Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

An Integrated S-SFIX and FFNN Framework for Explainable and Robust Feature Reduction in IoT DDoS Detection

Abdullah Emir Çil

abdullah.cil@marmara.edu.tr

Marmara University

Ayla GÜLCÜ

Bahçeşehir University

Buket DOĞAN

Marmara University

Serhat ÖZEKEŞ

Marmara University

Kazım YILDIZ

Marmara University

Research Article

Keywords:

Posted Date: November 27th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11501999/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

An Integrated S-SFIX and FFNN Framework for Explainable and Robust Feature Reduction in IoT DDoS Detection

Abdullah Emir ÇİL[1*]
Ayla GÜLCÜ[2†]
Buket DOĞAN[3‡]
Serhat ÖZEKES[3§]
Kazım YILDIZ[3¶]

March 2026

Abstract

The effective detection of DDoS attacks in IoT ecosystems remains a challenge, primarily due to high-volume traffic and the limited computational resources of edge devices. While high accuracy is essential, practical deployment requires feature representations that are compact, interpretable, and robust. To this end, we introduce an integrated S-SFIX–FFNN framework designed for explainable feature reduction. Our approach employs Simplified SFIX (S-SFIX) to generate low-dimensional feature subsets, which are then evaluated via a Feed-Forward Neural Network (FFNN). Using the CICIoT2023 dataset, we simplified the 33 attack subtypes into a binary classification problem (Benign vs. Attack). To establish a baseline, an FFNN was first trained on the full set of 46 features. We then benchmarked the S-SFIX selected subsets (top 5, 10, 15, and 20 features) against established methods including SHAP, LIME, and Random Forest feature importance. The results indicate that S-SFIX achieves a significant reduction in dimensionality without compromising performance; the framework maintained 98.05% accuracy with only 20 features, compared to the 98.45% baseline. Even at a 78% reduction rate (10 features), the detection capability remained stable. Notably, S-SFIX outperformed Random Forest in low-dimensional settings and matched the performance of more computationally expensive explainability tools like SHAP. Robustness tests under Gaussian noise further validate the framework. While noise naturally affects feature rankings, the core "Top-10" set remained consistent. This stability suggests that the S-SFIX–FFNN integration provides a reliable mechanism for IoT security, ensuring both explainability and resilience against data corruption.

Keywords

XAI, S-SFIX, Feature Reduction, FFNN, IoT DDoS Detection, AI Resilience

1 Introduction

As AI-driven decision systems become embedded in high-stakes domains like cybersecurity and finance, the demand for both transparency and resilience has intensified [1]. Relying solely on predictive accuracy is no longer viable; the opaque nature of complex, "black-box" models poses significant risks to regulatory compliance, auditing, and overall operational trust.

While Explainable Artificial Intelligence (XAI) methods, such as SHAP and LIME [2], have successfully made model outputs more interpretable, they are typically evaluated under static data assumptions. In real-world environments, however, the consistency of these explanations can degrade rapidly under distribution

*abdullahemircil@hotmail.com
†ayla.gulcu@bau.edu.tr
‡buketb@marmara.edu.tr
§serhat.ozekes@marmara.edu.tr
¶kazim.yildiz@marmara.edu.tr

and domain shifts [3, 4]. This vulnerability highlights the growing need for AI resilience—the capacity of a system to maintain consistent functionality and adapt to adversarial noise or unexpected data variations [1, 5]. In high-risk scenarios, it is not enough for a model to be merely explainable; its underlying logic must also be robust [6].

IoT-based DDoS detection perfectly exemplifies this challenge. Beyond maximizing predictive accuracy, combating heterogeneous attack patterns in noisy environments requires detection frameworks that are compact, interpretable, and robust. Effective feature reduction is critical here, as it isolates the most discriminative variables and lowers model complexity. Yet, if the underlying feature ranking is unstable, the overall reliability of the intrusion detection system is fundamentally compromised.

To address these interconnected requirements, we propose an integrated S-SFIX–FFNN framework for robust and explainable feature reduction in IoT DDoS detection. We adapt the SFIX approach—originally designed for financial decision support [7]—by removing its time-dependent terms to create a Simplified SFIX (S-SFIX) suitable for static datasets. In our architecture, S-SFIX handles the explainable feature scoring and reduction, while a Feed-Forward Neural Network (FFNN) serves as the classification engine. Together, they provide a unified experimental framework to systematically evaluate feature compactness, classification performance, and resilience against data perturbation.

The main objectives of this study can be summarized as follows:

- To propose an integrated S-SFIX–FFNN framework for explainable and robust feature reduction in IoT-based DDoS attack detection.
- To use S-SFIX as a multi-component feature scoring and reduction mechanism that identifies compact and discriminative feature subsets.
- To evaluate the extent to which FFNN models trained with S-SFIX-selected low-dimensional feature subsets can preserve classification performance compared with the full-feature FFNN baseline.
- To compare the proposed S-SFIX–FFNN framework with SHAP-, LIME-, and Random Forest-based feature reduction configurations under the same FFNN architecture, preprocessing pipeline, and data split.
- To analyze the robustness of the S-SFIX feature ranking mechanism under Gaussian noise using score-based, rank-based, and Top-k preservation metrics.

Accordingly, the study is structured around two complementary axes. First, it evaluates the extent to which the integrated S-SFIX–FFNN framework can preserve IoT DDoS detection performance under different feature reduction scenarios. In this axis, S-SFIX generates compact and interpretable feature subsets, while FFNN evaluates the classification capability of these subsets. Second, the proposed framework is compared with SHAP-, LIME-, and Random Forest-based feature reduction configurations, and the stability of the S-SFIX ranking mechanism is analyzed under Gaussian noise. By addressing these dimensions, we demonstrate that the S-SFIX–FFNN integration functions as a unified, explainable framework that effectively bridges the gap between dimensionality reduction, classification power, and operational resilience.

The main contributions of this study can be summarized as follows:

- It proposes an integrated S-SFIX–FFNN framework in which S-SFIX performs explainable feature scoring and reduction, while FFNN serves as the classification layer for IoT DDoS detection.
- It demonstrates that the proposed framework can maintain competitive detection performance with substantially fewer input features, reducing the feature space from 46 features to 20, 15, and 10 features with limited performance degradation.
- It provides a controlled comparative evaluation against SHAP-, LIME-, and Random Forest-based feature reduction configurations using the same FFNN architecture and experimental pipeline.
- It evaluates the robustness of the S-SFIX ranking mechanism under Gaussian noise through Pearson score correlation, Spearman rank correlation, Kendall Tau, rank-shift measures, Jaccard similarity, and Top-k retention counts.
- It provides empirical evidence that the integration of explainable feature reduction and neural classification can support low-dimensional, interpretable, and robust IoT DDoS detection.

The remainder of the paper is structured as follows. Section 2 reviews the related literature on explainable AI in cybersecurity, feature reduction in IoT/IIoT security, and robustness-oriented explanation stability. Section 3 describes the CICIoT2023 dataset, subset construction, data balancing, and preprocessing pipeline. Section 4 presents the proposed integrated S-SFIX–FFNN framework and experimental design, including the FFNN architecture, S-SFIX-based feature reduction, comparative evaluation, and robustness analysis under Gaussian noise. Section 5 reports the experimental results. Section 6 discusses the findings in terms of explainability, dimensionality reduction, classification performance, and resilience. Finally, Section 7 concludes the paper and outlines future research directions.

2 Related Work

Recent research on IoT-based intrusion and DDoS detection has shifted from accuracy-oriented model development toward more transparent, compact, and robust detection frameworks. In cybersecurity applications, especially in IoT environments, the reliability of a detection model depends on whether the underlying feature representation is interpretable, low-dimensional, and stable under data perturbations as well as on its classification performance. Therefore, explainable feature reduction has become an important research direction at the intersection of XAI, intrusion detection, and AI resilience. This section reviews the related literature in three main groups: XAI-based cybersecurity and intrusion detection, feature reduction for IoT/IIoT security, and robustness-oriented explanation stability.

2.1 Explainable AI in Cybersecurity and Intrusion Detection

Recent research has increasingly focused on establishing rigorous frameworks for evaluating XAI within Intrusion Detection Systems (IDS). Arreche et al. [8], for instance, introduced an end-to-end evaluation pipeline to scrutinize SHAP and LIME across diverse environments, including the NSL-KDD, CICIDS-2017, and RoEduNet-SIMARGL2021 datasets. By employing seven distinct machine learning architectures—ranging from RF and AdaBoost to DNN and SVM—their analysis utilized six key metrics: descriptive accuracy, sparsity, stability, efficiency, robustness, and completeness. Their findings highlighted a critical trade-off: while SHAP delivered superior consistency in global explanations, LIME proved computationally more efficient despite being prone to instability and robustness fluctuations. Their work underscored that adversarial model manipulations can severely undermine explanatory integrity, suggesting that XAI reliability remains a concern under domain shifts [8].

The push toward high-performance yet interpretable IoT security is further exemplified by the NGSHAP-IDS framework proposed by Dong et al. [9]. This approach leverages NGBoost, a natural gradient-based probabilistic boosting method, to perform multi-class classification while integrating a SHAP module for granular feature contribution analysis. A distinguishing feature of their work is the inclusion of uncertainty analysis via entropy and maximum probability measures to quantify prediction confidence. Rigorous testing on UNSW-NB15, CICIDS2017, and N-BaIoT datasets yielded impressive results, with NGSHAP-IDS achieving accuracies of 97.9%, 99.8%, and 99.9%, respectively. Beyond raw performance, the authors demonstrated the framework's suitability for IoT hardware, noting a compact model size (1–2 MB) that significantly outperformed heavier architectures like Deep Forest. Their use of SHAP interaction and waterfall analyses provides a roadmap for maintaining transparency and resilience even amidst distribution shifts [9].

Practical constraints in IoT, such as class imbalance and resource-limited edge deployment, were addressed by Hasan and Tasnim [10]. Their holistic IDS architecture incorporates a sophisticated preprocessing pipeline—including kNN imputation and CTGAN-based oversampling—alongside unsupervised feature generation via K-means. Evaluating a broad spectrum of models (from LightGBM to TabNet) on the DS2OS and ToN_IoT datasets, they reported a macro F1-score of 97.15% for the CTGAN-supported XGBoost model on DS2OS, with binary classification performance reaching 100% F1-score in specific ToN_IoT scenarios. For interpretability, they combined global SHAP and local LIME explanations, validating the system's real-time feasibility on a Jetson Nano device with inference times as low as 0.01 seconds. This integration of data balancing and XAI offers a critical perspective on maintaining AI resilience in heterogeneous IoT environments [10].

The broader landscape of these applications is synthesized in the extensive review by Capuano et al. [11], which systematically classifies over 300 studies across domains like malware and botnet detection. Their taxonomy contrasts model-agnostic and post-hoc methods, identifying SHAP as the predominant tool while noting a comparative scarcity of inherently interpretable (intrinsic) models. A pivotal contribution of

their review is the identification of the "transparency paradox" in cybersecurity: while XAI fosters human-machine trust, it may simultaneously expose system vulnerabilities to attackers, thereby expanding the threat surface. They conclude that a significant research gap persists regarding the stability and reliability of explanations in adversarial settings [11].

These studies show that XAI methods such as SHAP and LIME have become widely used in cybersecurity and intrusion detection tasks. However, most of these approaches are mainly employed to explain already trained models rather than to construct an explainable feature reduction mechanism before classification. Therefore, the role of explainability in producing compact and discriminative feature subsets for IoT DDoS detection remains an important research direction.

2.2 Feature Reduction and Low-Dimensional Detection in IoT/IIoT Security

Aiming to achieve both attack detection and proactive intervention in SD-IoT networks, Alotaibi [12] has proposed a hybrid GNN-Transformer and privacy-focused architecture. The study integrates multimodal pre-processing that combines telemetry and traffic flow, feature optimisation based on SHAP and RFE, a GNN component for local topological dependencies, and a Transformer encoder component for global temporal relationships. PPO-based deep reinforcement learning (DRL) is employed during the intervention phase, whilst privacy is ensured through ADP-supported federated learning. In evaluations conducted on the CIC-IoT-2023, ToN-IoT and Edge-IIoTset datasets, remarkable results were achieved on CIC-IoT-2023, including 98.7% accuracy and a 1.2% false alarm rate (FAR). Supported by GJO-based hyperparameter optimisation, this architecture demonstrates resilience against distribution shifts in heterogeneous IoT environments, thereby making a comprehensive contribution to the AI resilience perspective [12].

