C-VAE)	[29]
OS	Conditional-WGAN-GP (C-WGAN)	[5]
OS	Conditional Denoising Diffusion Probabilistic Model (C-DDPM)	[26]
US	Random Undersampling (RUS)	[15]
US	Tomek Links (Tomek)	[31]
US	Edited Nearest Neighbor (ENN)	[32]
US	Neighborhood Cleaning Rule (NCR)	[18]
US	Near Miss 3 (NM-3)	[23]

dataset is given. The NetFlows per class in each dataset are listed in Table 2. We split each NIDS dataset into stratified train-test sets, with the training set comprising 80% of the data and the test set containing the remaining 20%. Given the extreme imbalance in the datasets, we generate stratified splits to ensure that each class is proportionally represented in both the training and test sets, maintaining the original class distributions. Following the publishers of the NIDS datasets [28], we exclude IP addresses and ports, as they are strong indicators of attacks, given that certain attacks are often executed from similar IPs. Moreover, we exclude the L7_PROTO field which contains Layer 7 protocol information which is sometimes unique for normal or attack behavior thus constituting an indicator for the class label. Furthermore, we normalize NetFlow attributes using min-max normalization based on the value ranges of each field of the NetFlow V9 specification [24], before applying resampling methods and classification models. Resampling methods are applied on the train set exclusively.

Table 2. Listing of the NetFlows per class in each dataset

NF-Bot-IoT	Benign	Theft	DDoS	DoS	Reco.	—	—	—	—	—
	1 108 995	1 909	56 844	56 833	470 655	—	—	—	—	—
NF-ToN-IoT	Benign	DoS	Injection	DDos	Scanning	Password	MitM	XSS	Backdoor	Ransom.
	1 108 995	17 717	468 539	326 345	21 467	156 299	1 295	99 944	17 247	142
NF-UNSW-NB15	Benign	Exploits	Reco.	DoS	Generic	Shellcode	Backdoor	Fuzzers	Worms	Analysis
	1 550 712	24 736	12 291	5 051	5 570	1 365	1 782	19 463	153	1 995
NF-CIDDS-001	Benign	PortScan	DoS	PingScan	BruteForce	—	—	—	—	—
	1 921 568	47 675	267 369	1 693	1 047	—	—	—	—	—
NF-CIDDS-002	Benign	UDP-Scan	FIN-Scan	Ping-Scan	ACK-Scan	SYN-Scan	—	—	—	—
	1 713 056	23 936	51 451	288	36 166	35 572	—	—	—	—

Classification Models Our setup comprises several multi-class classification models chosen for NetFlow classification based on current literature. To be precise, we incorporate a broad range of popular and well-known models, specif-

ically K-Nearest Neighbors (kNN) [15], Decision Tree (DTree) [15], Random Forest (RF) [15], Extra Tree (ExTree) [9], Multi-Layer Perceptron (MLP) [15], Extreme Gradient Boosting (XGBoost) [3] and E-GraphSAGE (E-GSAGE) [22]. We evaluate the multiclass classification performance via the established multiclass metric Matthews Correlation Coefficient (MCC) [11].

Synthetic Data Evaluation requires specific metrics to measure fundamental characteristics of the synthetic data. The quality of generated NetFlow data can be measured in terms of similarity as proposed by [34], where various measures are aggregated into a so-called Data Dissimilarity Score and Domain Dissimilarity Score, visualized in Fig. 2. The Data Dissimilarity Score utilizes distributional distances or correlations within the data, as well as unsupervised models such as Isolation Forest (IF) and One-Class-SVM (OCSVM) to detect anomalies in (synthetic) data. The Domain Dissimilarity Score includes NetFlow specific measures like classification tasks and syntax checks, designed for the NetFlow domain. The binary classification (normal or attack) is tested to evaluate the quality of data and its labels. In our experiments, we used these measures to assess the data quality of the synthetic data generated by the oversampling strategies.

