International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 13 Issue: 10 | Oct 2026 www.irjet.net p-ISSN: 2395-0072

Machine Learning in Intrusion Detection: A Cross-Dataset Generalization Study

[1]Ansh Pratap Singh, [2]Ayush Singh, [3]Dr. Nidhi Saxena

[123]School of Computer Applications (Data Science & Artificial Intelligence), Babu Banarasi Das University, Lucknow, India

---

Abstract - *Machine learning based Intrusion Detection Systems (ML-IDS) show promising results when trained and tested on the same dataset, however perform poorly on different networks or datasets. This paper proposes a practical research driven framework to understand why Intrusion Detection (IDS) systems do not generalize well across different datasets. We highlight new insight into intrusion detection problems by introducing IDS features (variables extracted from flows used to train the classifier) and the contextual information of the dataset which can affect how the model generalizes from its prior knowledge. The 18 factors that affect generalization are separated into three categories; dataset/attack, data/feature and context/realism. We provide the protocol required to perform a controlled cross dataset study. Definitions for evaluating generalization, metrics and implementation independent pseudocode are also provided. In order to demonstrate the motivation for this work we use data provided by Cantone et al. [1] which shows an average intra-dataset Matthews Correlation Coefficient (MCC) of 94.63% compared to an inter-dataset MCC of 29.35% (a difference of 65.28%). These results were not collected from our experiments. This work does not contain any original data. Our contributions consist of empirical evaluations that will be performed.*

Keywords - Intrusion, Detection, Machine Learning, Cross Dataset Generalization, Dataset Quality, Feature Distribution Shift, Attack Diversity, Class Imbalance.

I. INTRODUCTION

Intrusion detection systems (IDS) monitor data flows to detect any malicious activity. Machine learning (ML) is commonly used to automate this process [2]. In a supervised setting, given a labelled dataset a classifier can be trained to learn the relationship between flow-based features and whether an instance is benign or an attack. However, creating and labelling realistic attacks is expensive, meaning most research relies upon a small set of readily available benchmark datasets such as NSL-KDD, UNSW-NB15 and CIC-IDS2017 [3]. Surveys show that these datasets suffer from issues including class imbalance, old attack scenarios, low attack variety, labelling problems and low realism [3]-[5].

A model which has been tested on a split of the training data it was trained on will be in the best scenario possible. The training and test examples will have come from the same network, using the same attack tools, settings, feature extractor and labelling. However this does not give any insight into how well the model can generalise if any of these factors change [1]. Cantone et al. trained ML-IDS classifiers on one dataset and then tested them on another. They found that the classifiers performed very well when tested on the same dataset with results close to perfect, however they performed poorly on many cases when tested on a different dataset with results close to chance [1]. There are many factors in the literature which cause this issue. These are often referred to in isolation, for example dataset make-up, attack generation, labelling quality, feature definitions, environment and age of data [1], [3]-[5]. There is no overarching framework to address these factors together and link them to a quantifiable generalization gap.

This paper provides a theoretical solution to this problem. Contributions include: (i) Separating the generalization factors of an IDS models input features from those of the dataset. (ii) An 18-factor taxonomy split into 3 categories to consider attack, sample, data, feature, environment and age factors. (iii) A methodology for evaluating models in a controlled manner across datasets considering these factors. A generalization gap, metrics and pseudocode are presented. (iv) A structured way of interpreting evidence of degradation from the literature. (v) A plan to validate the proposed framework in future empirical work. No new experiments were conducted and all numerical results are taken from the literature.