Gad et al. [13], who examined the dynamic nature of IoT attacks using modern datasets, developed a comprehensive IDS model on the ToN-IoT dataset, which represents VANET environments. Highlighting the inadequacy of traditional datasets such as NSL-KDD in reflecting modern threats, the authors applied SMOTE and Chi-squared-based feature reduction strategies to heterogeneous data comprising 44 features and 9 attack types. Experimental results showed that the XGBoost algorithm demonstrated the best performance in binary classification with 99.1% accuracy, 98.4% sensitivity and 98.7% F1-score; even when the number of features was reduced to 20 using Chi2, 98.3% accuracy was maintained. Although XGBoost maintained its superiority in multi-class scenarios, the study's failure to analyse the model's generalisation capacity under different distribution shifts highlights a gap that needs to be addressed in terms of AI resilience [13].

Prioritising efficiency and interpretability for Industrial IoT (IIoT) networks, Alani [14] developed a flow-based E2I3DS system using the WUSTL-IIOT-2021 dataset. By using RFE during the feature engineering stage to reduce the feature set from 41 to just 11 critical parameters, the author achieved 99.97% accuracy and an F1 score in the Decision Tree model using this low-dimensional configuration. The model's prediction time of just 0.1517 ms per sample represents a critical advantage for real-time IIoT operations. SHAP (TreeExplainer) analyses demonstrated that flow attributes such as DIntPkt, sTtl and Dport are decisive in distinguishing DoS and reconnaissance attacks. This study serves as an important reference for the construction of reliable IDS architectures in resource-constrained environments, as it offers both dimensionality reduction and an explainable framework [14].

Viewed from a general perspective, Alotaibi's hybrid models, Gad's modern dataset-focused approach and Alani's work aimed at high efficiency reinforce the role of XAI methods in cybersecurity. However, the focus of these studies is generally limited to explaining the model's decision post-hoc. At this point, the S-SFIX–FFNN framework we propose aims to fill this gap in the literature by utilising explainability not merely as an analytical tool, but as an 'active reduction mechanism' to directly generate robust and low-dimensional feature subsets.

2.3 Robustness, Explanation Stability, and AI Resilience

Nikolaidis and Doumpos [15], focusing on the "population/concept drift" problem in credit scoring models, have proposed an adaptive classification framework based on sample-specific local competence regions. Experiments conducted on a massive dataset covering the period 2009–2019, comprising 3.52 million observations and 125 variables, demonstrated that the local Logistic Regression model improved the AUC score from 87.29% to 92.56% and the H-Measure value from 49.87% to 63.60% compared to its global counterpart. The authors demonstrated that kNN-based local region selection offers a clear advantage over random

sampling, emphasising that the model's resilience to distribution shifts is dependent on the "local similarity structure". This approach makes a significant contribution to the AI resilience literature by reducing the need for continuous recalibration whilst preserving interpretability [15].

To make the feature selection problem more reliable, Rossi et al. [16] developed the VAENN architecture, a deep learning framework based on variational inference. By directly integrating feature selection into the model-building process, this architecture outperforms traditional techniques in terms of robustness to noise, sparse data handling, and performance in high-dimensional analyses. The study demonstrates that explainable deep learning models should be evaluated not only on accuracy but also on noise robustness and dimensional analysis capability [16].

Cheng et al. [17], who extend the scope of explainability from decision mechanisms to uncertainty analysis, have used the SHAP method not to explain model outputs but to analyse 'variation in prediction uncertainty'. Tested across diverse fields such as housing price forecasting and electrocardiogram classification, this approach has identified that rare feature patterns in training data generate high uncertainty. These findings are critical in demonstrating that explainability must encompass not only the "decision" but also "model confidence" [17].

Finally, Thuy and Benoit [18] have established a comprehensive framework for reliable decision-making processes under distribution shifts by positioning uncertainty estimation as an operational XAI method. Distinguishing uncertainty as 'aleatory' (data-driven) and 'epistemic' (model-driven), the authors have integrated the Monte Carlo Dropout and Deep Ensembles methods. In scenarios without distribution shifts, standard neural networks achieved an accuracy of 63.85%, whilst MC Dropout reached 84.43% and Deep Ensembles 84.93%. Under large distribution shifts, whilst standard models produce "confidently wrong" predictions, it has been reported that the Deep Ensembles method achieves a "non-rejected accuracy" rate of 95% thanks to an uncertainty-based rejection mechanism [18].

This body of literature clearly demonstrates that resilience, robustness and uncertainty management in explainable AI systems are no longer a luxury, but a fundamental requirement. However, specifically in the context of IoT DDoS detection, the stability of feature ranking mechanisms under data corruption has received less attention compared to classification performance. Motivated by this, in our study, the S-SFIX ranking mechanism was tested under Gaussian noise and subjected to an in-depth robustness analysis using score-based, rank-based and Top-k protection metrics.

2.4 Research Gap and Positioning of the Proposed S-SFIX–FFNN Framework

Existing studies have contributed significantly to IoT intrusion detection, explainable AI, feature reduction, and robustness analysis. However, most studies focus either on improving classification performance, explaining already trained models through post-hoc XAI methods, or reducing feature dimensionality using conventional feature selection techniques. Comparatively fewer studies jointly investigate explainable feature reduction, neural detection performance, controlled comparison with alternative XAI/feature importance methods, and the stability of feature rankings under data perturbation. To address this gap, this study proposes an integrated S-SFIX–FFNN framework in which S-SFIX performs explainable feature scoring and reduction, while FFNN evaluates the detection capability of the resulting low-dimensional feature subsets.

3 Dataset and Preprocessing

3.1 CICIoT2023 Dataset

In this study, the CICIoT2023 attack dataset, created by Neto et al. and containing IoT-based network traffic data, was used [19]. The combined dataset contains a total of 46,686,579 instances and 34 different attack labels. The dataset covers various DDoS types, DoS attacks, Mirai variants, reconnaissance attacks, and a range of network-based threat scenarios. It includes a total of 46 features related to network traffic, such as flag counts, duration, variance, packet size statistics, and protocol information. In addition, there are 33 different attack subtypes grouped under the Attack super-class. The overall label distribution is summarized in Table 1.

Table 1. Distribution of the 34 Labels in the Full Dataset

Label	Number of Samples	Percentage (%)
DDoS-ICMP_Flood	7,200,504	15.42

Label	Number of Samples	Percentage (%)
DDoS-UDP_Flood	5,412,287	11.59
DDoS-TCP_Flood	4,497,667	9.63
DDoS-PSHACK_Flood	4,094,755	8.77
DDoS-SYN_Flood	4,059,190	8.69
DDoS-RSTFINFlood	4,045,285	8.66
DDoS-SynonymousIP_Flood	3,598,138	7.71
DoS-UDP_Flood	3,318,595	7.11
DoS-TCP_Flood	2,671,445	5.72
DoS-SYN_Flood	2,028,834	4.35
BenignTraffic	1,098,195	2.35
Mirai-greeth_flood	991,866	2.12
Mirai-udpplain	890,576	1.91
Mirai-greip_flood	751,682	1.61
DDoS-ICMP_Fragmentation	452,489	0.97
MITM-ArpSpoofing	307,593	0.66
DDoS-UDP_Fragmentation	286,925	0.61
DDoS-ACK_Fragmentation	285,104	0.61
DNS_Spoofing	178,911	0.38
Recon-HostDiscovery	134,378	0.29
Recon-OSScan	98,259	0.21
Recon-PortScan	82,284	0.18
DoS-HTTP_Flood	71,864	0.15
VulnerabilityScan	37,382	0.08
DDoS-HTTP_Flood	28,790	0.06
DDoS-SlowLoris	23,426	0.05
DictionaryBruteForce	13,064	0.03
BrowserHijacking	5,859	0.01
CommandInjection	5,409	0.01
SqlInjection	5,245	0.01
XSS	3,846	0.01
Backdoor_Malware	3,218	0.01
Recon-PingSweep	2,262	0.00
Uploading_Attack	1,252	0.00
Total	46,686,579	100

3.2 Subset Construction and Data Balancing

3.2.1 Subset Construction

Although subset selection was necessary due to computational constraints, it was carried out from among homogeneous partitions of the same dataset in order to preserve the structural integrity of the data generation process.

The dataset processed in the experimental workflow consists of a total of 470,500 samples, and these samples are processed through two CSV files.

In the experimental design, the numerous attack types in the dataset were merged under a single super-class labeled as Attack, and a binary classification problem was defined together with the BenignTraffic label. In this way, a more comprehensive and challenging experimental setup was obtained, incorporating multiple attack patterns simultaneously. This approach makes it possible to evaluate the proposed S-SFIX-based feature reduction framework over a broader threat space. In the training pipeline used, the labels are also remapped as BenignTraffic $\rightarrow$ 0 and all other attack types as Attack $\rightarrow$ 1.

As shown in Table 2, the Attack super-class consists of a total of 33 different attack subtypes. These subtypes exhibit a highly heterogeneous structure, ranging from high-volume DDoS and DoS attacks to Mirai-based patterns, and from reconnaissance types to spoofing, brute force, and injection-based attacks. In particular, intensive attack types such as DDoS-ICMP_Flood, DDoS-UDP_Flood, DDoS-TCP_Flood, and DDoS-SYN_Flood occupy a significant portion of the dataset, while attacks with relatively fewer samples,

such as Recon-PingSweep, Uploading_Attack, XSS, and Backdoor_Malware, are also included within the same super-class. This shows that, on the contrary, the dataset provides a more challenging and more realistic experimental scenario by including a large number of attack behaviors with differing characteristics within the same structure.

Table 2. Sorted distribution of attack subtypes under the Attack super-class in descending order

Attacks	**Samples**
DDoS-ICMP_Flood	36554
DDoS-UDP_Flood	27626
DDoS-TCP_Flood	23149
DDoS-PSHACK_Flood	21210
DDoS-SYN_Flood	20739
DDoS-RSTFINFlood	20669
DDoS-SynonymousIP_Flood	18189
DoS-UDP_Flood	16957
DoS-TCP_Flood	13630
DoS-SYN_Flood	10275
Mirai-greeth_flood	5016
Mirai-udpplain	4661
Mirai-greip_flood	3758
DDoS-ICMP_Fragmentation	2377
MITM-ArpSpoofing	1614
DDoS-ACK_Fragmentation	1505
DDoS-UDP_Fragmentation	1484
DNS_Spoofing	925
Recon-HostDiscovery	697
Recon-OSScan	517
DoS-HTTP_Flood	414
Recon-PortScan	430
VulnerabilityScan	210
DDoS-HTTP_Flood	169
DDoS-SlowLoris	106
DictionaryBruteForce	63
SqlInjection	31
BrowserHijacking	30
CommandInjection	28
Backdoor_Malware	22
XSS	18
Uploading_Attack	8
Recon-PingSweep	6

3.2.2 Imbalance Mitigation Strategy

Since the full CICIoT2023 dataset is too large to be processed at once in terms of system resources, the experimental study was conducted on a subset that preserves the homogeneous structure of the dataset. In the first stage, one of the homogeneous partitions of the dataset was taken as the basis; however, a clear imbalance between the Benign and Attack classes was observed in this subset. To reduce this imbalance, only benign samples were filtered from the other CSV partitions that make up the dataset and were then added in a controlled manner to the initially selected working subset. In this way, the dataset was prevented from being supplemented by an entirely different source, while the Benign class was strengthened using homogeneous subfiles belonging to the same data generation process, thereby improving the class distribution.

The main purpose of this approach is to reduce class imbalance while preserving the statistical structure of the dataset as much as possible, and to prevent the model from turning into a one-sided structure that learns only the dominant attack patterns. In other words, dataset balancing was not performed through external or synthetic sample generation, but rather through the controlled addition of benign samples selected from homogeneous subparts of the same dataset. This choice is important in that it preserves the

representativeness of the dataset while providing a more realistic learning environment.