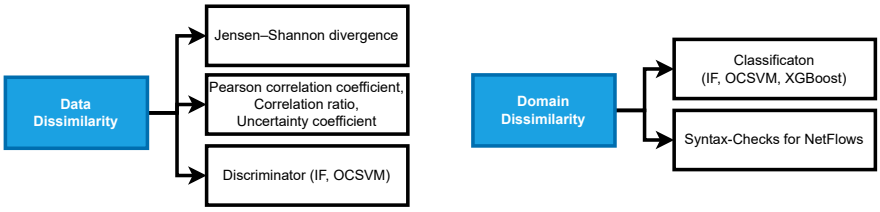

Fig. 2. Individual metrics of the Data and Domain Dissimilarity

4 Computational Complexity

The tested oversampling techniques consist of two major categories: classical oversampling approaches and deep learning based techniques. Classical oversampling methods such as ROS, SMOTE, and ADASYN do not require model training. ROS operates in linear time, while SMOTE and ADASYN rely on nearest-neighbor searches whose complexity is quadratic in the naive case and $O(N \log N)$ with spatial indices (e.g via Ball Trees), but still suffers from the curse of dimensionality. In contrast, deep generative methods (e.g., C-VAE, C-WGAN-GP, C-DDPM) require upfront training cost dominated by neural network size, number of layers, and training epochs, which can be approximated as $O(E \cdot L \cdot N)$. C-DDPM is the most computationally expensive due to its iterative de-noising process during both training and sampling. After training, sampling from deep generative models is relatively efficient, though C-DDPM again incurs additional cost proportional to the number of de-noising steps.

5 Improvement on Classification Performance

Overall, we tested 210 configurations for each oversampling approach. The configurations result from combinations of six undersampling methods with seven classification models applied on five datasets. Our experiments reveal that some models show small improvements in classification performance when trained on resampled data. However, most of the results indicate lower performance compared to training on the original imbalanced training data across all tested classifier models. The bar charts in Fig. 3 show the relative improvement in classification performance for each model trained on a specific resampling combination (sorted by oversampling approaches on the y-axis), calculated by subtracting the baseline MCC values (no resampling applied) from each resampling combination score. The numbered bars represent the number of classifiers for each category. We categorize the results into five intervals based on the MCC difference between the baseline and the tested resampling method, where the difference is defined as MCC-baseline - MCC-resampling. The five colored categories are:

- Red: negative impact (MCC difference $<$ –0.1)
- Orange: slight negative impact (–0.1 $\geq$ MCC difference $<$ -0.01)
- Yellow: negligible impact (-0.01 $\geq$ MCC difference $\leq$ 0.01)
- Light green: slight positive impact (0.01 $>$ MCC difference $\leq$ 0.1)
- Green: positive impact (difference $>$ 0.1)

The bar chart demonstrates a minimal positive impact on the classification performance in general. There is no consistent improvement trend where a single resampling strategy consistently improves the results of a model across all datasets. The deep learning models (C-DDPM, CVAE and C-WGAN) seem to deliver more strongly improved classifiers (23, 24, 22) than the baseline variants (ADASYN, ROS and SMOTE), which decrease the performance in most cases. Notably, resampling has very little impact on the classification performance (yellow bar) in several cases.

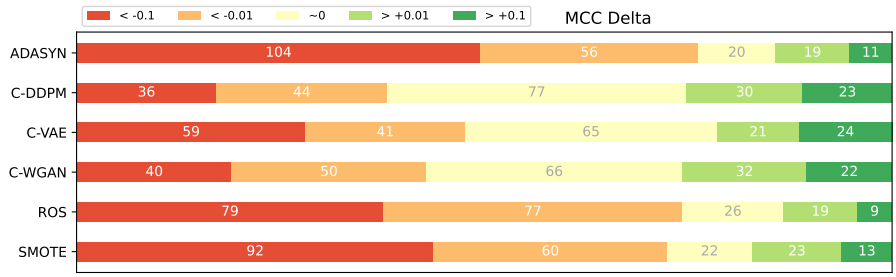

Fig. 3. Counts of resampled models (x-axis) for each category of delta MCC values per oversampling, denoting the improvement over the original data without resampling.