II. LITERATURE REVIEW AND RESEARCH GAP

A. Cross-dataset performance analysis

Cantone et al used four different classifiers (LDA, decision tree, random forest and XGBoost) to train on four different flow data sets (CIC-IDS2017, CSE-CIC-IDS2018, LycoS-IDS2017, LycoS-Unicas-IDS2018) [1]. It was noticed that the CIC data sets [6] and LycoS data sets [7], had similar flow features, however this data set had been re-extracted and relabeled due to issues with the initial extraction [7]. When training on one data set and testing on a different data set it was observed that the average MCC dropped significantly by 65.28pp, to just 29.35% compared to the within data set analysis which achieved an average MCC of 94.63% [1]. Of the sixteen within data sets nine had an

MCC greater than 99.6%. This is unusual as it indicates that there was little variability and many samples were very similar [1]. The between data set results also depend on the direction. The same attacks can be very different between the two data sets with greatly varying feature distributions. This can be attributed to difference in tools/configurations when creating the attack [1].

TABLE I FINDINGS FROM CANTONE ET AL. [1]

Quantity	Value
Average within-dataset MCC	94.63%
Average cross-dataset MCC	29.35%
Difference (derived)	65.28 pp
Lowest / highest average cross-dataset combination (LycoS17→LycoS18 / CIC18→CIC17)	≈10.83% / ≈40.24%
Feature-selection experiments: within / cross MCC	83.86% / 14.91%
Within-dataset experiments with MCC > 99.6%	9 of 16

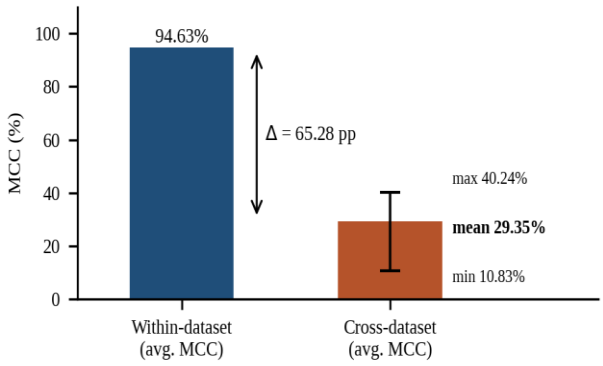


Fig. 1. Average MCC for within- and cross-dataset evaluation from Cantone et al. [1]; the whisker spans the lowest and highest average train–test combinations. Literature values, not results of this paper.

B. Dataset Quality and Related Studies

A survey of 89 publicly available IDS datasets has shown limitations on data authenticity, annotation, reproducibility and suitability for tasks [4]. A structured literature review suggests the development of a framework for selecting datasets based on environment, attack variety, annotation quality, feature representation, class balance, authenticity, time sensitivity, documentation and reproducibility [5]. Systematic studies highlight that class imbalance, data limitations and evaluation are amongst the challenges facing ML-IDS [3]. Integrity studies have shown that: Rosay et al. identified duplicate features, miscalculations and protocol detection errors within the CIC-IDS2017 dataset extraction [7], Engelen et al. identified that over 20% of flows needed to be rebased and annotated [8], Liu et al. estimated 6.67% and 7.53% mislabeling within the CIC-IDS2017 and CSE-CIC-IDS2018 datasets [9] respectively, all of which have been summarized in [1]. Lanvin et al. also highlighted duplicate and incorrectly classified attacks [10]. Catillo et al. indicated that the results achieved on public intrusion datasets lack practicality [11]. Cross-dataset studies have also shown similar issues. D'Hooge et al. concluded that classifiers cannot operate correctly on data from unknown networks [12]. Verkerken et al. observed an average loss of 30.45% on the AUROC [13]. Federated learning has been suggested however the models in the studies shown in [1] all had access to all data within all sources therefore do not address transferring to an unknown dataset.

C. Gap in the Literature

Whilst it is clear from the literature that there are issues with degradation and dataset characteristics, there is no connection between them. Whilst attack diversity, sample diversity, data redundancy, feature skewness, context and authenticity are mentioned throughout [1], [3]-[5], there is no mention of a quantifiable gap within one framework. The key gap is a combined analysis of factors.