In addition to the imbalance mitigation process, a class weight approach was also used to reduce the effect of the remaining class imbalance during training. Class weights were calculated solely on the basis of the benign and attack distribution in the training set and were incorporated into the training process. In this way, information from the validation and test sets was prevented from leaking into the training phase, while at the same time the negative impact of class dominance on learning was reduced. This implementation was explicitly defined in the training pipeline.

3.3 Training, Validation, and Test Split

The data processing pipeline was designed in a way that prevents data leakage and allows the experimental results to be reported transparently. The data were first divided into training, validation, and test sets; in practice, this split corresponded approximately to 72% training, 8% validation, and 20% test. In the current experimental setup, the training set contains 338,759 samples, the validation set contains 37,640 samples, and the test set contains 94,101 samples. Thus, training and evaluation were performed on a total working subset of 470,500 samples.

During the split process, the labels were converted into a binary structure, and stratified splitting was applied whenever possible in order to preserve class proportions. In the code workflow, the test set is first separated, and then the training pool is further divided into training and validation subsets. This structure ensures that the test data are never seen during the model training process. In addition, the scaling process is learned only on the training set, and the same transformation parameters are applied to the validation and test sets. This approach is important for preventing data leakage and ensuring reliable performance evaluation.

Table 3 presents the class distribution across the training, validation, and test subsets. The results show that the Benign and Attack classes are represented in very similar proportions across all subsets. In the training set, the number of Benign samples is 170,937, while the number of Attack samples is 167,822. Similarly, the preservation of class proportions in the validation and test sets indicates that the data splitting procedure was carried out consistently in terms of class distribution. In the total working subset, the Benign ratio is approximately 50.46%, whereas the Attack ratio is 49.54%. This result shows that the class imbalance initially observed was largely reduced through the controlled addition of benign samples from homogeneous subparts of the same dataset, and that the model was trained in a more balanced learning environment.

Table 3. Class distribution in the training, validation, and test sets

Subset	Benign	Attack	Total
Train	170,937	167,822	338,759
Validation	18,993	18,647	37,640
Test	47,483	46,618	94,101
Total	237,413	233,087	470,500

All experiments were conducted using the same train-test split and the same FFNN architecture; only the feature reduction strategy was changed. In feature reduction, the importance scores produced by Simplified SFIX (S-SFIX) were used, and the following scenarios were constructed:

- Top-5 features
- Top-10 features
- Top-15 features
- Top-20 features
- All features (Full Features)

This structure made it possible to systematically analyze the effect of S-SFIX-based feature reduction on both model performance and model complexity (parsimony).

3.4 Preprocessing

Table 4 summarizes step by step the data preparation process followed in this study. In this process, first, a homogeneous subset of the CICIoT2023 dataset was selected due to system constraints, and then the problem structure was redefined as a binary classification task consisting of Benign and Attack classes. By merging multiple attack subtypes under the Attack class, a broader threat space was constructed, while the benign class was supported with samples carefully drawn from other homogeneous parts of the same dataset. In the final stage, the data were divided into training, validation, and test subsets; scaling and class weighting operations were applied only on the basis of the training data, thereby preventing data leakage.

Table 4. Stages of the experimental data preparation process

No	Stage	Description
1	Selection of the initial subset	Since the full CICIoT2023 dataset could not be processed at once due to system resource limitations, a homogeneous subset preserving the structural integrity of the data generation process was selected as the initial working set.
2	Construction of the binary label structure	The BenignTraffic label in the dataset was mapped to the Benign class, while all remaining attack labels were mapped to a single Attack super-class. Thus, the problem was transformed into a binary classification structure covering multiple attack subtypes.
3	Merging attack subtypes	The 33 different attack subtypes in the dataset were grouped under a single Attack class, yielding a broader threat space instead of a narrow experimental setup based on only one attack type.
4	Balancing with benign samples	Since a clear imbalance between the Benign and Attack classes was observed in the initially selected subset, only benign samples were filtered from other homogeneous CSV partitions of the same dataset and added to the working set in a controlled manner.
5	Train/validation/test split	The prepared dataset was divided into training, validation, and test subsets for model training and fair evaluation. In this process, it was ensured that the test set was never seen during the training stage.
6	Scaling and leakage prevention	Feature scaling was learned only on the training set, and the same transformation parameters were applied to the validation and test sets. In this way, data leakage was prevented and the reliability of performance evaluation was improved.
7	Management of class imbalance during training	In order to reduce the effect of the remaining class imbalance in the dataset, class weights were calculated only on the basis of the training set and were incorporated into the learning process during model training.

4 Proposed Integrated S-SFIX–FFNN Framework and Experimental Design

The main objective of the proposed experimental design is to evaluate the integrated S-SFIX–FFNN framework in terms of explainable feature reduction, low-dimensional detection performance, comparative effectiveness, and ranking robustness. In this framework, S-SFIX operates as the explainable feature scoring and reduction component, while FFNN serves as the neural detection component used to assess the classification capability of the reduced feature subsets. The experimental design was constructed to determine whether compact feature subsets identified by S-SFIX can preserve IoT DDoS detection performance while maintaining interpretability and robustness under data perturbation.

Stage 1 – Full-Feature FFNN Baseline

In the first stage, an FFNN model was trained using all 46 features of the CICIoT2023 dataset. This full-feature model served as the baseline for evaluating the effect of feature reduction on detection performance.

Stage 2 – S-SFIX-Based Explainable Feature Reduction

In the second stage, S-SFIX was applied to compute multi-component feature scores and rank the input features according to their explanatory and discriminative contribution. Based on this ranking, the top 5, 10, 15, and 20 features were selected to construct low-dimensional feature subsets.

The purpose of this design is to test to what extent the most important features identified by an explainability-based method can preserve model accuracy.

Stage 3 – FFNN Classification with Reduced Feature Subsets

In the third stage, separate FFNN models were trained using the reduced feature subsets generated by S-SFIX. The same FFNN architecture, preprocessing pipeline, train-validation-test split, and hyperparameters were preserved across all scenarios to ensure a fair evaluation of the selected feature subsets.

Stage 4 - Comparative Evaluation with SHAP, LIME, and Random Forest

In the fourth stage, the proposed S-SFIX–FFNN configuration was compared with alternative feature reduction configurations based on SHAP, LIME, and Random Forest feature importance. For each method, the top 5, 10, 15, and 20 features were selected, and the resulting feature subsets were evaluated using the same FFNN classification setting.

Stage 5 - Robustness Analysis of S-SFIX Rankings Under Gaussian Noise

In the fifth stage, the robustness of the S-SFIX feature ranking mechanism was examined under Gaussian noise. Different levels of noise were added to the numerical features, S-SFIX scores were recalculated, and the resulting rankings were compared with the clean-data ranking using score-based, rank-based, and Top-k preservation metrics.

The noise levels used in the experiments were determined as 0.00 (clean data), 0.01, 0.03, 0.05, 0.10, and 0.20.

At this stage, rank Spearman correlation, Kendall Tau coefficient, mean absolute rank shift, maximum rank shift, and Jaccard similarity for Top-5, Top-10, Top-15, and Top-20 were used to evaluate robustness [20, 21, 22, 23, 24].

Figure 1 presents the workflow of the proposed integrated S-SFIX–FFNN framework. In this framework, S-SFIX serves as the explainable feature reduction module by producing feature rankings based on DIS, CS, AS, and RS components, while FFNN serves as the detection module for evaluating the classification performance of full and reduced feature sets. The framework further includes a controlled comparison with SHAP-, LIME-, and Random Forest-based feature reduction configurations and a robustness analysis of S-SFIX rankings under Gaussian noise. Thus, the proposed pipeline jointly addresses explainability, low-dimensional representation, detection performance, and ranking stability in IoT DDoS detection.

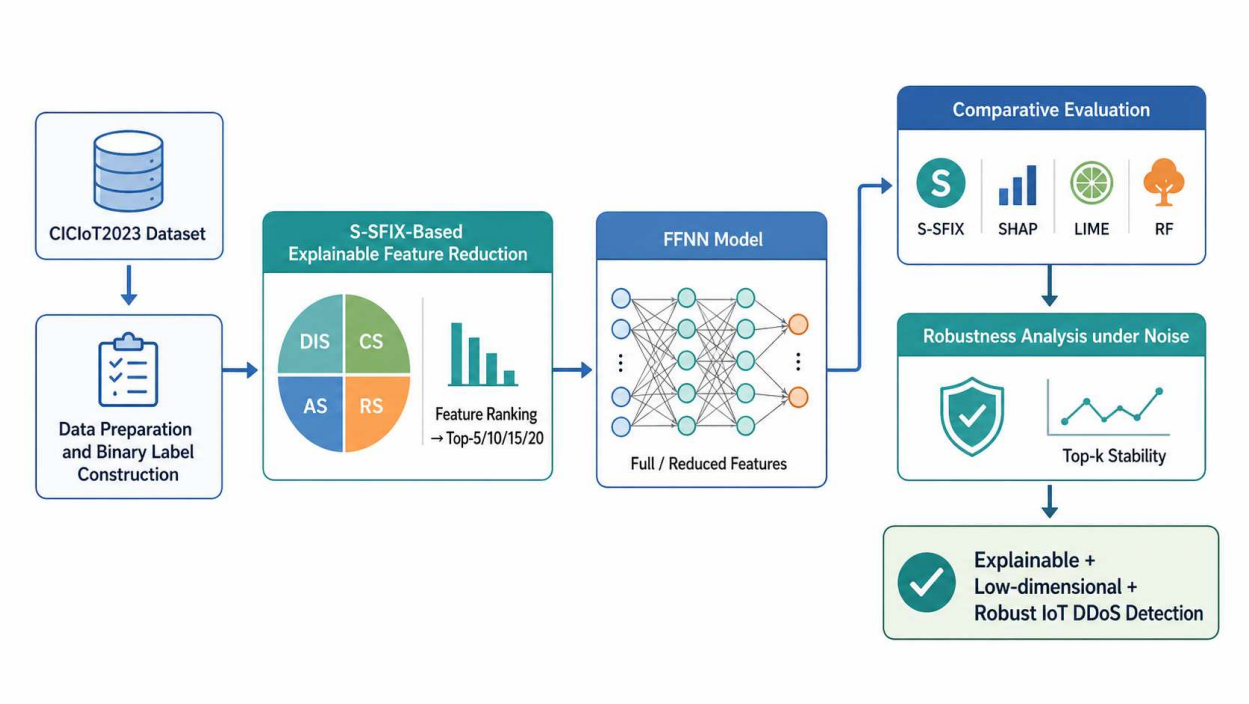


Fig. 1. Workflow of the proposed integrated S-SFIX–FFNN framework for explainable and robust feature reduction in IoT DDoS detection. S-SFIX performs explainable feature scoring and reduction, while FFNN evaluates the detection performance of full and reduced feature sets. The framework also includes comparative evaluation against SHAP, LIME, and Random Forest, and robustness analysis under Gaussian noise.

4.1 FFNN Detection Component

A feed-forward fully connected neural network (FFNN) was used as the detection component for evaluating the classification performance of full and reduced feature sets. The model consists of an input layer, three hidden layers, and an output layer. The network structure has a classical forward-directed (feed-forward) architecture without feedback [25].

4.1.1 Network Architecture

The number of neurons in the input layer of the model is equal to the number of features used. Within the scope of the experimental design, five different feature sets were used:

- Top-5 S-SFIX features (5 input neurons)
- Top-10 S-SFIX features (10 input neurons)
- Top-15 S-SFIX features (15 input neurons)
- Top-20 S-SFIX features (20 input neurons)
- All features (46 input neurons)

In each experimental scenario, the same network architecture was preserved, and only the number of neurons in the input layer was changed.

The hidden layer structure consists of three layers, and each hidden layer contains 50 neurons. The sigmoid function was used as the activation function in the hidden layers. The transformation for each hidden layer is expressed as follows:

$$g(x) = \sigma(xW + b) \tag{1}$$

Here, x denotes the input vector, W the weight matrix, b the bias term, and $\sigma(\cdot)$ the sigmoid activation function. The sigmoid function is defined as follows:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{2}$$

The sigmoid function provides a controlled and nonlinear transformation between layers by mapping neuron outputs to the range between 0 and 1.

