3RD WORKSHOP ON MACHINE LEARNING IN NETWORKING (MaLeNe) PROCEEDINGS

SEPTEMBER 1, 2025

CO-LOCATED WITH
THE 6TH INTERNATIONAL CONFERENCE ON NETWORKED SYSTEMS (NETSYS 2025)
ILMENAU, GERMANY

I Choose You: Evaluating the Impact of Feature Selection on XAI Consensus for ML-NIDS

Katharina Dietz*, Johannes Schleicher§, Nikolas Wehner*, Mehrdad Hajizadeh†,
Pedro Casas‡, Stefan Geißler*, Michael Seufert§, Tobias Hoßfeld*
*University of Würzburg, Germany, †Technical University of Chemnitz, Germany
‡AIT Austrian Institute of Technology, Vienna, Austria, §University of Augsburg, Germany
*{katharina.dietz, nikolas.wehner, stefan.geissler, tobias.hossfeld}@uni-wuerzburg.de,
†mehrdad.hajizadeh@etit.tu-chemnitz.de, ‡pedro.casas@ait.ac.at, §{johannes.schleicher, michael.seufert}@uni-a.de

***Abstract*—Machine learning-based network intrusion detection systems (ML-NIDS) are increasingly enhanced with explainable AI (XAI) techniques to support transparency and trust in automated security decisions. However, recent studies have shown that different post-hoc XAI methods often yield inconsistent explanations. These variations depended on the dataset and underlying model, and were possibly caused by training the ML models on correlated features. In this work, we investigate the hypothesis that feature selection prior to model training can influence the level of consensus among XAI methods. Through a comprehensive evaluation across multiple datasets, we analyze the impact of different feature selection strategies on explanation agreement. While we found that feature selection can improve XAI consistency in controlled synthetic settings, its effects on real-world NIDS data are mixed: occasionally enhancing, but sometimes reducing consensus, while offering only modest gains over using all features. These insights highlight the importance of thoughtful feature selection to improve interpretability and consistency in XAI-driven network intrusion detection systems.**

***Index Terms*—Machine Learning, Intrusion Detection, Explainable AI, Disagreement Problem, Feature Selection.**

I. Introduction

The rapid evolution of data networks has revolutionized modern life by enabling seamless communication, automation, and large-scale data exchange. However, this increased connectivity has also expanded the attack surface, offering more opportunities for cyber adversaries. According to the European Union Agency for Cybersecurity (ENISA), there has been a marked increase in the frequency, diversity, and impact of cyber attacks [1]. Adversaries are now exploiting automation and artificial intelligence (AI) to design more evasive attack strategies [2], [3]. These developments have exposed the limitations of traditional security mechanisms, especially signature-based detection methods, which depend on predefined patterns and struggle with evolving threats [4], thereby prompting the emergence of machine learning (ML)-based network intrusion detection systems (NIDS) as promising tools to identify malware and network attacks [3], [5]–[7].

Despite these advances, the integration of ML into computer security still often encounters mistrust and skepticism [8], [9], not least due to the lack of explainability [10]. The opaque nature of many AI models limits their practical deployment, as security analysts must be able to interpret and trust the decisions made by automated systems [3], [7], [10]. In cyber-security, errors or blind trust in automated decisions can have severe consequences, potentially endangering infrastructure, privacy, and even human safety [3], [11]. While explainable AI (XAI) has emerged to demystify ML behavior, current XAI methods often produce divergent explanations, which creates confusion rather than clarity [10], [12], [13]. Thus, to ensure actionable insights for security professionals, it is essential to establish consensus among different explanation techniques. Furthermore, the European Union's General Data Protection Regulation (GDPR) reinforces this through its "right to explanation" for decisions made by algorithms [14].

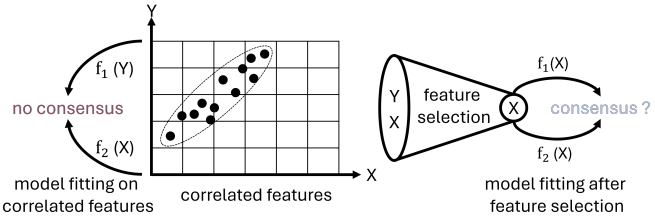


Fig. 1: Feature selection for decorrelation.

A detailed analysis on the consensus of XAI methods for ML-NIDS [13] revealed that the level of agreement between different post-hoc XAI approaches, varied significantly depending on the dataset and, in some cases, the underlying model. Disagreements might stem from the selection of related or correlated features, indicating that multiple, seemingly divergent explanations could still be valid. Figure 1 exemplifies this phenomenon, where two explainers, f_1 and f_2, select different features, X and Y, to explain the same decision. Although these features encode similar characteristics, their disagreement results in a lack of consensus. When applying feature selection (FS), both explainers may agree on the same feature. This suggests that FS before model training might play a crucial role w.r.t. consensus, and that we can potentially improve the consensus by using the right FS method.

In this work, we conduct extensive experiments on both synthetic and real-world NIDS datasets, using six different FS strategies and two widely used post-hoc XAI methods. To the best of our knowledge, this is the first systematic analysis of the relationship between FS and XAI disagreement by bringing the disagreement problem into the context of NIDS. We highlight both the potential and limitations of improving

explanation consistency, and more broadly, the challenges of relying on post-hoc XAI for trustworthy interpretations.