III. PROBLEM FORMULATION

Let M(A) be a model trained on dataset A and MCC(A→B) be the MCC of the model on dataset B. The generalization gap between the ordered pair (A,B) can be calculated as:

$$\Delta(A, B) = MCC(A \rightarrow A) - MCC(A \rightarrow B) \quad (1)$$

The relative gap can be calculated as Δ/MCC(A→A). This was shown to be an average of Δ = 94.63 – 29.35 = 65.28 pp [1]. A small gap doesn't necessarily mean a good model as both values may be low. The main hypothesis being that degradation is not due to one reason but correlated with differences in attack, sample, data, feature, context and time factors. Three hypotheses were made to test this, H1- a greater mismatch between factors in the source and target will result in a larger Δ. H2- Controlling for data redundancy and labelling noise will affect the intra-score more than inter-score. H3- Aligning features will reduce but not remove the generalization gap. None of the hypotheses were tested.

IV. PROPOSED FRAMEWORK

A. IDS Features and Generalization Factors

IDS features are quantitative values within the flow that the classifier uses. These include statistical packet lengths, flow duration, IAT, packet and byte rate, ports, TCP flags, window sizes and forward/backward packet stats. After removing any identifying values, timestamps and redundant information, 77 features were used in [1].

Generalization factors relate to the dataset, not necessarily the model. They describe the data and how it was collected. This determines if patterns learnt from the features will be transferred. For example, Flow IAT Mean is a feature but attack diversity will cause the distribution to change between different datasets. Feature definitions depend on the flow extractor [7].

B. Taxonomy of Eighteen Factors

Eighteen factors have been identified across three groups (Table II, Fig. 2). These were identified from the dataset quality and selection criteria [4], [5], and degradation mechanisms [1]. Equal numbers of factors across groups have been selected as this was just based on number of defined factors.

TABLE II EIGHTEEN GENERALIZATION FACTORS AND INDICATORS

Group	Factors (measurable indicator)
Dataset & attack	Diversity of attacks (types, variants). Coverage (proportion of attacked classes covered by the attack). Attack generation diversity (tools, parameters used). Sample diversity (proportion of unique records). Sample duplication (proportion of duplicates). Temporal relevance (collection time vs. deployment time).
Data & feature	Label quality: verification method, error-rate Label consistency: labelling rules are consistent across different data sets Class balance: imbalance ratio Feature representation: flow (packet/payload/raw) Feature compatibility: percentage of shared definitions. Feature distribution shift: distance between distributions.
Context & realism	Environmental factors (topology, services, hardware, capture point), source compatibility (telemetry type, extractor), dataset authenticity (closeness to real data), representativeness (coverage of target scenarios), documentation (notes on dataset creation), replicability (dataset versions, access, scripts) [4], [5].

These factors are not independent. Differences in how attacks are generated will affect the distribution of some features (mean packet size). With increased redundant samples the number of unique examples that the model can train on reduces. This means that the model can score almost perfectly on very few unique behaviors. The reliability of label assignment depends on the environment and documentation. A labelling script trained on an attack timetable in a dataset may mislabel flows from a different capture scenario [7]. While each factor can be individually scored, they must be jointly considered to make sense of the results.

C. Framework Architecture

In general, the workflow follows the steps shown in Fig. 2. For a given target IDS context (threat model, environment, candidate datasets), the three different factor groups are recorded for all candidate datasets. This is followed by the application of the profiles to a controlled evaluation protocol. Metrics are calculated along with the gap and related to the factor profiles. This provides guidance on how to select datasets and evaluate them. By recording factors before evaluation, the degradation observed can be interpreted.