Since a binary classification problem (Benign / Attack) is addressed in the output layer, a single neuron was used. The sigmoid function was again preferred as the activation function:

$$\hat{y} = \sigma(hW_o + b_o) \tag{3}$$

Here, h represents the output of the last hidden layer, W_o denotes the weights of the output layer, and b_o represents the output bias term. This structure produces the probability that each sample belongs to the Attack class.

4.1.2 Loss Function and Optimization

During the training of the model, binary cross-entropy was used as the loss function:

$$L = -y \log(\hat{y}) - (1 - y) \log(1 - \hat{y}) \tag{4}$$

Here, y denotes the true class label, while $\hat{y}$ represents the probability predicted by the model.

Adamax was used as the optimization algorithm. Adamax is an infinity norm-based variant of the Adam algorithm and exhibits more stable convergence behavior in high-dimensional data spaces. The weight update process was carried out iteratively.

4.1.3 FFNN Hyperparameters and Rationale for Their Selection

The FFNN model used in this study was designed to evaluate the feature subsets obtained from different feature reduction methods within a common and fixed classification environment. Rather than proposing a new deep learning architecture, the aim is to ensure a fair comparison of S-SFIX-, SHAP-, LIME-, and Random Forest-based feature reduction approaches under the same classification framework. For this reason, the FFNN model was used not as an independent variable in the experiments, but as a common evaluation infrastructure kept constant for all methods.

The FFNN model consists of three hidden layers, each containing 50 neurons. This three-layer structure provides sufficient representational capacity to learn nonlinear patterns in the dataset, while also making it possible to avoid an excessively deep and computationally costly architecture. The choice of 50 neurons per layer was made in order to preserve model capacity while preventing the differences between feature reduction methods from being masked by a very large network architecture. In other words, the aim here is not to build the most complex network possible, but to use a balanced classifier that makes the effect of different feature reduction strategies visible.

The use of the sigmoid activation function in the hidden layers was preferred in order to ensure that the model outputs are transformed in a more controlled and stable manner. Since the main objective of the study is not deep architecture optimization, the intention was to establish a fixed experimental basis using a widely adopted and stable activation function from the literature. In the output layer, a single-neuron sigmoid structure was used in accordance with the binary classification problem. Thus, the model directly produces the probability that each sample belongs to the attack class.

Adamax was selected as the optimization algorithm. Adamax was preferred because it uses an adaptive learning rate and exhibits stable optimization behavior, especially in large-scale data structures. Considering the size of the dataset and the fact that the experiments were repeated across many different feature subsets, the aim was to use an optimizer that provides both stable convergence and practical ease of implementation during training. In this context, Adamax offers an appropriate choice that preserves the same learning dynamics across all comparative experiments.

The batch size was set to 2048. The main reason for this choice is that the dataset used is large-scale and the training process was intended to be completed within a reasonable amount of time. Smaller batch size values may lead to longer training times for each epoch and can substantially increase the total computational cost, especially in comparative experiments repeated for all methods. For this reason, the value of 2048 was

preferred in order to increase training speed, reduce computational cost, and at the same time maintain sufficient learning stability.

The number of epochs was fixed at 10. Considering the size of the dataset, each epoch already includes a very large number of samples. Therefore, instead of using very high epoch values, a training duration was preferred that would be sufficient for the model to learn the main patterns while not excessively increasing the cost of comparative experiments. In addition, since the purpose of the study is not to achieve the highest absolute FFNN accuracy but rather to compare the relative effect of different feature reduction methods, preserving the same number of epochs for all methods was considered methodologically more appropriate.

4.2 S-SFIX Component of the Integrated Framework

The Scalable Financial-oriented Interpretable eXplanation (SFIX) model developed by Cil and Yildiz was used as the explainability layer [7]. The SFIX model is a time-sensitive, risk-aware, and multi-component XAI framework specifically designed for financial decision support systems. The model consists of five main components: Dynamic Importance Score (DIS), Confidence Score (CS), Pattern Score (PS), Anomaly Score (AS), and Risk Score (RS). These components are combined through a multiplicative structure to calculate the overall explainability score of each feature.

However, since the dataset used in this study does not contain an explicit time-series dimension, the simplified version of the SFIX model that does not include the time parameter, namely the Simplified SFIX (S-SFIX) approach, was preferred. S-SFIX was specifically designed by Cil and Yıldız [7] for rapid model testing and feature reduction in datasets where the real-time dimension is absent or limited.

Within the scope of S-SFIX, the following components were taken into consideration:

- **Dynamic Importance Score (DIS):** The formulation of the Dynamic Importance Score is as follows:

$$DIS(f) = BI(f) \cdot II(f) \tag{5}$$

  - **Basic Impact Score (BI):** The linear relationship between the feature and the target variable was calculated using Pearson correlation.
  - **Interaction Impact Score (II):** Monotonic relationships among features were measured using Kendall's Tau coefficient.

- **Confidence Score (CS):** Data quality (missing data ratio, outlier ratio, coefficient of variation) and model consistency were taken into account.

- **Anomaly Score (AS):** Z-score-based deviation analysis was applied.

- **Risk Score (RS):** A Bayes' Theorem-based probabilistic risk assessment was used.

The overall S-SFIX score was calculated as follows:

$$XAI_Score(f) = DIS(f) \cdot CS(f) \cdot (1 + AS(f)) \cdot RS(f) \tag{6}$$

This score provides a holistic explainability metric that jointly evaluates the statistical significance of each feature, its contribution to the model, data quality, and the risk dimension.

In the proposed framework, S-SFIX is not used as a standalone explanation method; rather, it functions as the feature scoring and reduction layer that generates compact feature subsets for the FFNN-based detection component. This integration enables the framework to evaluate feature importance, dimensionality reduction, classification performance, and robustness within a unified experimental pipeline.

4.3 S-SFIX Training and Feature Ranking

In the first stage of proposed framework, the Simplified SFIX (S-SFIX) algorithm was trained using a subset of the filtered dataset. S-SFIX produces four components for each feature:

- DIS (Dynamic Importance Score)
- CS (Confidence Score)
- AS (Anomaly Score)

- RS (Risk Score)

These components were combined, and the final S-SFIX score ($final_score$) for each feature was calculated in Table 5.

Table 5. S-SFIX component scores and final feature scores under the clean data scenario

Feature	DIS	CS	AS	RS	Final Score
Variance	0.135935	0.128765	0.938962	0.063389	0.002151
Magnitue	0.098970	0.168156	0.967091	0.053460	0.001750
Max	0.148827	0.104572	0.986363	0.039884	0.001233
Duration	0.048785	0.400740	0.982228	0.026200	0.001015
AVG	0.106227	0.100267	0.970204	0.041329	0.000867
Tot size	0.104599	0.099450	0.969328	0.041343	0.000847
Tot sum	0.102647	0.094919	0.972567	0.038593	0.000742
Std	0.125253	0.066885	0.964162	0.043584	0.000717
Radius	0.125289	0.066746	0.964187	0.043553	0.000715
ack_flag_number	0.055540	0.088194	1.000000	0.069242	0.000678
ICMP	0.015887	0.429828	0.996978	0.018373	0.000250
HTTPS	0.073661	0.071805	1.000000	0.020072	0.000213
syn_flag_number	0.021676	0.057371	1.000000	0.061003	0.000152
rst_count	0.021349	0.057970	0.998465	0.053374	0.000131
Header_Length	0.028756	0.015627	0.968213	0.084737	0.000074
Min	0.005731	0.024258	0.993935	0.092672	0.000026
rst_flag_number	0.001255	0.091099	0.999998	0.127145	0.000023
fin_flag_number	0.001133	0.090268	1.000000	0.125214	0.000021
TCP	0.000114	0.012245	1.000000	0.545419	0.000002
syn_count	0.000147	0.002313	0.999969	0.285386	0.000000
urg_count	0.000040	0.000626	1.000000	0.420906	0.000000
Covariance	0.000002	0.005299	0.931663	0.212584	0.000000
UDP	0.000001	0.000011	1.000000	0.991828	0.000000
psh_flag_number	0.000001	0.000007	1.000000	0.990389	0.000000
Protocol Type	0.000000	0.000210	0.640544	0.762364	0.000000
fin_count	0.000000	0.000000	1.000000	0.506001	0.000000
ack_count	0.000000	0.000000	1.000000	0.502326	0.000000
Rate	0.000000	0.000000	1.000000	0.441340	0.000000
Srate	0.000000	0.000000	1.000000	0.441221	0.000000
IPv	0.000000	0.000000	0.987566	0.568212	0.000000
LLC	0.000000	0.000000	0.997129	0.505295	0.000000
flow_duration	0.000000	0.000000	0.998198	0.450683	0.000000
HTTP	0.000000	0.000000	1.000000	0.337118	0.000000
DNS	0.000000	0.000000	0.998863	0.338484	0.000000
ARP	0.000000	0.000000	0.998250	0.339285	0.000000
IAT	0.000000	0.000000	0.946673	0.455902	0.000000
SSH	0.000000	0.000000	1.000000	0.333334	0.000000
Drate	0.000000	0.000000	1.000000	0.333334	0.000000
Weight	0.000000	0.000000	0.866445	0.500041	0.000000
Number	0.000000	0.000000	0.979380	0.406873	0.000000
DHCP	0.000000	0.000000	1.000000	0.333334	0.000000
IRC	0.000000	0.000000	1.000000	0.333334	0.000000
Telnet	0.000000	0.000000	1.000000	0.333334	0.000000
cwr_flag_number	0.000000	0.000000	1.000000	0.333334	0.000000
ece_flag_number	0.000000	0.000000	1.000000	0.333334	0.000000
SMTP	0.000000	0.000000	1.000000	0.333334	0.000000

Table 5 presents the DIS, CS, AS, RS, and final score values calculated by S-SFIX for each feature. The results reveal that features such as Variance, Magnitue, Max, Duration, AVG, and Tot size are ranked

among the top in terms of the final S-SFIX score. In particular, the fact that the Variance and Magnitue features have the highest final scores indicates that these features play a strong role in distinguishing attack and benign classes in terms of both discriminativeness and consistency. In contrast, it is observed that many features listed in the lower part of the table have final scores that are very close to zero. This indicates that these features provide limited discriminative power between classes under the clean data scenario and are assessed by S-SFIX as having a low level of importance. Overall, the table shows that the proposed method highlights only a few dominant features, while the remaining features make relatively lower contributions.

Figure 2 presents both the component-level scores and the final S-SFIX scores of the top 10 features identified by S-SFIX as having the highest importance. In the upper part, the contributions of the DIS, CS, AS, and RS components are presented separately for each feature, while the lower part shows the final importance scores obtained through the combination of these components. This structure demonstrates that S-SFIX does not rank features based on a single criterion only; rather, it uses a multidimensional evaluation framework that jointly considers discriminativeness, consistency, anomaly sensitivity, and risk-based separability.