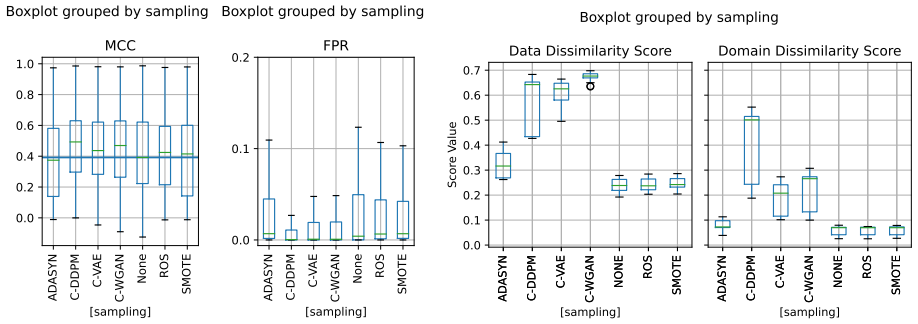

Fig. 4. The bar charts show the mean MCC and Dissimilarity values per resampling method, for all datasets.

6 Correlation of Data Quality and Oversampling

In this section, we analyze the influence of the dissimilarity of synthetic data on the oversampling performance gain. To be precise, we want to investigate the relation between the Dissimilarity metrics of the synthetic data and the MCC-delta, i.e. the gain or loss in classification performance, obtained by the various classification models. The boxplots in Fig. 4 visualize the distributions of MCC, FPR and Dissimilarity Scores per resampling method. The plots show that Deep Learning models like C-VAE, C-WGAN and C-DDPM feature a higher median MCC, while maintaining a low FPR and a larger dissimilarity in their synthetic data, which seems to improve the MCC slightly on average. In contrast, methods that feature an identical dissimilarity to the original data (called *None* in the figure), such as SMOTE and ROS, have a negligible influence on the MCC score and FPR and no improvement on the classification performance.

The analysis of the influence of resampling on classification performance shows that different classification models respond differently to resampling strategies. This observation motivates an analysis of the correlation between oversampling performance and synthetic data quality for each classification model.

The bar charts in Fig. 5 depict the relationship between the MCC delta and the Dissimilarity Score in terms of Pearson, Kendall and Spearman correlations. In accordance with previous findings, classifiers respond differently to the quality of synthetic data and the underlying dataset. In summary, 22 out of 35 setups exhibit a positive correlation between Dissimilarity and oversampling performance. This suggests that synthetic oversampling data may benefit from being (slightly) dissimilar to the original data to improve a model's classification performance. However, our experiments indicate that this effect is classifier- and dataset-dependent. Notably, ExTree is the only model that consistently demonstrates a positive correlation across all datasets, whereas the other models show varying correlations depending on the dataset.

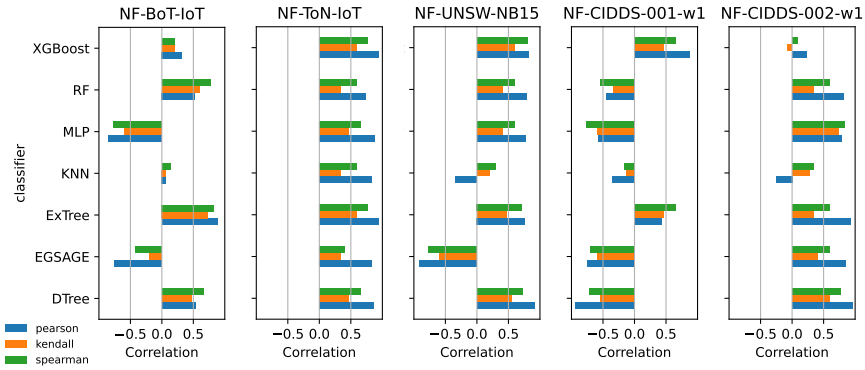

Fig. 5. The bar charts show the correlation (pearson, kendall and spearman) of Dissimilarity and MCC delta per model

7 Conclusion

This work evaluated various classical and deep learning approaches for synthetic data generation. Our experiments reveal different levels of quality based on the generative approaches. We observe that most classical approaches generate data closely resembling the original distribution, whereas deep learning models tend to produce data with greater distributional differences. Combined the results of resampling performance and data quality to assess the impact of data quality on the effectiveness of oversampling does not reveal a consistent trend across all datasets, making it difficult to establish a clear correlation between data quality and oversampling performance. However, classification performance generally tends to improve with synthetic data that is more dissimilar to real data. This finding supports the intuition that data used for augmentation should provide new information. Augmenting with (almost) identical data fails to introduce additional insights, limiting the model's ability to learn a broader range of information. Similar effects are observed in other areas of data augmentation, such as training with adversarial data [19]. One limitation of our work is that special forms of sampling bias, namely temporal bias [25], are outside the scope of our experimental setup.