The remainder of this paper is structured as follows: Section II provides information on intrusion detection, XAI, FS, and related works. Section III outlines the used datasets and ML workflow for analyzing consensus. Section IV presents the obtained results, and finally, Section V summarizes the key findings and contributions of this study.

II. Background and Related Work

A. ML-NIDS

In network security, an intrusion implies that the *confidentiality*, *integrity*, or *availability* of network resources (e. g., devices, data) is being compromised [15]. Thus, intrusion detection systems (IDS) are employed to detect such attacks and enable further steps for their containment. IDS can be deployed at different vantage points, e. g., in global locations to monitor the network traffic on a large scale (NIDS) or directly on the host (HIDS) to locally investigate malicious programs, files, etc. The former is often based on features extracted from coarse-grained flows (i. e., aggregates based on the 5-tuple of IPs, ports, and protocol such as NetFlow) or directly on the packets for more fine-grained monitoring.

Research has shifted towards ML-NIDS, since recent trends such as 5G/6G, Internet of Things (IoT), and Industry 4.0 have increased attack surfaces, necessitating more sophisticated countermeasures. Though, ML-based solutions are often perceived as more complex than traditional solutions (e. g., based on signatures or rules) [16] and deemed less trustworthy, particularly in sensitive areas such as cybersecurity. So, techniques that provide model insights should be adopted, especially regarding the GDPR [14] and European AI Act [17].

B. Explainable AI

The aforementioned challenges have led to the development of XAI to provide insights into the decisions of ML models, either by utilizing white-box models that are interpretable by design or by utilizing post-hoc explainers, which explain the output of black-box models [10]. XAI methods can be classified in various ways, e. g., w.r.t. model compatibility or algorithm type [18]. In this work, we focus on two of the most prominent post-hoc XAI representatives: LIME (Local Interpretable Model-agnostic Explanations) [19] and SHAP (SHapley Additive exPlanations) [20]. Generally, both of them are compatible with any ML model and work via *perturbation*, i. e., masking or obfuscating input features of a data sample to determine their influence on the model's decision. That is, LIME builds a linear (therefore interpretable) surrogate model by learning the output of the original model by adding noise to each input feature. SHAP, on the other hand, is based on Shapley values [21] from game theory. This concept assigns feature contributions by evaluating all possible feature subsets.

Even though XAI has gained increasing popularity, new challenges arise, e. g., in form of the *disagreement problem* [12]. This problem stems from the fact that produced explanations by the various explainers often differ, sometimes even contradict. This has been observed for use cases like network security [10], [13], [22]–[24], but also areas outside of communication networks [12], [25], [26]. Feature interactions, such as correlation or other relationships, are often cited to be a contributing factor to this phenomenon [13], [25]–[30]. Naturally, redundant features enable multiple valid explanations.

C. Feature Selection

Selecting only a relevant feature subset is a standard process in many ML workflows, since fewer features not only improve training and inference time, but also reduce overfitting [18]. They also improve interpretability, since less features have to be interpreted. Here, reducing the number of features may help improve XAI consensus twofold. One, the explainers have less choices to choose from in general, and two, depending on the FS mechanism, feature interactions may be reduced.

Similar to XAI, FS aims to identify the top features. This is a preprocessing step (i. e., before model training), while post-hoc XAI is applied after model training. Stańczyk [31] (as well as Khani et al. [18]) groups FS into three groups: *filters*, *wrappers*, and *embedded* techniques, many of which are implemented in scikit-learn [32]. One of the most well-known selection techniques is impurity-based FS that leverages the structure of tree-based models. This is an *embedded* technique, since it makes use of the internals of a pretrained model. It quantifies how much a feature reduces the "impurity" at each split across all trees w.r.t. the mix of class labels.

Instead of making use of model internals, *wrappers* utilize a classifier to evaluate different feature (sub)sets and their "usefulness in classification" [31, p. 32]. That is, they observe how the model's performance (e. g., accuracy) changes under different conditions. For example, permutation importance shuffles the value of a feature. Alternatively, recursive feature elimination starts with a full feature set and iteratively prunes the feature with the least impact. Similarly, backward sequential FS works accordingly, while the forward variant starts from an empty feature set.

Lastly, *filters* work separately from any ML model and observe the relationship between a feature and the class label, e. g., via ANOVA F-values [18], as implemented by default by sklearn's *SelectKBest()*. Another approach clusters correlated features together by treating correlation as a similarity measure, before simply choosing one feature from each group [33].

D. Related Work

Many works on XAI-driven NIDS have already shifted their focus to a quantitative comparison of different explainers instead of merely using XAI to explain a decision, e. g., regarding an explanation's robustness or faithfulness to the ground truth [22], [34]–[38], and/or conduct qualitative studies with security admins to gain insights from practitioners [24]. Some works leverage XAI itself for FS (e. g., [18], [39]), whereas we view this as two separate steps in the ML pipeline. The disagreement problem is sometimes a partial factor in these works, but rarely a focal point. In contrast, our work focuses on investigating the XAI consensus in a more detailed manner.

Our goal goes beyond stating the existence of the disagreement problem, which we explored ourselves previously [13].