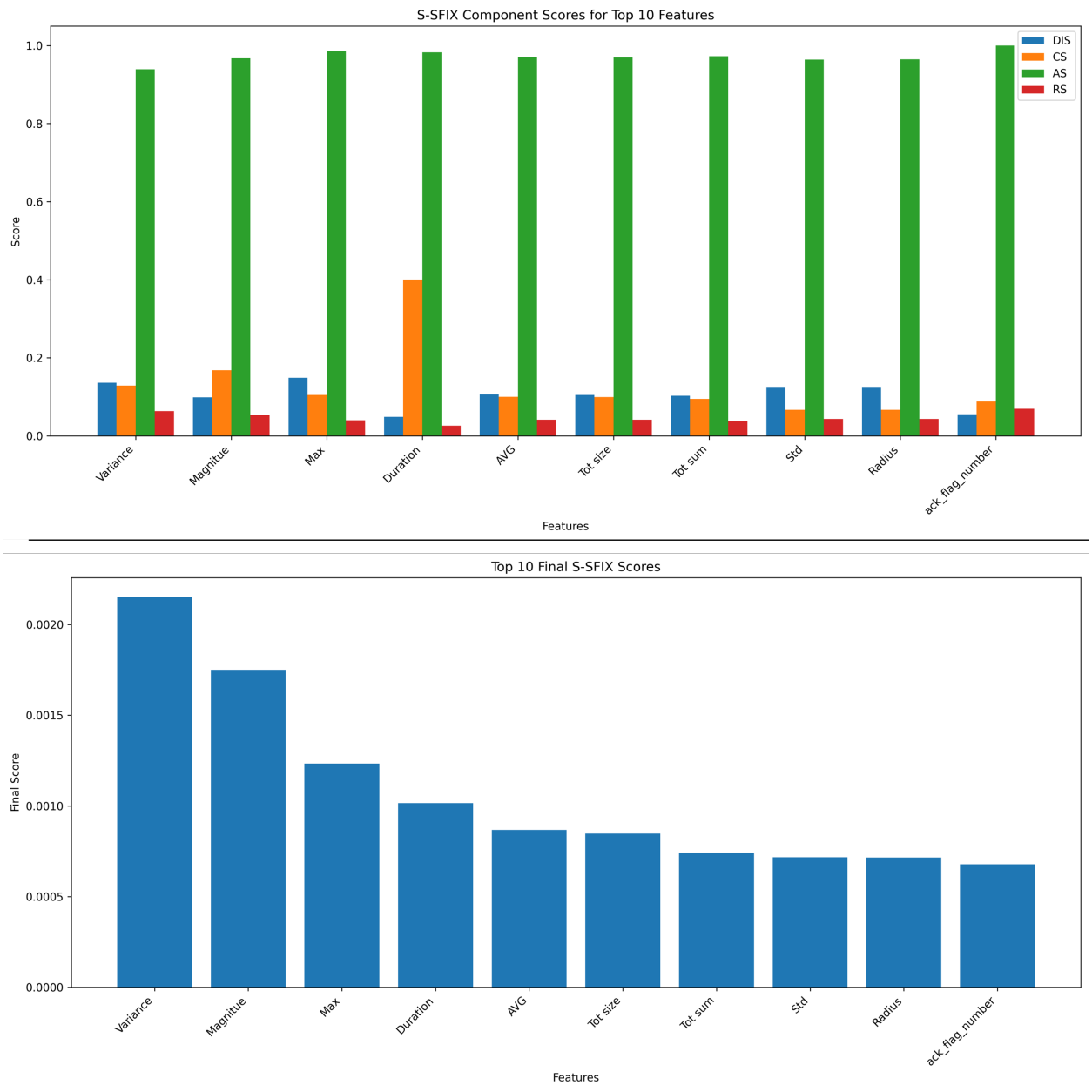


Fig. 2. S-SFIX component scores (DIS, CS, AS, RS) and final S-SFIX feature ranking. The upper panel shows the component scores, while the lower panel presents the distribution of final scores.

When Figure 2 is examined, it can be seen that the variables included among the top 10 features exhibit different profiles in terms of their component scores. While some features stand out through high discriminativeness and consistency, others are supported more by risk-based separability or anomaly sensitivity components. This indicates that the final S-SFIX score is not formed on the basis of a single dominant component, but rather through the combined effect of multiple components. The final scores presented in the lower part also clearly reveal which features are included in the core importance set as a result of this multi-component structure.

This figure demonstrates that S-SFIX evaluates feature importance not through a one-dimensional feature importance approach, but through a multi-component and more holistic scoring structure.

4.4 Experimental Design of S-SFIX Robustness Under Gaussian Noise

In this study, the stability of the proposed S-SFIX method under random data perturbations was also examined. The main purpose of this section is to directly evaluate **to what extent the feature scores and rankings produced by S-SFIX are preserved under noise**. For this reason, the experimental design was established in a way that focuses on the internal consistency and robustness of S-SFIX itself before model performance.

For this purpose, **Gaussian noise** at different severity levels was added to the numerical features of the dataset, and the S-SFIX method was rerun for each perturbation level. The noise levels used in the experimental workflow were determined as **0.00, 0.01, 0.03, 0.05, 0.10, and 0.20**, respectively. Here, the 0.00 level represents the reference clean data scenario, while the other levels represent gradually increasing perturbation scenarios. Noise was applied only to the numerical features, while the label column and target structure were preserved.

A single fixed standard deviation was not used for all features in noise generation. Instead, the perturbation level for each feature was defined based on the standard deviation of the corresponding feature. In other words, for each numerical feature, the perturbed values were generated as

$$x' = x + \mathcal{N}(0, (\alpha \cdot \sigma_j)^2)$$

where α denotes the corresponding noise level and σ_j denotes the standard deviation of the jth feature on the clean data. In this way, noise proportional to their own variation structure was applied to features distributed on different scales. In addition, if desired, the numerical values after noise injection were clipped to the min–max range observed in the clean data in order to prevent extreme synthetic deviations.

For each noise level, S-SFIX was rerun on the same dataset, and the DIS, CS, AS, RS, and final_score values were recalculated for each feature. The implementation used was designed to remain faithful to the original simplified formulation of S-SFIX. Within this structure, DIS represents the discriminativeness component, CS represents the consistency component, AS represents anomaly sensitivity, and RS represents risk-based separability; the final score is obtained through the combination of these components.

In order to measure the change between the clean data and noisy data scenarios, both score-based and ranking-based robustness metrics were used. In the score-based evaluation, score Pearson correlation, mean absolute score difference, and maximum absolute score difference were calculated to measure the relationship between the final_score values obtained under the clean and perturbed data scenarios. In the ranking-based evaluation, the feature rankings obtained under clean and noisy data were compared; for this purpose, Spearman rank correlation and Kendall Tau coefficient were used. In addition, mean absolute rank shift and maximum rank shift were calculated to show how much each feature changed its position with respect to its ranking in the clean data.

Since the preservation of top-ranked feature sets was considered especially important in this study, a Top-k-based similarity analysis was also conducted. For this purpose, for $k = 5, 10, 15, 20$, the top-ranked feature sets obtained under the clean data scenario were compared with their counterparts under noise, and Jaccard similarity was calculated. In addition, the number of features that remained within the same Top-k sets after noise among those that were originally in the Top-k under clean data conditions was reported using *retention count*. Thus, both the overall ranking correlation and the practically most critical question of whether the "most important features are preserved" were directly addressed.

Through this experimental design, the AI resilience concepts discussed in the introduction are operationalized and evaluated directly at the S-SFIX ranking level. Thus, these concepts have been made directly measurable experimentally for the proposed method.

Table 6. S-SFIX stability experimental setup under Gaussian noise

No	Component	Description
1	Clean data scenario	Noise-free data were used for the reference ranking.
2	Noise levels	0.00, 0.01, 0.03, 0.05, 0.10, and 0.20
3	Noise application scope	Numerical features only
4	Noise generation	Standard deviation-based Gaussian perturbation for each feature
5	Rerunning S-SFIX	Scores were recalculated for each noise level
6	Score-based metrics	Score Pearson, mean and maximum absolute score difference
7	Ranking-based metrics	Rank Spearman, Kendall Tau, mean and maximum rank shift
8	Top-k stability	Jaccard and retention count for Top-5, Top-10, Top-15, and Top-20
9	Visual outputs	Rank correlation, rank shift, Top-k Jaccard, and top-20 score comparisons

Table 6 summarizes the main components of the S-SFIX robustness experiment conducted under Gaussian noise. This design strengthens the AI resilience dimension of the study by focusing directly on the stability of the feature ranking mechanism.

5 Results

In this section, the impact of the proposed S-SFIX–FFNN framework on classification performance under different feature reduction scenarios is analyzed.

Table 7 presents the performance results obtained using the full feature set and the reduced feature subsets generated by S-SFIX, SHAP, LIME, and Random Forest feature importance. As expected, the FFNN model trained with all 46 features achieved the highest overall performance. However, the results show that substantial feature reduction can be achieved with only limited performance degradation.

In particular, SHAP- and S-SFIX-based configurations produced results very close to the full-feature FFNN baseline, especially at the 15- and 20-feature levels. LIME also achieved stable and high performance at the 10-, 15-, and 20-feature levels, although its performance decreased noticeably in the 5-feature scenario. By contrast, Random Forest feature importance-based reduction produced the weakest results at lower feature counts, particularly in the 5- and 10-feature configurations. Overall, although classification performance generally improved as the number of selected features increased, the results indicate that SHAP and S-SFIX were able to preserve high detection performance even with reduced feature subsets. This finding supports that the proposed S-SFIX–FFNN framework can provide effective explainable feature reduction while maintaining competitive detection performance in a low-dimensional representation space.

Table 7. FFNN performance results under different feature reduction methods and numbers of features

Method	Number of Features	Accuracy	Benign Precision	Recall	Attack Precision	Attack Recall	Macro F1 Score
Full	46	0.9845	0.9734	0.9966	0.9965	0.9722	0.9845
LIME	5	0.9600	0.9576	0.9634	0.9625	0.9565	0.9600
LIME	10	0.9819	0.9670	0.9983	0.9982	0.9652	0.9819
LIME	15	0.9816	0.9674	0.9971	0.9969	0.9658	0.9816
LIME	20	0.9821	0.9669	0.9986	0.9985	0.9652	0.9820
RF	5	0.9525	0.9543	0.9515	0.9507	0.9536	0.9525
RF	10	0.9588	0.9502	0.9691	0.9679	0.9483	0.9588
RF	15	0.9734	0.9559	0.9931	0.9926	0.9533	0.9733
RF	20	0.9738	0.9571	0.9925	0.9921	0.9547	0.9738
S-SFIX	5	0.9689	0.9528	0.9872	0.9865	0.9502	0.9689
S-SFIX	10	0.9696	0.9562	0.9849	0.9841	0.9541	0.9696
S-SFIX	15	0.9793	0.9661	0.9939	0.9936	0.9645	0.9793
S-SFIX	20	0.9805	0.9673	0.9949	0.9947	0.9657	0.9804
SHAP	5	0.9785	0.9621	0.9967	0.9965	0.9600	0.9785
SHAP	10	0.9767	0.9678	0.9867	0.9861	0.9666	0.9767
SHAP	15	0.9823	0.9673	0.9987	0.9986	0.9656	0.9823
SHAP	20	0.9823	0.9681	0.9978	0.9977	0.9665	0.9823

When the comparison across methods is examined in more detail, it is observed that in the 5-feature scenario the highest accuracy value was obtained with SHAP, whereas the lowest value was obtained with Random Forest feature importance. At the 10-feature level, LIME produced the highest result, while at the 15- and 20-feature levels SHAP- and S-SFIX-based selections emerged as the most prominent approaches. In particular, the fact that the S-SFIX-20 and SHAP-20 results are very close to the full model indicates that these two methods are able to keep information loss low with a more limited feature set. In addition,

the fact that the S-SFIX-15 and S-SFIX-20 results are higher than all Random Forest-based configurations and the LIME-5 result supports that the proposed method provides a stronger feature ranking than classical importance-based selection.

5.1 Performance Evaluation

When Table 7 is examined, it can be observed that classification performance does not decrease dramatically despite the substantial reduction in the number of input features. In particular, the S-SFIX–FFNN configurations produced high accuracy and macro F1 values even with lower-dimensional feature subsets. While the full-feature FFNN baseline achieved 98.45% accuracy using all 46 features, the S-SFIX–FFNN configuration achieved 98.05% accuracy using only the top 20 S-SFIX features and 97.93% accuracy using the top 15 features. This finding shows that the proposed framework can successfully prioritize compact feature subsets that retain most of the discriminative information required for the classification task.

Reducing the feature dimension from 46 to 20 corresponds to an approximately 56.5% reduction, while reducing it from 46 to 15 corresponds to an approximately 67.4% reduction. Despite this substantial reduction, the performance loss remains limited, indicating that S-SFIX brings forward features with high discriminative power before the FFNN-based detection stage. Even under more aggressive dimensionality reduction scenarios, the S-SFIX–FFNN configurations using 10 and 5 features achieved 96.96% and 96.89% accuracy, respectively. These results support the effectiveness of the proposed framework in low-dimensional representation spaces.

In the comparison across methods, the S-SFIX–FFNN framework produced stronger results than the Random Forest feature importance-based configuration, especially at lower feature counts. In addition, it achieved higher and more balanced performance than the LIME-based 5-feature configuration. Although SHAP achieved the highest absolute results in the 15- and 20-feature scenarios, the S-SFIX–FFNN configuration produced performance values very close to SHAP-based configurations. This result is important because SHAP is a *post-hoc* explanation approach that derives feature importance from an already trained model, whereas S-SFIX is used here as a pre-classification feature scoring and reduction component within the integrated framework. Therefore, the fact that the S-SFIX–FFNN configuration produces results close to the full-feature baseline and SHAP-based configurations, especially at the 15- and 20-feature levels, indicates that the proposed framework is both effective and practically applicable for explainable feature reduction in IoT DDoS detection.