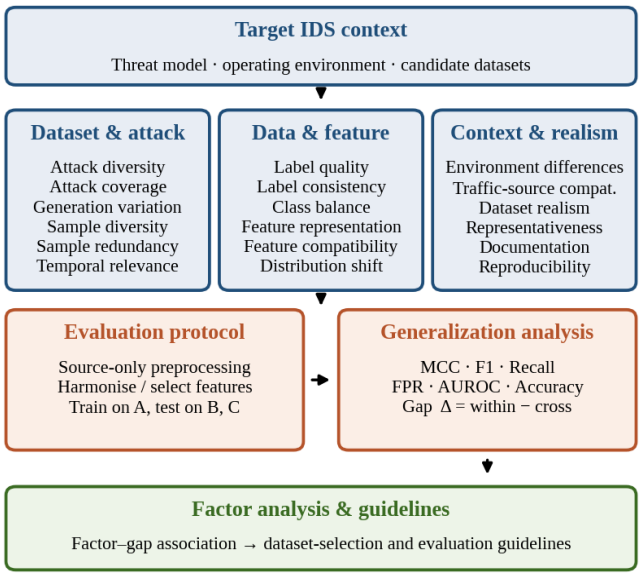


Fig. 2. Proposed factor-based framework for cross-dataset generalization analysis.

V. RESEARCH METHODOLOGY

A. Factor Extraction and Dataset Assessment

The factors in Table II were identified through structured narrative analysis of [1]-[5], and the integrity studies [7]-[10]. All documents were assessed to identify information about degradation mechanisms, quality criteria and selection criteria. Repeating points were then brought together to form Table II. As this is a proposal, the list could be improved. Each potential dataset is assessed against the indicators in Table II and basic analysis undertaken where possible. Basic analysis includes calculating duplicate rate, class proportions and distance of distribution.

B. Controlled Cross-Dataset Protocol and Leakage Control

The protocol was adapted from [1], with the addition of factor tracking. The data was preprocessed using (1) source only model training to determine the preprocessing parameters (imputation, scaling, encoding), (2) harmonizing features with consistent definitions, and dropping features that could cause leakage or contain identifiers (flow identifiers, addresses and time stamps) [1]. (3) Feature selection was performed using a predetermined method, again only on the source. (4) Hyperparameter tuning was then performed using a subset of the source data. Lastly, the model was evaluated using a holdout set within the source, and the unseen target data with no retraining, fixed preprocessing parameters. Cross dataset evaluations were only performed on classes that were present in both datasets. The coverage factor tracks what was not included. Each pair was evaluated in both directions as the transfer is not symmetric [1]. To prevent leakage the data was de-duplicated prior to splitting. Random seeds were fixed, and any stochastic processes were repeated e.g. under sampling [1]. The version of the dataset, list of features, preprocessing code and range of hyperparameters should all be published to allow for replication.

C. Evaluation Metrics

Let TP, TN, FP and FN be confusion-matrix counts with the attack class positive. Accuracy is reported but not relied on, because benign traffic dominates: in [1], an always-benign model reaches 99.5% accuracy for attacks with a benign-to-attack ratio above 200. Precision, recall, F1, the false-positive rate FPR = FP/(FP+TN) and AUROC are also reported. The primary metric is the Matthews correlation coefficient,

$$MCC = TP \cdot TN - FP \cdot FN$$

$$\sqrt{[(TP+FP)(TP+FN)(TN+FP)(TN+FN)]} \quad (2)$$

which is considered more informative than accuracy and F1 for imbalanced binary classification [14] and is the primary metric of [1]. Decision thresholds are set on the source only and applied unchanged to the target, so that reporting AUROC together with fixed-threshold metrics (MCC, FPR) reveals cases where the ranking of flows transfers but the calibrated operating point does not.

D. Factor-Wise Analysis and Algorithm

Factor profiles were assigned scores in the range 0-3, based on documentary data, for each dataset i. For an ordered pair of datasets (i,j) the discrepancy between factor profiles was calculated as $d(i,j)=\Sigma k|fik-fjk|$, where this can be done within groups. The relationship between d and Δ and the individual factors was investigated via rank correlation. If there were sufficient pairs of data, it was possible to analyze the relationship using regression with group level variables. Factor scores were not summed into a single index as we do not know the relative importance of the factors and do not want to suggest more accuracy than we have. With a small number of dataset pairs, relationships are identified in an exploratory manner, using confidence intervals. Algorithm 1 shows the process, which does not depend on any implementations. The target variable is only transformed, never fitted meaning no information about the target can be passed to the model.