Besides works on XAI-NIDS, other research areas have put a more in-depth emphasis on the disagreement problem by investigating the impact of varying model parameters or steps in the ML workflow. One approach is limiting the scope of the explanation to *regional areas* [40], [41], i. e., adjusting the background datasets of explainers to be more locally relevant to the instance to explain. Instead of restricting the reference data, other works limit the actual input features via dimensionality reduction to reduce multicollinearity [13], [26], [42]. Lastly, other works analyze the reasons of the disagreement problem by controlling dataset parameters (e. g., features, samples, labels, noise, redundancy) [27], [28], while others explore the impact of different preprocessing techniques (e. g., scaling, encoding) [43], or influence of model parameters (e. g., training duration and loss functions) [30], [44]. Often, these works also make use of synthetic data, which is easier to configure. While there exist some works that make use of FS, as well as investigating feature interactions, our work specifically zeroes in on the differences between various selection methods. To the best of our knowledge, we are also the first to bring the disagreement problem into the NIDS domain in-depth (or monitoring in general).

III. Methodology

A. NIDS Datasets

In this work, we use three NIDS datasets of varying complexity and feature granularity: CICIDS2017 [45], CIDDS-001 [46], and Edge-IIoTset [47]. CICIDS2017, one of the most popular NIDS datasets in state-of-the-art literature [7], provides 77 flow-based features[1], e. g., statistical moments of packet sizes and IATs. We use the Wednesday subset with almost 700k samples and DoS/DDoS attacks. CIDDS-001 offers 14 features based on NetFlow, which is one of the most commonly used protocols in practice for traffic monitoring. We use the first week of the dataset (over 8M samples), which includes Pingscan, Portscan, Bruteforce, and DoS attacks. Note that we additionally derived flow IATs and number of parallel flows to enrich the feature set. Edge-IIoTset covers diverse IoT/IIoT protocols (e. g., TCP, MQTT, MODBUS) with 35 features for over 2M samples, including DDoS, Portscan, and other attacks. While we generally follow the authors' proposed preprocessing steps, we remove further features with limited generalizability (e. g., IPs, checksums, ACK numbers). Our code is available for reproducibility[2].

B. XAI Workflow

To ensure temporally coherent splits, we use sklearn's *StratifiedGroupKFold()*, where groups are defined as 30s time intervals w.r.t. each dataset's timestamp column[3], instead of simply shuffling the entire dataset randomly before splitting. We use three folds, ensuring that each sample appears during testing. Categorical features are one-hot-encoded, zero-variance features are filtered out, and features are minmax-scaled. The top ten features are selected via the six selection methods described in Section II-C: impurity-based, permutation-based, recursive, (forward) sequential, and correlation-based FS, as well as *SelectKBest()*. Similar subsets have been found useful in related work on NIDS [18], [39]. For FS methods that require a classifier, we utilize a lightweight Random Forest (RF; 10 trees, max. depth 10). After encoding and filtering, we balance the training data by selecting 250k samples of each class (benign, malign). For the actual classification task, we also use an RF (50 trees, max. depth 20) and a Multi-Layer-Perceptron with two layers (MLP; 64 neurons per layer, followed by ReLU). Both are commonly used in recent XAI [12], [30] and NIDS literature [7], giving insights for shallow ML and Deep Learning (DL).

For our explainers, we utilize the aforementioned LIME and SHAP. For the latter, we use the more efficient, model-specific implementations (TreeSHAP, DeepSHAP). To calculate the consensus between pairs of explanations (i. e., SHAP vs. LIME-based explanations), we use metrics similar to Krishna et al. [12]. We focus on two types: unordered (UC) and ordered consensus (OC) of the top 5 features. For the UC, we simply calculate the intersection of features. For the OC, we take the actual order of importance into account. In detail, we compute how many of the top features match in order until the first mismatch. Consequently, we are not interested if, e. g., only the fifth feature matches if previous features do not. We only count features as matching if their sign also matches.

C. Synthetic Data

We also utilize synthetic data to analyze the effect of FS in a configurable manner, for which we make use of sklearn's *make_classification()*, adapted from a benchmark data generator for a FS competition [48]. The algorithm has four feature types: *informative*, *redundant*, *repeated*, and *useless*. The *informative* features are the ones actually relevant to the prediction target, and the informativeness is split among all of them. The *redundant* features are linear combinations of other features, while *repeated* features are simple duplications. Last, *useless* features are just noise. For all feature types, we add new features (up to 50 extra, in increments of 10), select the top 10, and calculate the XAI consensus. Each experiment starts with five informative features. For each combo of type and number, we generate 250 balanced synthetic datasets of 1k samples, which we split in a 80:20 ratio.

IV. Evaluation

A. Preliminary Experiments on Synthetic Data

Before diving into the experiments of the NIDS datasets, we first want to analyze the impact of FS in a controllable fashion. Figure 2 illustrates the results of various experiments described previously. Each row of subfigures illustrates the four different feature types, while the columns of subfigures represent the two consensus metrics for all 200 test set samples, as well as the accuracy. Since the synthesized datasets are balanced,

[1] Before encoding, filtering etc. (for all datasets).

[2] https://github.com/lsinfo3/malene2025-xai-nids-feature-selection

[3] Edge-IIoTset has >100k samples with invalid timestamps, which we drop.