5.2 S-SFIX Robustness Results Under Gaussian Noise

Table 8 presents in detail the changes in the feature ranking produced by S-SFIX as the Gaussian noise level increases. Overall, the method did not produce a completely different ranking under low and moderate noise levels; instead, it largely preserved the core feature set located at the top of the ranking. The most striking finding is that the Variance and Magnitue features, which occupy the first two positions, only swapped places with each other across all noise levels and remained within the top two ranks in every case. Similarly, the Max and Duration features also maintained their positions within the top four ranks, showing only minor ranking changes. The AVG, Tot size, and ack_flag_number features preserved their positions exactly at the fifth, sixth, and tenth ranks, respectively, across all noise levels. This indicates that S-SFIX is able to stably keep the most discriminative core features at the forefront even under low and moderate perturbation levels.

Table 8. S-SFIX ranking changes of 46 features across Gaussian noise levels

Feature	**Clean**	**0.01**	**0.03**	**0.05**	**0.10**	**0.20**
Variance	1	2	1	1	1	1
Magnitue	2	1	2	2	2	2
Max	3	3	4	4	4	4
Duration	4	4	3	3	3	3
AVG	5	5	5	5	5	5
Tot size	6	6	6	6	6	6
Tot sum	7	7	7	7	9	9
Std	8	8	8	8	7	7
Radius	9	9	9	9	8	8
ack_flag_number	10	10	10	10	10	10

Feature	Clean	0.01	0.03	0.05	0.10	0.20
ICMP	11	11	12	12	15	16
HTTPS	12	12	11	11	11	11
syn_flag_number	13	13	13	15	16	15
rst_count	14	15	15	13	12	12
Header_Length	15	16	16	16	13	13
Min	16	14	14	14	14	14
rst_flag_number	17	18	19	21	21	21
fin_flag_number	18	20	20	22	22	22
TCP	19	17	17	17	17	17
syn_count	20	19	18	18	18	20
urg_count	21	21	21	19	19	19
Covariance	22	22	22	20	20	18
UDP	23	23	23	23	23	23
psh_flag_number	24	25	25	25	25	25
Protocol Type	25	24	24	24	24	24
fin_count	26	26	26	26	26	26
ack_count	27	27	27	27	27	27
Rate	28	29	29	29	29	31
Srate	29	28	28	28	28	29
IPv	30	33	33	33	34	34
LLC	31	32	32	31	32	33
flow_duration	32	30	30	30	30	30
HTTP	33	31	31	32	35	35
DNS	34	35	35	36	36	36
ARP	35	36	37	37	37	37
IAT	36	34	34	34	33	32
SSH	37	38	38	38	38	39
Drate	38	39	39	39	40	40
Weight	39	37	36	35	31	28
Number	40	40	40	40	39	38
DHCP	41	41	41	41	41	41
IRC	42	42	42	42	42	42
Telnet	43	43	43	43	43	43
cwr_flag_number	44	44	44	44	44	44
ece_flag_number	45	45	45	45	45	45
SMTP	46	46	46	46	46	46

The table also shows that limited but meaningful shifts occur in some of the top-ranked features. For example, while Tot sum is ranked seventh under the clean data scenario, it falls to ninth place when the noise level increases to 0.10 and 0.20. In contrast, Std rises from eighth to seventh place, and Radius rises from ninth to eighth place. Such changes indicate that the overall feature set at the top of the ranking is preserved, but that the relative positions of some features with similar discriminative power may be affected by noise. Nevertheless, the fact that nearly all features within the Top-10 remain in positions close to the same set across all perturbation levels suggests that S-SFIX exhibits strong stability in preserving the critical feature set.

When the mid-ranked features are examined, sensitivity to noise becomes more apparent. In particular, while the ICMP feature is ranked 11th under clean data, it declines successively to the 12th, 15th, and 16th positions as the noise level increases. Similarly, syn_flag_number starts from the 13th position and drops as far as the 15th and 16th positions; features such as rst_count and Header_Length also shift within the middle band as the noise level increases. In contrast, the fact that the HTTPS feature shows very limited variation within the 11–12 range reveals that not every mid-level feature is equally sensitive. This observation indicates that S-SFIX can produce greater ranking volatility particularly among features with marginal or closely similar importance scores, but that this volatility is concentrated more in the middle band than in the core feature set.

In the lower and lower-middle ranks, two different trends stand out. The first trend is that some low-

importance features fall even further back under noise. For example, while rst_flag_number is ranked 17th in the clean scenario, it declines to as low as 21st under the noise levels, while fin_flag_number drops from 18th to 22nd. Features such as HTTP, DNS, and ARP similarly shift back by several positions. The second trend is that some features that are ranked lower in the clean scenario can move relatively upward under noise. In particular, while the Weight feature is ranked 39th under clean data, it rises to as high as 28th when the noise level reaches 0.20. Similarly, Covariance rises from 22nd to 18th place, and Number rises from 40th to 38th place. This indicates that as noise increases, the relative discriminative power of some secondary features may increase, or that they may move upward in relative ranking as some more sensitive features weaken.

When the lower part of the table is examined, it is seen that some features remain almost completely unchanged. In particular, features such as DHCP, IRC, Telnet, cwr_flag_number, ece_flag_number, and SMTP preserve the same rankings across all noise levels. Since these features also have low discriminative power under the clean data scenario, the addition of noise does not produce a meaningful differentiation in their relative importance structure. This observation is also important in showing that S-SFIX consistently evaluates some low-importance features as low in importance, and that noise does not create an artificial jump in importance for such marginal features.

The ranking changes observed under Gaussian noise show that S-SFIX does not exhibit a completely unstable structure; on the contrary, it provides a high degree of stability, especially in the top-ranked core feature set. Although more noticeable shifts are observed in the middle and lower ranks as the noise level increases, the fact that most of the most critical features remain within the top 10–15 positions indicates that the proposed method is able to preserve discriminative patterns even under data perturbations. This result supports that S-SFIX offers a feature scoring mechanism with a certain degree of *AI resilience*.

Table 9 presents a summary of the total change in the feature ranking produced by S-SFIX as the Gaussian noise level increases. The results reveal that at low noise levels (0.01 and 0.03), approximately half of the features preserve their rankings completely, while most of the remaining features shift by only 1–2 positions. When the noise level reaches 0.05, shifts in the range of 3–5 positions become more frequent, and at the 0.10 and 0.20 levels, moderate-level shifts become more pronounced. Despite this, it is striking that the Top-10 feature set from the clean scenario is fully preserved across all noise levels. The Top-20 set is also fully preserved at the 0.01 and 0.03 levels, while at higher perturbation levels only two features are observed to move out of this set. This finding shows that although a gradual degradation occurs in the overall ranking structure of S-SFIX, the method exhibits strong robustness, particularly in preserving the most critical features.

Table 9. Summary of changes in S-SFIX feature rankings across Gaussian noise levels

Noise	Unchanged	1–2 positions	3–5 positions	>5 positions	Top-10 preserved	Top-20 preserved
0.01	23	22	1	0	10	20
0.03	21	23	2	0	10	20
0.05	19	23	4	0	10	18
0.10	14	25	6	1	10	18
0.20	16	22	7	1	10	18

Under the highest perturbation level, 0.20, the most pronounced ranking changes are presented in Table 10. The results show that some low- or medium-importance features can move relatively upward in the ranking under noise. In particular, the Weight feature exhibits the most striking change, rising from 39th to 28th place. Similarly, the Covariance and IAT features are also shifted upward by several positions. In contrast, features such as ICMP, rst_flag_number, fin_flag_number, and IPv decline to lower ranks under noise. This table shows that S-SFIX preserves the top core feature set; however, especially in the middle and lower bands, noise-dependent rearrangements may occur in the relative importance of some features.

Table 10. Features showing the largest ranking changes at the Noise = 0.20 level

Features with the largest upward shifts			Features with the largest downward shifts		
Feature	**Clean → Noisy**	**Change**	**Feature**	**Clean → Noisy**	**Change**
Weight	39 → 28	+11	ICMP	11 → 16	-5
Covariance	22 → 18	+4	rst_flag_number	17 → 21	-4
IAT	36 → 32	+4	fin_flag_number	18 → 22	-4
Number	40 → 38	+2	IPv	30 → 34	-4
Protocol Type	25 → 24	+1	Rate	28 → 31	-3

Table 11 presents the positions of the Top-10 features with the highest importance under the clean data scenario across different noise levels. The results reveal that this core feature set is fully preserved across all noise levels. Variance and Magnitue remain in the top two positions in every scenario, while Max and Duration preserve their positions within the top four. AVG, Tot size, and ack_flag_number do not change their rankings at any perturbation level. The only notable fluctuation in the upper ranks is the limited reordering of Tot sum, Std, and Radius within the 7–9 band due to their similar importance levels. Nevertheless, these three features also remain within the Top-10 at all noise levels. This result shows that S-SFIX is able to preserve the most critical feature set with a high degree of stability under low and moderate levels of Gaussian noise, without disrupting the core ranking structure.

Table 11. Positions of the Top-10 features under the clean data scenario across Gaussian noise levels

Feature	**Clean**	**0.01**	**0.03**	**0.05**	**0.10**	**0.20**
Variance	1	2	1	1	1	1
Magnitue	2	1	2	2	2	2
Max	3	3	4	4	4	4
Duration	4	4	3	3	3	3
AVG	5	5	5	5	5	5
Tot size	6	6	6	6	6	6
Tot sum	7	7	7	7	9	9
Std	8	8	8	8	7	7
Radius	9	9	9	9	8	8
ack_flag_number	10	10	10	10	10	10

Figure 3 shows to what extent the Top-k feature sets are preserved as the Gaussian noise level increases. The results indicate that the top-ranked feature set is particularly stable. Accordingly, the Top-10 feature set is fully preserved across all noise levels, and the Jaccard similarity remains 1.00 in every case. The Top-15 feature set is largely preserved under low and moderate perturbation levels, with the Jaccard similarity remaining constant at 0.875. The Top-20 set is fully preserved at low perturbation levels, while it declines to approximately 0.82 at the 0.05 noise level and beyond. Taken together, these findings show that S-SFIX exhibits a high degree of robustness, particularly in preserving the most critical feature set, while as noise increases, changes emerge more in the expanding and marginal-importance feature sets.

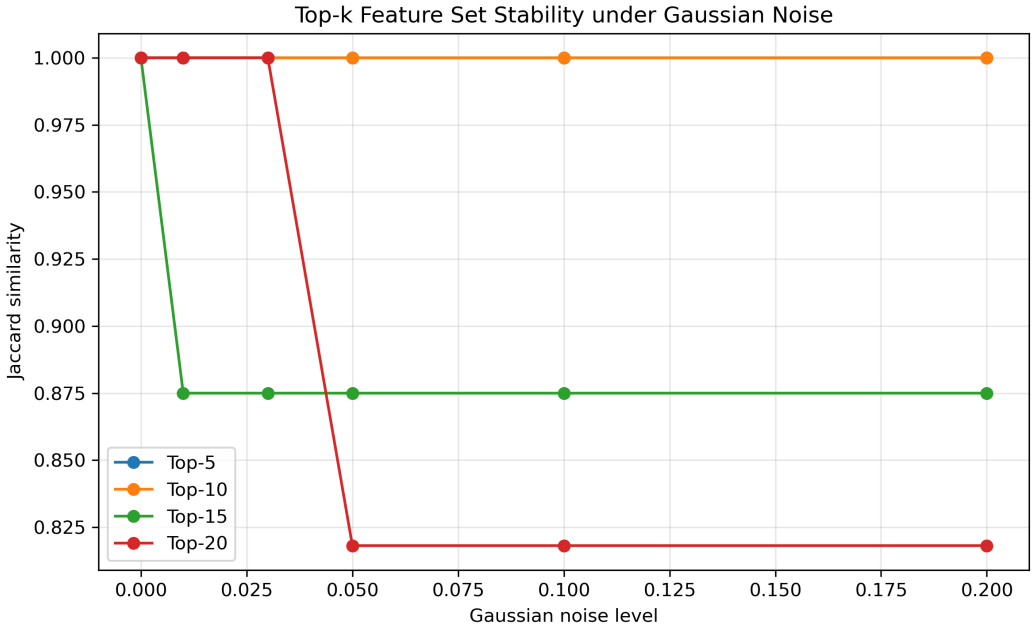


Fig. 3. Top-k feature set stability across Gaussian noise levels (Jaccard similarity).