Algorithm 1. Factor-aware cross-dataset evaluation
for each source dataset D_i **do**
  Learn preprocessing parameters θ on D_i only
  $X_i \leftarrow$ Harmonize (Transform (D_i, θ))
  S ← Select Features (X_i; predefined protocol)
  m ← Train (M, X_i[S]); tune on source subset
  Evaluate m on held-out $D_i \rightarrow E_{ii}$
  for each target D_j, $j \neq i$ **do**
    $X_j \leftarrow$ Harmonize (Transform(D_j, θ)); no refitting
    Compute Acc., Prec., Rec., F1, FPR, AUROC, MCC $\rightarrow E_{ij}$
    $\Delta_{ij} \leftarrow MCC(E_{ii}) - MCC(E_{ij})$; $d_{ij} \leftarrow$ Mismatch (f_i, f_j)
    Store (i, j, Δ_{ij}, d_{ij}, E_{ij})
  end for
end for
Relate Δ to d (overall, per factor group); repeat with controlled variants
Rank factors by consistency of association with Δ
return generalization profile, factor ranking, guidelines

VI. CROSS DATASET FINDINGS AND ANALYSIS

A. Interpretation of the Findings

Table III cross-references observations in [1] to framework factors. The associations are interpretive in nature since the factors were not measured in the original study and are inferred to exist by the work's authors or are consistent with the dataset-quality literature [4], [5].

TABLE III OBSERVATIONS FROM [1] MAPPED TO FACTORS

Reported observation	Associated factor(s)
9 of 16 within-dataset runs exceeded 99.6% MCC	Sample redundancy / diversity
56 distinct instances among 105,550 Slowhttptest samples (LycoS18)	Redundancy; generation variation

Reported observation	Associated factor(s)
LOIC-HTTP pkt_len_mean: one value in LycoS18, three clusters in LycoS17	Attack-generation variation
Bot results depend on training dataset; CIC labelling errors	Label quality / consistency
Benign share 73–83% in all four datasets	Class balance; metric choice
Few features suffice within, not across, datasets	Feature-distribution shift; redundancy
PortScan and Heartbleed only in 2018 datasets	Attack coverage

A. Motivation, Evidence, and Analysis of Findings

The weakest link (LycoS17→LycoS18, ~10.83%) is due to the difference between the attack distribution from two re-extracted datasets using the same tool (meaning similar features). This means there will be less features involved in the discrepancy, but more to do with the attack distributions within the datasets. The differences between the CIC datasets are due to the number of machines (14 vs 500) and the errors made by the extraction tool [7]. These could be due to contextual or label issues. Some discrepancies may be more impactful than others, therefore considering the discrepancies together can provide useful information. The features also show how some features did help distinguish between attack classes within a dataset but not between datasets. This is due to the attacks being extremely similar [1]. Again, the features and labels are linked due to the factors they are created under. The type of model used will affect this issue. In [1] the best performing model when using cross-datasets was LDA (MCC = 60.41%). A decision tree had a score of ~1% in the same direction. This means that each model family must be individually analyzed.

B. Findings and Implications

Intra dataset validation. This tests how well the model is replicating the label process, rather than identifying attacks within a new network. This results in any artefacts being included, meaning they are positively rewarded [1]. The class imbalance worsens this. Intra-dataset bias is a confounding variable that has not been considered. This framework provides clarity on this issue by providing information on how the data was created (contextual, redundant, label process) [8][10].