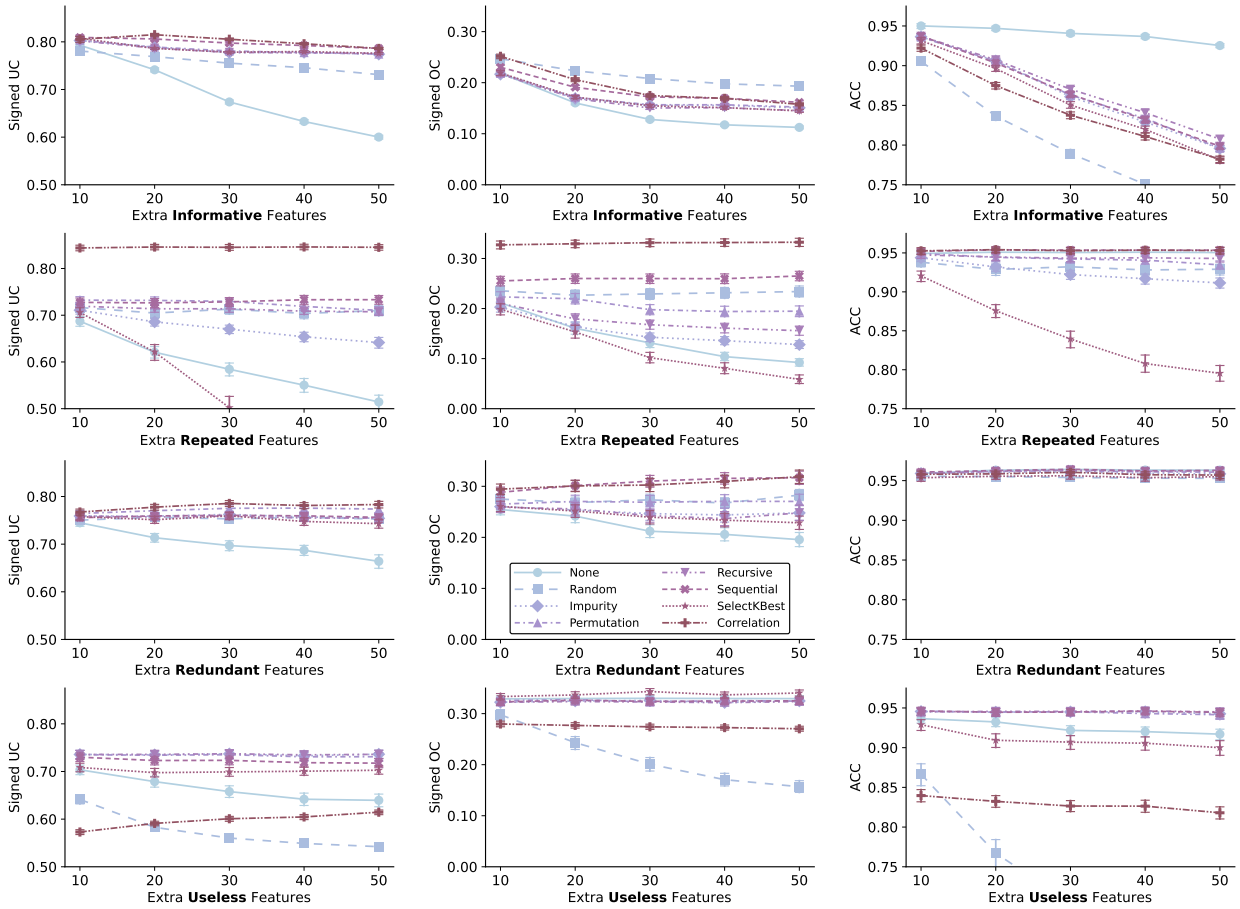
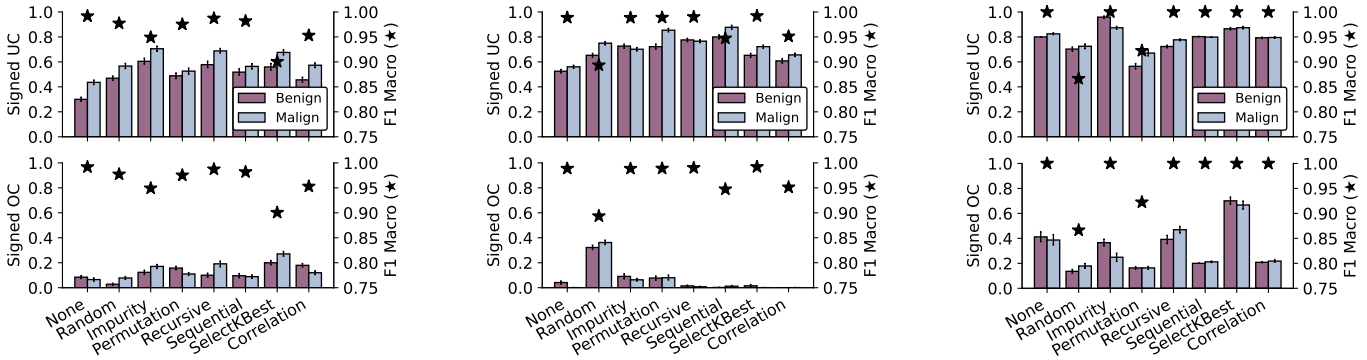


Fig. 2: Comparison of FS methods across feature types (rows) and (un)ordered consensus+accuracy (columns; UC/OC, ACC).

using accuracy is adequate here. The x-axis represents the extra added features (in addition to the five starting features), and the y-axis the respective metric. The different linestyles depict the different FS methods as well as results with all features ("None") and a random FS. Errorbars depict the 95% confidence intervals of the 250 runs. For the sake of brevity, results are shown for the RF as underlying model.