Bibliography

[1] Boudegzdame, N., Sedki, K., Tspora, R., Lamy, J.B.: An approach for improving oversampling by filtering out unrealistic synthetic data. In: ICAART (3). pp. 291–298 (2024)

[2] Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: synthetic minority over-sampling technique. Journal of artificial intelligence research **16**, 321–357 (2002)

[3] Chen, T., Guestrin, C.: Xgboost: A scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. p. 785–794. ACM, San Francisco California USA (2016). https://doi.org/10.1145/2939672.2939785

[4] Elkan, C.: The foundations of cost-sensitive learning. In: International joint conference on artificial intelligence. vol. 17, pp. 973–978. Lawrence Erlbaum Associates Ltd (2001)

[5] Engelmann, J., Lessmann, S.: Conditional wasserstein gan-based oversampling of tabular data for imbalanced learning. Expert Systems with Applications **174**, 114582 (2021). https://doi.org/10.1016/j.eswa.2021.114582

[6] ENISA: European union agency for cybersecurity threat landscape 2025 (Dec 2025), https://www.enisa.europa.eu/publications/enisa-threat-landscape-2025

[7] Fernández, A., Garcia, S., Herrera, F., Chawla, N.V.: Smote for learning from imbalanced data: progress and challenges, marking the 15-year anniversary. Journal of artificial intelligence research **61**, 863–905 (2018)

[8] Fernández, A., López, V., Galar, M., Del Jesus, M.J., Herrera, F.: Analysing the classification of imbalanced data-sets with multiple classes: Binarization techniques and ad-hoc approaches. Knowledge-Based Systems **42**, 97–110 (2013). https://doi.org/10.1016/j.knosys.2013.01.018

[9] Geurts, P., Ernst, D., Wehenkel, L.: Extremely randomized trees. Machine Learning **63**(1), 3–42 (2006). https://doi.org/10.1007/s10994-006-6226-1

[10] Gholampour, S.: Impact of nature of medical data on machine and deep learning for imbalanced datasets: Clinical validity of smote is questionable. Machine Learning and Knowledge Extraction **6**(2), 827–841 (2024)

[11] Grandini, M., Bagli, E., Visani, G.: Metrics for multi-class classification: an overview (2020), https://arxiv.org/abs/2008.05756

[12] Guo, Y., Xiong, G., Li, Z., Shi, J., Cui, M., Gou, G.: Combating imbalance in network traffic classification using gan based oversampling. 2021 IFIP Networking Conference (IFIP Networking) pp. 1–9 (2021)

[13] Gutiérrez, Ó.M., Núñez, J.C.S., Ávila, M., Caro, A.: A detailed study of resampling algorithms for cyberattack classification in engineering applications. PeerJ Computer Science **10**, e1975 (2024). https://doi.org/10.7717/peerj-cs.1975