High scores for intra and cross-datasets. This is positive however further validation should be done using more datasets. High intra-low cross scores (most common) indicate that the model has learnt the details of the network it was trained on. Check the factors to see if there may have been any discrepancies with the redundancies, creation, label process or features. Low scores for both indicate that the model hasn't learnt enough (or there may be issues with the data). Therefore, you shouldn't trust this model on your network. Alternatively, it could be an issue with how it has been evaluated, so the experiment should be repeated. Low intra-high cross scores are unlikely. If they occur then check for any label issues.

Practitioners. When reporting intra dataset results, an ML-IDS practitioner should not assume the model works universally. Test the model on at least one external dataset. Report more than just the accuracy (ideally MCC, F1, FPR, AUROC). Consider the factors before trusting the model to give almost perfect results. Validate the model on the target network. When publishing a dataset paper consider cross-dataset validation as a way to check the validity of your dataset. Report the tools/software used to create the data, any duplicates, where the labels came from and what features were used to differentiate. This will allow others to assign scores to the factors. When publishing a model paper mention what network(s) the model was tested on. When reviewing an ML-IDS paper ask for cross-dataset validation. Consider the factors that may differ between the source and target networks.

Link to dataset choice. The purpose of this survey [4] is to compare individual datasets against each other however each dataset is looked at individually. Survey [5] does the same. This framework builds upon the findings of [4] and [5]. It goes into more detail by comparing the factors between the source and target datasets. This creates a link between the mismatch and the loss in performance. Even if two datasets are the 'best', they may not match well so should be considered carefully. The factors do not aim to improve accuracy but encourage responsible evaluation by improving reliability, validity and interpretability.

VII. LIMITATIONS AND THREATS TO VALIDITY

This is a theoretical paper. Therefore, no new experiments were performed and no new models were trained or new factors measured. All claims regarding numbers are derived from the work of Cantone et al. [1]. Limitations include factors potentially being correlated meaning if a factor scores higher then all others will score higher. Realism, representativeness and documentation can only be measured qualitatively. The 0-3 scale is arbitrary and could be replaced. Harmonizing features and schemas require domain knowledge and could be prone to errors. Contextual realism is subjective to interpret. A dataset may be realistic for some networks but not others. Evidence is limited as it was taken from one article [1] and CIC datasets. It is yet to be proven how generalizable this framework would be. This taxonomy is descriptive not predictive. Claims regarding related works [2]-[5] are

limited to what has been analyzed in this work and how much this paper contributes.

VIII. CONCLUSION AND FUTURE SCOPE

In practice, ML-based Intrusion Detection Systems (IDS) are often evaluated on their training data. However, there appears to be some evidence suggesting that this can be deceptive; in recent studies the average MCC has dropped significantly from intra-dataset (94.63%) to inter-dataset (29.35%) performance [1]. This paper aims to investigate why there is a decrease in performance between the in-distribution and out-of-distribution scenarios by considering 18 different factors: data factors (dataset/attack characteristics, discrepancy between data and features) and context factors (degree of realism and documentation). The importance of these factors was highlighted by providing a controlled experiment with leakage precautions. We defined what we mean by generalisation gap, what metrics to use, provided pseudocode, and discussed how the results can be interpreted. There was no new experimentation carried out and this is purely a theoretical contribution. In the future, it would be important to validate our framework with real experimentation. Firstly, it should be tested using several different datasets with overlapping attack types (e.g. [1]). This would require baseline, reverse, matched-schema, matched-attack, balanced vs original, duplicate removed and context matched experiments. Each of these would test one factor in isolation. We could also test how much each factor impacts performance through ablation. The framework could be tested with multiple models (linear, tree-based, ensembles, deep learning). We could also look at how well the models generalise temporally (training on older data and testing on more recent data). Additionally, newer (or production grade) datasets could be tested where the generation process is documented. We could also control how the attacks are generated by varying them. Testing raw packet features, domain adaptation and leave-one-dataset-out multi-source training would help overcome the exposure bias in federated methods [1]. The experiments should be run with at least 10 seeds. Paired non-parametric tests should be used, bootstrap confidence intervals calculated, spearman correlation calculated between mismatch in factors and generalization gap, multiple comparison correction and exploratory analysis should be reported due to the limited amount of dataset pairs.