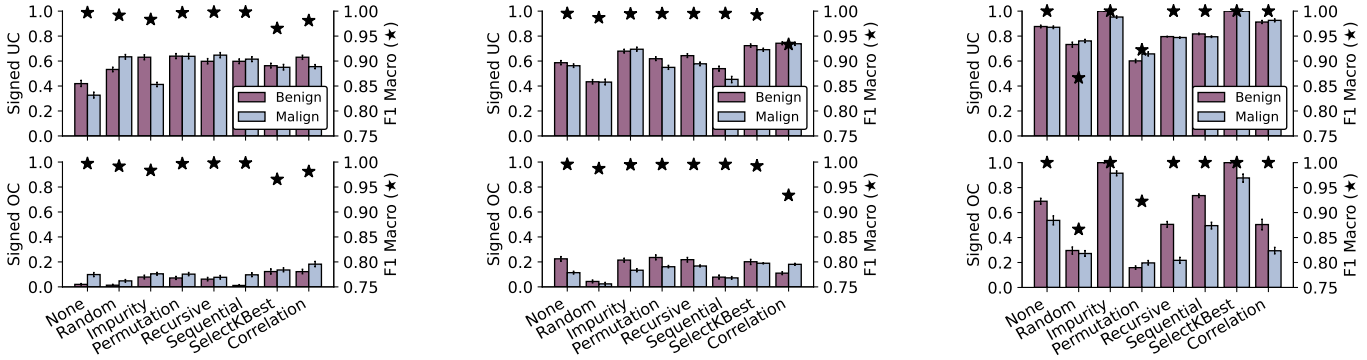
For the *informative* features in the first row, the UC stays almost constant for all FS methods, but drops significantly when no FS is applied. We can also observe a slight edge of the correlation-based and the sequential approach over the others. Since informative features are generated in a way so they do indeed contain covariance, taking this into account may help slightly. For the OC, we see a similar trend, though the difference between no FS and the rest is less severe and the consensus is generally lower. Interestingly, the random baseline comes out on top here. For the actual accuracy, however, we see that only keeping all features maintains model performance. This is expected, as we increase the number of informative features, which are all relevant to the classification, retaining only the top 10 is not sufficient anymore. We also see that while the correlation-based approach has a slight edge for the consensus, it slightly underperforms. We also see that, despite being best for the OC, the random baseline drastically underperforms, too. Our hypothesis is, that the OC may be increased since the random approach might have chosen one feature that may be most import, while the rest is not as descriptive (as reflected by the accuracy), making the explainers agree on that feature. In other words, the FS is so suboptimal, that it makes explainers agree on the top feature.

For the *repeated* features in the second row, we see more distinct trends. For the UC, the correlation-based selection outshines the rest, since it is able to determine that in total only five features are relevant, since it can only establish five clusters, because all features are perfect duplicates. This is followed by the three wrapper-based methods. Interestingly, even the permutation-based approach is able to keep the UC constant. Although permutation is prone to correlation, this effect is negated here since all initial features have the same chance to be duplicated. In other words, while correlated features dampen each other's importance, this effect happens for all features uniformly. This is also the reason why the random baseline performs decent, as on average, it will choose each original feature at least once. The impurity-based approach and *SelectKBest()*, however, are both biased towards the top feature. Especially the latter even drops its consensus below the baseline. In the worst-case scenario, the selected features only contain a single feature and its replicates, thus making it hard for SHAP and LIME to determine which is most important. For the OC, we generally see a similar trend to



(a) CICIDS (b) CIDDS (c) Edge-IIoT

Fig. 3: Signed UC (top) and Signed OC (bottom) on NIDS data with **MLP** as underlying model.



(a) CICIDS (b) CIDDS (c) Edge-IIoT

Fig. 4: Signed UC (top) and Signed OC (bottom) on NIDS data with **RF** as underlying model.

the UC. Though, since it is generally lower, the baseline and *SelectKBest()* cannot drop as steep numerically. The accuracy also shows the suboptimal FS of *SelectKBest()*.

The *redundant* features in the third row generally show a more "noisy" trend compared to the previous feature types. That is, while this feature type still contains redundancies, they are more obfuscated. In other words, choosing a different feature set other than the initial features still holds value. For both the UC and OC we still see the correlation-based approach having an edge over the others, though far less significant than for the repeated features. Additionally, *SelectKBest()* and the impurity-based approach also do not diverge as strongly. For the accuracy, all methods are able to keep up the performance, since there is no issue of choosing (perfect) duplications.