[14] Haixiang, G., Yijing, L., Shang, J., Mingyun, G., Yuanyue, H., Bing, G.: Learning from class-imbalanced data: Review of methods and applications. Expert systems with applications **73**, 220–239 (2017)
[15] Han, J., Kamber, M., Pei, J.: Data mining: concepts and techniques. Elsevier/Morgan Kaufmann, Amsterdam, 3rd ed edn. (2012)
[16] He, H., Bai, Y., Garcia, E.A., Li, S.: Adasyn: Adaptive synthetic sampling approach for imbalanced learning. In: 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence). p. 1322–1328. IEEE, Hong Kong, China (Jun 2008). https://doi.org/10.1109/IJCNN.2008.4633969, http://ieeexplore.ieee.org/document/4633969/
[17] Jiang, X., Liu, S., Gember-Jacobson, A., Bhagoji, A.N., Schmitt, P., Bronzino, F., Feamster, N.: Netdiffusion: Network data augmentation through protocol-constrained traffic generation. Proceedings of the ACM on Measurement and Analysis of Computing Systems **8**(1), 1–32 (2024). https://doi.org/10.1145/3639037
[18] Laurikkala, J.: Improving identification of difficult small classes by balancing class distribution. In: Quaglini, S., Barahona, P., Andreassen, S. (eds.) Artificial Intelligence in Medicine. p. 63–66. Springer Berlin Heidelberg, Berlin, Heidelberg (2001)
[19] Lella, E., Macchiarulo, N., Pazienza, A., Lofù, D., Abbatecola, A., Noviello, P.: Improving the robustness of dnns-based network intrusion detection systems through adversarial training. In: 2023 8th International Conference on Smart and Sustainable Technologies (SpliTech). p. 1–6. IEEE, Split/Bol, Croatia (Jun 2023). https://doi.org/10.23919/SpliTech58164.2023.10193009, https://ieeexplore.ieee.org/document/10193009/
[20] Liu, C., Antypenko, R., Sushko, I., Zakharchenko, O.: Intrusion detection system after data augmentation schemes based on the vae and cvae. IEEE Transactions on Reliability **71**(2), 1000–1010 (2022). https://doi.org/10.1109/TR.2022.3164877
[21] Liu, X., Li, T., Zhang, R., Wu, D., Liu, Y., Yang, Z.: A gan and feature selection-based oversampling technique for intrusion detection. Secur. Commun. Networks **2021**, 9947059:1–9947059:15 (2021)
[22] Lo, W.W., Layeghy, S., Sarhan, M., Gallagher, M., Portmann, M.: E-graphsage: A graph neural network based intrusion detection system for iot. In: NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium. p. 1–9. IEEE (2022)
[23] Mani, I., Zhang, I.: knn approach to unbalanced data distributions: a case study involving information extraction. In: Proceedings of workshop on learning from imbalanced datasets. vol. 126, p. 1–7. ICML (2003)
[24] Ntop: nprobe 10.1 documentation (2023), https://www.ntop.org/guides/nprobe/index.html
[25] Pendlebury, F., Pierazzi, F., Jordaney, R., Kinder, J., Cavallaro, L.: Tesseract: Eliminating experimental bias in malware classification across space and time. In: USENIX Security Symposium (2018)

[26] Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10684–10695 (2022)

[27] Salman, O., Elhajj, I.H., Kayssi, A., Chehab, A.: A review on machine learning–based approaches for internet traffic classification. Annals of Telecommunications **75**(11–12), 673–710 (2020). https://doi.org/10.1007/s12243-020-00770-7

[28] Sarhan, M., Layeghy, S., Moustafa, N., Portmann, M.: Netflow datasets for machine learning-based network intrusion detection systems. In: Big Data Technologies and Applications: 10th EAI International Conference, BDTA 2020, and 13th EAI International Conference on Wireless Internet, WiCON 2020, Virtual Event, December 11, 2020, Proceedings 10. pp. 117–135. Springer (2021)

[29] Sohn, K., Lee, H., Yan, X.: Learning structured output representation using deep conditional generative models. Advances in neural information processing systems **28** (2015)

[30] Steininger, M., Kobs, K., Davidson, P., Krause, A., Hotho, A.: Density-based weighting for imbalanced regression. Machine Learning **110**, 2187–2211 (2021)

[31] Tomek, I.: Two modifications of cnn. IEEE Transactions on Systems, Man, and Cybernetics **SMC-6**(11), 769–772 (Nov 1976). https://doi.org/10.1109/TSMC.1976.4309452

[32] Wilson, D.L.: Asymptotic properties of nearest neighbor rules using edited data. IEEE Transactions on Systems, Man, and Cybernetics **SMC-2**(3), 408–421 (1972). https://doi.org/10.1109/TSMC.1972.4309137

[33] Wolf, M., Landes, D., Hotho, A., Schlör, D.: Systematic evaluation of synthetic data augmentation for multi-class netflow traffic. 6th Workshop on Machine Learning for CyberSecurity at Machine Learning and Knowledge Discovery in Databases: ECML PKDD 2024, Vilnius, Lithuania September 09–13, 2024 **to appear, preprint arXiv:2408.16034** (2024)

[34] Wolf, M., Tritscher, J., Landes, D., Hotho, A., Schlör, D.: Benchmarking of synthetic network data: Reviewing challenges and approaches. Computers and Security p. 103993 (2024). https://doi.org/https://doi.org/10.1016/j.cose.2024.103993