REFERENCES

[1] M. Cantone, C. Marrocco, and A. Bria, “Machine learning in network intrusion detection: A cross-dataset generalization study,” IEEE Access, vol. 12, pp. 144489–144508, 2024, doi: 10.1109/ACCESS.2024.3472907.

[2] A. H. Ali, M. Charfeddine, B. Ammar, B. B. Hamed, F. Albalwy, A. Alqarafi, and A. Hussain, “Unveiling machine learning strategies and considerations in intrusion detection systems: A comprehensive survey,” Front. Comput. Sci., vol. 6, Art. no. 1387354, 2024, doi: 10.3389/fcomp.2024.1387354.

[3] H. M. R. U. Rehman et al., “A systematic literature study of machine learning techniques based intrusion detection: Datasets, models, challenges, and future directions,” J. Big Data, vol. 12, Art. no. 264, 2025, doi: 10.1186/s40537-025-01323-2.

[4] P. Goldschmidt and D. Chudá, “Network intrusion datasets: A survey, limitations, and recommendations,” Comput. Secur., vol. 156, Art. no. 104510, 2025, doi: 10.1016/j.cose.2025.104510.

[5] “Security datasets for intrusion detection and prevention: A structured review and dataset-selection framework,” Appl. Sci., vol. 16, Art. no. 7473, 2026, doi: 10.3390/app16157473.

[6] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, “Toward generating a new intrusion detection dataset and intrusion traffic characterization,” in Proc. 4th Int. Conf. Inf. Syst. Secur. Privacy (ICISSP), 2018, pp. 108–116.

[7] A. Rosay, E. Cheval, F. Carlier, and P. Leroux, “Network intrusion detection: A comprehensive analysis of CIC-IDS2017,” in Proc. 8th Int. Conf. Inf. Syst. Secur. Privacy, 2022, pp. 25–36.

[8] G. Engelen, V. Rimmer, and W. Joosen, “Troubleshooting an intrusion detection dataset: The CICIDS2017 case study,” in Proc. IEEE Secur. Privacy Workshops (SPW), May 2021, pp. 7–12.

[9] L. Liu, G. Engelen, T. Lynar, D. Essam, and W. Joosen, “Error prevalence in NIDS datasets: A case study on CIC-IDS-2017 and CSE-CIC-IDS-2018,” in Proc. IEEE Conf. Commun. Netw. Secur. (CNS), Oct. 2022, pp. 254–262.

[10] M. Lanvin, P.-F. Gimenez, Y. Han, F. Majorczyk, L. Mé, and É. Totel, “Errors in the CICIDS2017 dataset and the significant differences in detection performances it makes,” in Proc. Int. Conf. Risks Secur. Internet Syst., Cham, Switzerland: Springer, 2022, pp. 18–33.

[11] M. Catillo, A. Pecchia, and U. Villano, “Machine learning on public intrusion datasets: Academic hype or concrete advances in NIDS?” in Proc.

53rd Annu. IEEE/IFIP Int. Conf. Dependable Syst. Netw., Jun. 2023, pp. 132–136.

[12] L. D'Hooge, T. Wauters, B. Volckaert, and F. De Turck, "Inter-dataset generalization strength of supervised machine learning methods for intrusion detection," J. Inf. Secur. Appl., vol. 54, Oct. 2020, Art. no. 102564.

[13] M. Verkerken, L. D'hooge, T. Wauters, B. Volckaert, and F. De Turck, "Towards model generalization for intrusion detection: Unsupervised machine learning techniques," J. Netw. Syst. Manage., vol. 30, no. 1, pp. 1–25, Jan. 2022.

[14] D. Chicco and G. Jurman, "The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation," BMC Genomics, vol. 21, no. 1, pp. 1–13, Dec. 2020.