Lastly, the *useless* features in the fourth row show a contrasting trend to the previous features. For the UC, we now see that the correlation-based approach performs significantly worse then the others, as well as illustrating why random FS is a bad idea. Even no FS is better here. The same can be observed for the OC, and is also reflected in the accuracy dropping. Since the correlation-based approach clusters the features first, it may choose a handful of noise. Similarly, the random FS will choose noise. Since this noise actually has no meaning for the explanations, it makes it increasingly hard for the explainers to decide which of these noisy features are actually more relevant, similar to the perfectly correlated features. Contrary to the previous feature types, the baseline without FS performs worse since it also uses the noise to train.

Discussion: The synthetic experiments validate our core assumptions: FS can improve consensus, particularly when feature interactions such as correlation or redundancy are present. Selection methods that respect such interactions outperform simpler approaches like *SelectKBest()*. However, when the added features are purely noisy, correlation-based selection may actually reduce consensus and accuracy by mistakenly prioritizing irrelevant features. Overall, sequential selection proved to be a balanced choice across all feature types.

B. Final Experiments on NIDS Datasets

We now shift our focus to more realistic NIDS data, which may not contain perfectly (un)correlated features, but more complex feature relationships. For this, Figures 3 and 4 show the consensus for all three previously described NIDS datasets as a barplot, for MLP and RF, respectively. The top row depicts the UC and the bottom row shows the OC. The x-axis depicts the selection method, while the left y-axis holds the consensus. The consensus is the average of 100 random samples from the test set for all three splits, i. e., each bar is made up by $3 \cdot 100$ values. We divide the consensus analysis between our two classes, as the datasets are imbalanced. The errorbars contain the 95% confidence intervals. The right y-axis marks the average macro F1 to take into account imbalances via stars. We also utilize the macro F1 for FS for performance scoring here where applicable, as the default is accuracy.

Starting with the MLP as underlying model in Figure 3 and the UC (top row), we see that no selection method really stands out, whether it is w.r.t. consensus or performance. The only method that consistently has similar performance to the baseline with no filtering is the recursive selection. For CICIDS, the majority of the methods are able to at least slightly boost the consensus, though only marginally in most cases. Features are reduced from around 70 features to only 10, but the consensus increase is nowhere near drastic. This also comes at a performance loss in some cases. For example, the impurity-based approach increases the consensus the most, but performance drops by roughly 5%. Interestingly, even a random FS performs better accuracy-wise. For CIDDS, results are similar concerning the UC boost. For Edge-IIoT, the consensus is already quite high, so any further filtering does not actually have a great impact. Interestingly, *SelectKBest()* and the impurity-based approach are the only ones that increase the UC here, which may seem counterintuitive compared to the synthetic data. For the OC (bottom row), we see similar observations. While the FS is able to establish some consensus in some cases, the improvement is not as drastic as desired. Again, for Edge-IIoT, consensus is higher and filtering can actually have a significant negative effect.

The results for the RF as underlying model are depicted in Figure 4. How the selection methods perform w.r.t. F1 follows similar trends as for the MLP, though the score is consistently higher. While the consensus is in a similar range compared to before, there are some nuanced differences. Which method improves the consensus most for the datasets is partially inconsistent with the results from the MLP, i. e., not following a noticeable pattern. For the UC, filtering features may have a positive effect for CICIDS and CIDDS, while for Edge-IIoT the baseline already reaches near perfect consensus, despite using much more features. For the OC, the consensus of CICIDS is only slightly improved by filtering and for CIDDS filtering has no huge impact either. For Edge-IIoT, *SelectKBest()* and impurity-based FS are the only methods that increases the consensus meaningfully here.

Discussion: Results on real-world NIDS datasets show that the impact of FS on consensus is less predictable. While consensus can improve over using all features, the gains are generally modest. CICIDS generally responded the best to FS. Some of its features are various statistical moments of packet sizes, IATs etc., which are naturally more correlated. CIDDS is also contains flow-based data, but already has very few features, thus FS will not have as much of an impact. For Edge-IIoT, *SelectKBest()* and impurity-based FS performed best, which is in stark contrast to the previous analyses. We hypothesize this is due to a few highly indicative (and likely related) features. More sophisticated methods may discard these in favor of spreading importance across weaker (thus noisier) features, reducing consensus. This is supported by the fact that random FS dropped performance the most drastically here for both MLP and RF. In other words, the other datasets contain more related features in terms of informativeness, i. e., random FS is partially feasible. Overall, the impact of FS is harder to assign for NIDS data, since it is composed of a more mixed set of features w.r.t. noise and correlation, so effects supposedly counteract each other.

TABLE I: MSE $\pm$ 95% CI for the toy example, 2.5k trials.

	X_1	X_2	**Both**
LR	0.53 ± 0.01	69.8 ± 0.61	0.00 ± 0.00
RF	0.60 ± 0.01	0.62 ± 0.01	0.56 ± 0.01

To illustrate the impact of removing correlated features on the actual model performance, we trained a linear regression (LR) and RF on two highly correlated inputs (one linear and one exponential feature, X_1 and X_2), and defined the regression target as their sum. Using both features yielded lower errors than only one (see Table I). Thus, unless a feature is an identical copy of another, correlated features might still carry information. These findings emphasize that reducing features to minimize potential correlations does not always help, both in terms of consensus as well as performance, and is highly dependent on dataset and model.