Figure 4 shows the average positional change in the feature ranking as the noise level increases. When the curve is examined, it is seen that the mean absolute rank shift, which is naturally zero under the clean data scenario, remains limited at low noise levels but gradually increases as the noise intensity grows. In particular, the fact that the mean absolute rank shift becomes more pronounced at the 0.10 and 0.20 levels indicates that noise creates a cumulative effect on the feature ranking. Nevertheless, it is important that this increase occurs in a regular and gradual manner rather than as an abrupt and discontinuous disruption. This indicates that S-SFIX does not become completely unstable under noise; however, it produces an increasingly pronounced reordering, especially among the middle- and lower-ranked features.

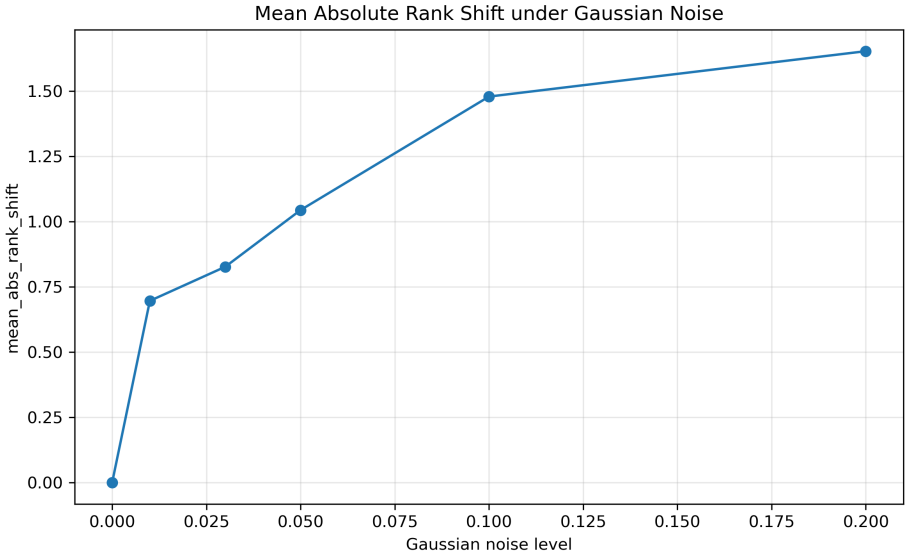


Fig. 4. Mean absolute rank shift across Gaussian noise levels.

Figure 5 shows how the rank relationship between the clean data ranking and the noisy data ranking changes in terms of the Kendall Tau coefficient. The results reveal that as the noise level increases, the Kendall Tau value gradually decreases, but this decrease remains limited. Even under the highest noise level of 0.20, the correlation remains at approximately 0.91, indicating that the ranking structure is largely preserved. In other words, although noise affects the S-SFIX ranking, it does not lead to a deviation strong enough to completely disrupt the overall relative ordering among features. This finding supports that the method exhibits strong robustness, particularly in terms of the global ranking structure.

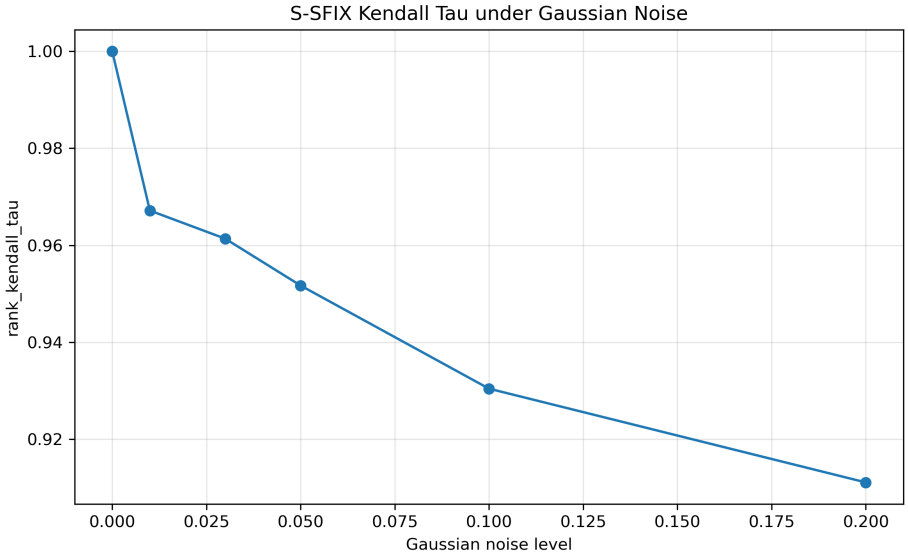


Fig. 5. Kendall Tau correlation of the S-SFIX ranking structure across Gaussian noise levels.

Figure 6 presents the rank correlation between the clean data and noisy data scenarios in terms of the Spearman coefficient. The curve shows that as the noise level increases, the correlation value gradually decreases, but remains quite high in all scenarios. Even at the highest perturbation level, the Spearman correlation remains approximately 0.98, indicating that the overall structure of the ranking produced by S-SFIX is strongly preserved. When considered together with the Kendall Tau graph, this result shows that although some local ranking changes occur under noise, the overall importance ordering of the features is largely maintained.

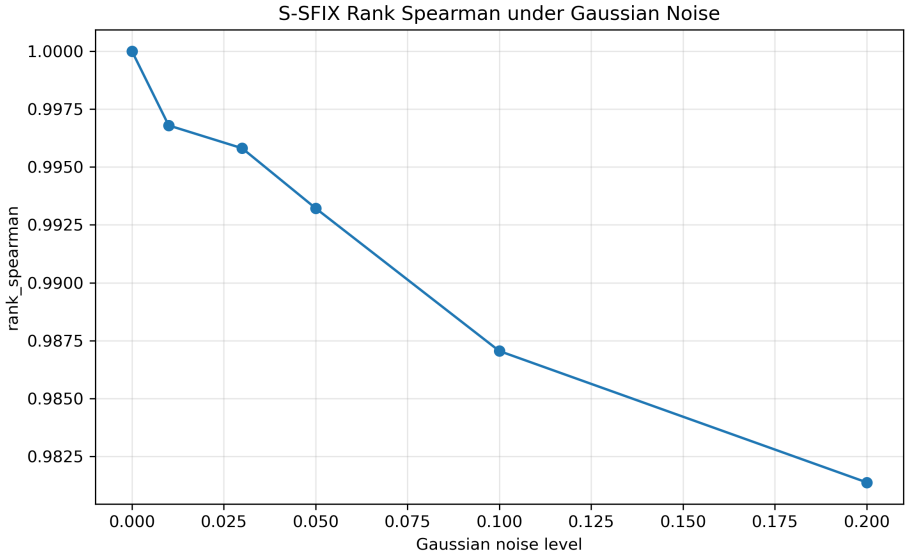


Fig. 6. Spearman correlation of the S-SFIX ranking structure across Gaussian noise levels.

Table 12 numerically summarizes the change in the score and ranking structure produced by S-SFIX as the Gaussian noise level increases. The results show that as the noise level increases, both the score correlation and the rank correlation gradually decrease, while remaining high at all levels. In particular, the fact that the rank Spearman coefficient remains approximately 0.98 and the Kendall Tau coefficient remains approximately 0.91 even at the highest perturbation level indicates that the overall ranking structure is largely preserved. At the same time, it is observed that the mean absolute rank shift remains quite limited

at low noise levels, while it increases regularly as the noise level rises. The Top-k similarity results further strengthen this finding. While the Top-5 and Top-10 feature sets are fully preserved across all perturbation levels, the Top-20 set shows only limited degradation at higher noise levels. Taken together, these results indicate that S-SFIX exhibits strong robustness, particularly in preserving the core feature set, while the degradation emerges more in extended and marginal-importance feature sets.

Table 12. Summary of S-SFIX robustness metrics across Gaussian noise levels

Noise Level	Score Pearson	Rank Spearman	Kendall Tau	Mean Abs. Rank Shift	Max Abs. Rank Shift	Top-5 Jaccard	Top-10 Jaccard	Top-20 Jaccard
0.00	1.0000	1.0000	1.0000	0.0000	0	1.0000	1.0000	1.0000
0.01	0.9968	0.9968	0.9672	0.6957	3	1.0000	1.0000	1.0000
0.03	0.9958	0.9958	0.9614	0.8261	3	1.0000	1.0000	1.0000
0.05	0.9932	0.9932	0.9517	1.0435	4	1.0000	1.0000	0.8182
0.10	0.9870	0.9870	0.9304	1.4783	8	1.0000	1.0000	0.8182
0.20	0.9813	0.9814	0.9111	1.6522	11	1.0000	1.0000	0.8182

The graphical results also support the table-based analyses. The fact that the Top-k Jaccard similarity, particularly at the Top-10 level, remains 1.00 across all noise levels shows that S-SFIX is able to fully preserve the most critical feature set. The gradual increase in the mean absolute rank shift with increasing noise reveals that the reordering is concentrated particularly in the middle and lower ranks. Nevertheless, the fact that both Spearman and Kendall Tau correlations remain high indicates that the overall ranking structure is not disrupted and that S-SFIX remains meaningfully stable under noise. Therefore, the proposed method provides a feature scoring mechanism that preserves the core feature ordering even under perturbed data conditions.

6 Discussion

The findings obtained in this study show that the proposed integrated S-SFIX–FFNN framework can identify compact, meaningful, and highly informative feature subsets in a heterogeneous IoT DDoS detection problem involving multiple attack subtypes. In the experimental design, all attack categories were merged under a single *Attack* superclass, while benign samples were added in a controlled manner from homogeneous subparts of the same dataset. This strategy provided a more balanced learning structure and enabled the proposed framework to be evaluated in a broader and more challenging threat environment. Within this structure, S-SFIX served as the explainable feature scoring and reduction component, whereas FFNN acted as the neural detection component used to assess the classification capability of the full and reduced feature sets.

6.1 Feature Reduction Capability of the Proposed S-SFIX–FFNN Framework

The findings show that the proposed framework can generate low-dimensional yet information-dense feature representations. Although the FFNN model trained with the full feature set yielded the highest absolute performance, the S-SFIX–FFNN configurations trained with only 15 and 20 features produced results very close to the full-feature baseline. This indicates that not all input features contribute equally to the classification decision and that the features ranked at the top by S-SFIX carry a substantial portion of the discriminative information required for IoT DDoS detection.

The performance loss remains limited despite the substantial reduction in the number of features shows that S-SFIX effectively prioritizes variables with high discriminative power. In this sense, S-SFIX does not only produce an interpretable feature ranking, but also functions as the feature reduction layer of the integrated framework. The FFNN component then validates the practical usefulness of these reduced feature subsets by demonstrating that competitive detection performance can be preserved under a fixed neural classification architecture. This result supports the view that the proposed S-SFIX–FFNN framework provides a balanced structure between explainability, dimensionality reduction, and detection performance.

6.2 Comparative Evaluation with Alternative Methods

One of the important aspects of this study is that the proposed S-SFIX–FFNN framework was directly compared with SHAP-, LIME-, and Random Forest-based feature reduction configurations. This compar-

ison enabled the proposed framework to be evaluated objectively against established explainability and importance-based methods in the literature. In order to ensure a fair comparison, all feature subsets generated by S-SFIX, SHAP, LIME, and Random Forest feature importance were evaluated under the same FFNN architecture, preprocessing pipeline, train-validation-test split, and hyperparameter settings.

The comparative results show that the S-SFIX–FFNN configuration produces stronger and more consistent results, especially compared with the Random Forest-based feature importance configuration. While Random Forest-based selections experience more pronounced performance loss at lower feature counts, S-SFIX-based feature subsets produce more balanced results under the same conditions. Compared with LIME, S-SFIX appears to behave more consistently, particularly in smaller feature sets, whereas LIME recovers in larger feature subsets. Although SHAP produced the highest absolute performance in some scenarios, it is important that the S-SFIX–FFNN configuration generated values close to these results.