To summarize the findings of this work in a more practical context, post-hoc explanations and their resulting consensus are highly sensitive to slight changes in the ML pipeline. Filtering *can* help by removing totally irrelevant noise or avoiding actual duplicates (e. g., analogous to radius vs. diameter of a circle), but it is not a cure for all problems by simple application and still requires critical thinking. It also raises the need for interpretable consensus metrics, which can be misleading when inflated by highly indicative, but correlated features (as potentially seen with the Edge-IIoT dataset).

V. Conclusion

In this work, we analyzed how feature selection influences the disagreement problem between post-hoc XAI methods in ML-NIDS. On synthetic datasets, where we could control redundancy and noise, feature selection behaved as expected by improving explanation consistency in a predictable manner. However, results on real-world NIDS datasets were less consistent. While in some cases feature selection modestly improved agreement between explainers, it sometimes actually reduced consensus, particularly when selection methods emphasized noisy features. Overall, the improvement over the full-feature baseline was often smaller than maybe desired. If aligning explainers remains unreliable, it hinders the practical adoption of these methods, and they risk becoming "rebranded" standard feature importance metrics. Our results suggest a need for stronger emphasis on inherently interpretable models, rather than reliance on post-hoc techniques that struggle to produce consistent and trustworthy explanations.

Acknowledgment

This work has been partly funded by the Bavarian Ministry of Economics, Regional Development and Energy (StMWI) as part of the project VIPNANO (DIK-2307-0006), by Deutsche Forschungsgemeinschaft (DFG) under grant SE 3163/3-1, project nr.: 500105691 (UserNet), and by the German Federal Ministry of Research, Technology and Space (BMFTR) under grant 18KIS2282 (SUSTAINET-Advance). The authors alone are responsible for the content.

REFERENCES

[1] C. Ardagna, S. Corbiaux, K. Van Impe, and R. Ostadal, "ENISA threat landscape 2023," *European Union Agency for Cybersecurity*, 2023.

[2] F. N. Motlagh *et al.*, "Large language models in cybersecurity: State-of-the-art," *arXiv:2402.00891*, 2024.

[3] Z. Zhang *et al.*, "Explainable artificial intelligence applications in cyber security: State-of-the-art in research," *IEEE Access*, vol. 10, 2022.

[4] L. Caviglione *et al.*, "Tight arms race: Overview of current malware threats and trends in their detection," *IEEE Access*, vol. 9, 2020.

[5] D. Ucci, L. Aniello, and R. Baldoni, "Survey of machine learning techniques for malware analysis," *Comput. Secur.*, vol. 81, 2019.

[6] M. A. Talib *et al.*, "APT beaconing detection: A systematic review," *Comput. Secur.*, vol. 122, 2022.

[7] K. Dietz *et al.*, "The missing link in network intrusion detection: Taking AI/ML research efforts to users," *IEEE Access*, vol. 12, 2024.

[8] G. Apruzzese, P. Laskov, and J. Schneider, "SoK: Pragmatic assessment of machine learning for network intrusion detection," in *IEEE Eur. Symp. Secur. Priv. (EuroS&P)*, 2023.

[9] S. Oesch *et al.*, "An assessment of the usability of machine learning based tools for the security operations center," in *Proc. Int. Conf. Internet Things (iThings)*, 2020.

[10] A. Nadeem *et al.*, "SoK: Explainable machine learning for computer security applications," in *IEEE Eur. Symp. Secur. Priv. (EuroS&P)*, 2023.

[11] G. Jaswal, V. Kanhangad, and R. Ramachandra, *AI and Deep Learning in Biometric Security: Trends, Potential, and Challenges*. CRC, 2021.

[12] S. Krishna *et al.*, "The disagreement problem in explainable machine learning: A practitioner's perspective," *Trans. Mach. Learn. Res. (TMLR)*, 2024.

[13] K. Dietz *et al.*, "Agree to disagree: Exploring consensus of XAI methods for ML-based NIDS," in *Workshop Netw. Secur. Oper. (NeSecOr)*, 2024.

[14] B. Goodman and S. Flaxman, "European union regulations on algorithmic decision-making and a 'right to explanation'," *AI Mag.*, vol. 38, no. 3, 2017.

[15] T. Grance *et al.*, "Guide to information technology security services," *NIST Special Publication 800-35*, 2003.

[16] J. Mink *et al.*, "Everybody's got ML, tell me what else you have: Practitioners' perception of ML-based security tools and explanations," in *IEEE Symp. Secur. Priv. (S&P)*, 2023.

[17] https://eur-lex.europa.eu/eli/reg/2024/1689, accessed: 2025-05-03.

[18] P. Khani, E. Moeinaddini, N. D. Abnavi, and A. Shahraki, "Explainable artificial intelligence for feature selection in network traffic classification: A comparative study," *Trans. Emerg. Telecommun. Technol. (ETT)*, vol. 35, no. 4, 2024.

[19] M. T. Ribeiro, S. Singh, and C. Guestrin, "'Why should I trust you?': Explaining the predictions of any classifier," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Min. (KDD)*, 2016.

[20] S. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst. (NeurIPS)*, 2017.

[21] L. Shapley, "A value for n-person games," *Contrib. Theory Games*, vol. 2, 1953.

[22] A. Warnecke, D. Arp, C. Wressnegger, and K. Rieck, "Evaluating explanation methods for deep learning in security," in *IEEE Eur. Symp. Secur. Priv. (EuroS&P)*, 2020.

[23] L. Rui and O. Gadyatskaya, "Position: The explainability paradox-challenges for XAI in malware detection and analysis," in *IEEE Eur. Symp. Secur. Priv. Workshops (EuroS&PW)*, 2024.

[24] D. Bhusal *et al.*, "SoK: Modeling explainability in security analytics for interpretability, trustworthiness, and usability," in *Proc. 18th Int. Conf. Availab. Reliab. Secur. (ARES)*, 2023.

[25] M. Flora, C. Potvin, A. McGovern, and S. Handler, "Comparing explanation methods for traditional machine learning models part 1: an overview of current methods and quantifying their disagreement," *arXiv:2211.08943*, 2022.

[26] ——, "Comparing explanation methods for traditional machine learning models part 2: Quantifying model explainability faithfulness and improvements with dimensionality reduction," *arXiv:2211.10378*, 2022.

[27] Z. Carmichael and W. Scheirer, "How well do feature-additive explainers explain feature-additive predictors?" in *Workshop XAI Action: Past Present Future Appl. (NeurIPS XAIA)*, 2023.

[28] A. F. Markus *et al.*, "Understanding the size of the feature importance disagreement problem in real-world data," in *ICML 3rd Workshop Interpret. Mach. Learn. Healthc. (IMLH)*, 2023.

[29] F. Fumagalli *et al.*, "Unifying feature-based explanations with functional ANOVA and cooperative game theory," in *Proc. 28th Int. Conf. Artif. Intell. Stat. (AISTATS)*, 2025.

[30] A. Schwarzschild *et al.*, "Reckoning with the disagreement problem: Explanation consensus as a training objective," in *Proc. 2023 AAAI/ACM Conf. AI Ethics Soc. (AIES)*, 2023.

[31] U. Stańczyk, "Feature evaluation by filter, wrapper, and embedded approaches," in *Feature Sel. Data Pattern Recognit.*, 2014.

[32] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res. (JMLR)*, vol. 12, 2011.

[33] https://scikit-learn.org/stable/auto_examples/inspection/plot_permutation_importance_multicollinear.html, accessed: 2025-05-29.

[34] O. Arreche, T. R. Guntur, J. W. Roberts, and M. Abdallah, "E-XAI: Evaluating black-box explainable AI frameworks for network intrusion detection," *IEEE Access*, vol. 12, 2024.

[35] A. N. Gummadi, O. Arreche, and M. Abdallah, "A systematic evaluation of white-box explainable AI methods for anomaly detection in IoT systems," *Internet of Things*, vol. 30, 2025.

[36] J. Tritscher *et al.*, "Evaluation of post-hoc XAI approaches through synthetic tabular data," in *Found. Intell. Syst.: 25th Int. Symp. (ISMIS)*, 2020.

[37] J. Tritscher, M. Wolf, A. Hotho, and D. Schlör, "Evaluating feature relevance XAI in network intrusion detection," in *World Conf. Explain. Artif. Intell. (xAI)*, 2023.

[38] O. Lukás and S. García, "Bridging the explanation gap in AI security: A task-driven approach to XAI methods evaluation." in *Proc. 16th Int. Conf. Agents Artif. Intell. (ICAART)*, 2024.

[39] O. Arreche, T. Guntur, and M. Abdallah, "XAI-based feature selection for improved network intrusion detection systems," *arXiv:2410.10050*, 2024.

[40] G. Laberge, Y. B. Pequignot, M. Marchand, and F. Khomh, "Tackling the XAI disagreement problem with regional explanations," in *Proc. 27th Int. Conf. Artif. Intell. Stat. (AISTATS)*, 2024.

[41] S. Aswani and S. D. Shetty, "Explainable news summarization–analysis and mitigation of disagreement problem," *arXiv:2410.18560*, 2024.

[42] P. Alves *et al.*, "Comparing LIME and SHAP global explanations for human activity recognition," in *Intell. Syst.: 34th Braz. Conf. (BRACIS)*, 2024.

[43] N. Koenen and M. N. Wright, "Toward understanding the disagreement problem in neural network feature attribution," in *World Conf. Explain. Artif. Intell. (xAI)*, 2024.

[44] P. Silva, C. T. Silva, and L. G. Nonato, "Exploring the relationship between feature attribution methods and model performance," in *Proc. 2024 AAAI Conf. Artif. Intell.*, 2024.

[45] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization." *Proc. 4th Int. Conf. Inf. Syst. Secur. Priv. (ICISSP)*, 2018.

[46] M. Ring *et al.*, "Flow-based benchmark data sets for intrusion detection," in *Proc. 16th Eur. Conf. Cyber Warf. Secur. (ECCWS)*, 2017.

[47] M. A. Ferrag *et al.*, "Edge-IIoTset: A new comprehensive realistic cyber security dataset of IoT and IIoT applications for centralized and federated learning," *IEEE Access*, vol. 10, 2022.

[48] I. Guyon, "Design of experiments of the NIPS 2003 variable selection benchmark," in *NIPS Workshop Feature Extr.*, 2003.