This finding is meaningful because SHAP is a *post-hoc* explanation approach that derives feature importance from the behavior of an already trained model, whereas S-SFIX is used in this study as a pre-classification feature scoring and reduction mechanism within the integrated framework. Therefore, the ability of the S-SFIX–FFNN configuration to produce performance close to SHAP and superior to Random Forest in several configurations indicates that the proposed framework is not only interpretable, but also practically competitive in terms of classification performance.

When evaluated as a whole, the comparative analysis supports the idea that explainable feature reduction can be integrated with a neural detection model without causing substantial performance degradation. In this respect, the proposed S-SFIX–FFNN framework provides a controlled and transparent alternative for constructing compact IoT DDoS detection models.

6.3 Relationship Between Explainability and Performance

One of the main findings of the study is that explainability-based feature reduction is not in an inherent contradiction with performance. The fact that the models trained with a limited number of top-ranked features achieved performance quite close to that of the model trained with the full feature set demonstrates that a properly designed feature scoring mechanism can support both interpretability and classification performance simultaneously.

In particular, the results achieved by S-SFIX in the 15- and 20-feature configurations show that information loss can be kept limited while reducing model complexity. This also indicates its value for practical applications that require lower computational cost and a more traceable decision structure.

6.4 AI Resilience and Stability Under Noise

One of the important contributions of this study is that the concept of AI resilience has been tested directly at the level of S-SFIX. The experiments conducted under Gaussian noise show that S-SFIX does not produce a completely randomized feature ranking as noise increases; rather, it largely preserves the core feature set.

The Top-10 feature set is fully preserved under low and moderate noise conditions shows that the method can stably identify the most critical discriminative features. Similarly, the high Spearman and Kendall Tau correlations reveal that the overall ranking structure is also largely preserved. On the other hand, the more visible positional changes observed in some middle- and lower-ranked features indicate that noise creates greater volatility particularly among variables with marginal importance levels. This should not be considered a weakness of the method; on the contrary, it is an expected outcome that dominant and secondary features exhibit different sensitivities under perturbation.

Taken together, these findings indicate that S-SFIX is not a fragile ranking mechanism that works only on clean data; rather, it is capable of preserving the core importance structure even under random data perturbations. In this respect, the method offers meaningful robustness potential in the context of AI resilience.

6.5 Cross-Domain Applicability

The SFIX method originally emerged from the need for explainability and risk-sensitive feature evaluation specific to the financial domain [7, 26]. In this study, however, the method was evaluated in a non-financial, more heterogeneous, and high-dimensional problem domain, namely IoT-based network traffic and DDoS detection. The results obtained show that S-SFIX is not merely a finance-oriented approach; rather, its

multi-dimensional, class-discriminative, and risk-based feature scoring logic can also be adapted to different domains.

Nevertheless, although the present study provides strong results in a different problem domain, it does not yet include a true *cross-dataset* validation. Therefore, the findings support the cross-domain potential of the method, while also showing that additional experiments are required for full generalizability across datasets.

6.6 Contribution to the Literature

The contribution of this study to the literature can be summarized under four main headings: The primary contributions of this research to the field of IoT security and XAI are summarized as follows:

- **Unified Detection Pipeline:** We introduce an integrated S-SFIX–FFNN framework that transitions S-SFIX from a standalone explanation tool to an active component of a neural detection pipeline. By leveraging S-SFIX for feature scoring and reduction, the framework bridges the gap between post-hoc interpretability and proactive feature engineering.
- **High-Fidelity Feature Reduction:** Our framework demonstrates that low-dimensional feature subsets can be constructed without compromising detection accuracy in complex, multi-vector IoT attack scenarios. This validates explainable feature reduction as a viable strategy for building compact, resource-efficient models suitable for IoT environments.
- **Rigorous Comparative Benchmarking:** The study provides an objective assessment by benchmarking the S-SFIX–FFNN configuration against established XAI methods, including SHAP, LIME, and Random Forest. By maintaining identical architectures, preprocessing, and hyperparameters across all tests, we ensure a controlled evaluation of the relative effectiveness of S-SFIX.
- **Quantified AI Resilience:** We systematically evaluate the robustness of the feature ranking mechanism through Gaussian noise experiments. Utilizing score-based, rank-based, and Top-k preservation metrics, this study provides empirical evidence of the framework's stability under data perturbation, directly addressing the resilience requirement for high-risk cybersecurity applications.

Ultimately, the S-SFIX–FFNN framework serves as an explainable and robust pipeline that effectively reconciles the trade-off between model complexity, predictive power, and operational reliability in non-stationary IoT environments.

7 Conclusion

This study introduced an integrated S-SFIX–FFNN framework designed to bridge the gap between explainable feature engineering and robust IoT DDoS detection. Rather than treating S-SFIX as an isolated diagnostic tool, we positioned it as a primary engine within a unified pipeline to evaluate the interplay between feature compactness, classification performance, and ranking resilience.

Our experiments on the CICIoT2023 dataset confirm that the S-SFIX–FFNN configuration maintains high fidelity even under significant dimensionality reduction. While the 46-feature baseline achieved 98.45% accuracy, our framework reached 98.05% with only 20 features and 97.93% with 15 features. Remarkably, even an aggressive 78% reduction (down to 10 features) did not cause a substantial drop in performance, suggesting that S-SFIX effectively isolates the most discriminative information required for identifying IoT-based attacks.

Beyond internal benchmarks, the comparative analysis demonstrated that our approach remains competitive with established methods like SHAP and LIME, while often outperforming Random Forest-based importance in low-dimensional settings. Perhaps more importantly, our robustness analysis under Gaussian noise revealed that the S-SFIX ranking mechanism is not fragile. The consistent preservation of the "Top-10" feature set across various perturbation levels provides empirical support for the AI resilience of our framework, proving that explainability can be both interpretable and stable under data corruption.

Despite these results, several avenues for further refinement remain. The current study focused on a binary classification task; however, the complexity of modern IoT threats necessitates testing the framework against multi-class taxonomies. Furthermore, while the CICIoT2023 dataset provided a robust testing ground, validating these findings across diverse IoT datasets is essential to ensure cross-dataset generalizability.

Moving forward, we intend to expand our evaluation pipeline to include adversarial attacks, domain shifts, and temporal traffic drift to better simulate real-world non-stationary environments. We also plan to conduct a granular analysis of the framework's computational efficiency—focusing on inference latency and memory footprint—to assess its feasibility for edge deployment on resource-constrained devices. Finally, investigating the integration of S-SFIX with alternative neural and non-neural architectures will allow us to determine the broader applicability of our explainable feature reduction mechanism across different classifier families.

References

[1] Igor Linkov, Sofia Galaitsi, Benjamin D. Trump, Jeffrey M. Keisler, Alexander Kott, and Michael J. Palmer. Cybertrust: From explainable to actionable and interpretable artificial intelligence. *Computer*, 53(9):91–96, 2020.

[2] Abdullah Emir Cil and Kazim Yildiz. A systematic literature review on applications of explainable artificial intelligence in the financial sector. *Internet of Things*, 33:101696, 2025.

[3] Dylan Slack, Sophie Hilgard, Emily Jia, Sameer Singh, and Himabindu Lakkaraju. Fooling lime and shap: Adversarial attacks on post hoc explanation methods. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pages 180–186, 2020.

[4] João Gama, Indrė Žliobaitė, Albert Bifet, Mykola Pechenizkiy, and Abdelhamid Bouchachia. A survey on concept drift adaptation. *ACM Computing Surveys*, 46(4):44, 2014.

[5] Giovanni Apruzzese, Michele Colajanni, Luca Ferretti, Alessandro Guido, and Mirco Marchetti. On the effectiveness of machine and deep learning for cyber security. In *Proceedings of the International Conference on Cyber Conflict*, pages 371–390, 2018.

[6] Christian Ziegler Wehlmann. Resilient and sustainable artificial intelligence. In *Artificial Intelligence for Disaster Risk Reduction and Sustainability*, pages 5–19. Springer, 2024.

[7] Abdullah Emir Cil and Kazim Yildiz. Sfix:scalable financial-oriented interpretable explanation. *Internet of Things*, 33:101713, 2025.

[8] Osvaldo Arreche, Tanish Guntur, Jack W. Roberts, and Mustafa Abdallah. E-xai: Evaluating black-box explainable ai frameworks for network intrusion detection. *IEEE Access*, 2024.

[9] Jingnan Dong, Haolei Chen, Shigen Shen, Huibin Xu, and Zhiquan Liu. Interpretable intrusion detection for iot security: A shap-enhanced ngboost model. *Computer Networks*, 275:111937, 2026.

[10] Tasnimul Hasan and Samia Tasnim. Real-time explainable iot security with machine learning and ctgan-enhanced detection for resource-constrained devices. *Ad Hoc Networks*, 178:103937, 2025.

[11] Nicola Capuano, Giuseppe Fenza, Vincenzo Loia, and Claudio Stanzione. Explainable artificial intelligence in cybersecurity: A survey. *IEEE Access*, 2022.

[12] Jamal Alotaibi. *AI-driven intrusion detection and mitigation framework for software-defined IoT networks*. Springer Science and Business Media LLC, 2025.

[13] R. Abdallah Gad, A. Ahmed Nashat, and M. Tamer Barkat. Intrusion detection system using machine learning for vehicular ad hoc networks based on ton-iot dataset. *IEEE Access*, 2021.

[14] Mohammed M. Alani. An explainable efficient flow-based industrial iot intrusion detection system. *Computers and Electrical Engineering*, 108:108732, 2023.

[15] Dimitrios Nikolaidis and Michalis Doumpos. Credit scoring with drift adaptation using local regions of competence. *Operations Research Forum*, 2022.

[16] Riccardo Rossi, Andrea Murari, and Michela Gelfusa. A deep learning framework for feature selection and dimensional analysis: Variational explainable neural networks. *Knowledge-Based Systems*, 324:113940, 2025.

[17] Ximeng Cheng, Tianqi Wang, Di Zhu, and Jackie Ma. Uncertainty explanation of artificial intelligence models by shap. *Knowledge-Based Systems*, 337:115437, 2026.

[18] Arthur Thuy and Dries F. Benoit. Explainability through uncertainty: Trustworthy decision-making with neural networks. *European Journal of Operational Research*, 317(2):330–340, 2024.

[19] Euclides Carlos Pinto Neto, Sajjad Dadkhah, Raphael Ferreira, Alireza Zohourian, Rongxing Lu, and Ali A. Ghorbani. Ciciot2023: A real-time dataset and benchmark for large-scale attacks in iot environment. *Sensors*, 23(13), 2023.

[20] Rocío Alaiz-Rodríguez and Andrew C. Parnell. An information theoretic approach to quantify the stability of feature selection and ranking algorithms. *Knowledge-Based Systems*, 195:105745, 2020.

[21] Qingqi Han, Ruikai Shi, Liang Hu, and Wanfu Gao. Multi-label feature selection based on positive sample information weighting. *Knowledge-Based Systems*, 329:114373, 2025.

[22] Mark Nicolas Gruensteidl and Sabrina Kirrane. An evaluation of xai methods applied to information disorder detection models. *Knowledge-Based Systems*, 336:115305, 2026.

[23] Halid Can Yıldırım. Knowledge-infused probabilistic modelling for decision support in metal additive manufacturing. *Knowledge-Based Systems*, 335:115212, 2026.

[24] Zsolt T. Kosztyán, Marcell T. Kurbucz, and Attila I. Katona. Network-based dimensionality reduction of high-dimensional, low-sample-size datasets. *Knowledge-Based Systems*, 251:109180, 2022.

[25] Abdullah Emir Cil, Kazim Yildiz, and Ali Buldu. Detection of ddos attacks with feed forward based deep neural network model. *Expert Systems with Applications*, 169:114520, 2021.

[26] Abdullah Emir Cil, Tolga Buyuktanir, and Kazim Yildiz. Comprehensive explainable ai approach for audit opinion classification using feed-forward neural networks. *Knowledge-Based Systems*, 339:115606, 2026.