Open Access ISSN: 3007-1437 (Online), ISSN: 3007-1429 (Print)

Alkadhim Journal for Computer Science (KJCS)

Journal Homepage: https://alkadhum-col.edu.iq/JKCEAS

CLEAR-IDS: Leakage-Free Adaptive Resampling and Diversity-Weighted Explainable Ensemble Learning for Network Intrusion Detection

[1]Muna Al-Saadi*, [1]Bushra A-Saadi

[1] University of Information Technology and Communications (UoITC), Baghdad – Iraq

Article information

Article history:
Received: August, 05, 2026
Accepted: September, 23, 2026
Available online: September, 25, 2026

Keywords:
Network intrusion detection;
Data leakage;
Class imbalance;
Adaptive resampling;
Ensemble learning;
Explainable artificial intelligence;
Probability calibration;
UNSW-NB15

Corresponding Author:
Muna Al-Saadi
munayousif@uitc.edu.iq

DOI:
https://doi.org/10.61710/s6qa0r78

Abstract

Reliable network intrusion detection requires more than high aggregate accuracy because data leakage, severe class imbalance, unstable feature selection, redundant ensemble decisions, and poorly calibrated probabilities can produce optimistic yet operationally weak results. This study presents CLEAR-IDS, a cross-fold leakage-free explainable framework that integrates fold-contained preprocessing, stability-aware multi-attribution feature selection, class-aware adaptive resampling, diversity-weighted ensemble learning, threshold optimization, and multi-level explanations. The framework was evaluated on the official UNSW-NB15 partitions using stratified five-fold model development and an untouched test set. SHAP, permutation importance, and mutual information were combined with redundancy control; Borderline-SMOTE, SMOTE-ENN, SMOTE-Tomek, ADASYN, and no resampling were assessed only within training folds; and random forest, XG-Boost, and multilayer perceptron models were aggregated according to out-of-fold quality, diversity, and stability. For binary detection, the optimized CLEAR-IDS decision rule achieved 87.94% accuracy, 86.83% balanced accuracy, 87.45% Macro-F1, and a 24.08% false-positive rate, improving Macro-F1 by 0.96 percentage points over the strongest tested binary baseline. In ten-class detection, CLEAR-IDS achieved 52.00% Macro-F1 and did not surpass XG-Boost (52.42%), revealing persistent difficulty for Analysis and Backdoor. The controller selected Borderline-SMOTE in three folds and no resampling in two, while the final 25-feature subset showed strong cross-fold stability (mean Jaccard 0.8803; mean Spearman 0.9421). Temperature scaling worsened NLL, Brier score, and ECE, underscoring the importance of reporting negative calibration results. The findings indicate that leakage control and adaptive decision design improve binary reliability, although rare overlapping attack classes remain an open challenge.

1. Introduction

Modern communication networks have grown in size and complexity due to the fast development of cloud computing, IoT, edge intelligence, and linked cyber-physical systems. In vital areas including healthcare, transportation, energy, banking, industry, and public infrastructure, these technologies enable intelligent automation, remote monitoring, and real-time data interchange. On the other hand, the attack surface that bad actors might exploit has grown in modern network settings due to their rising interconnection, heterogeneity, and openness. Threats include denial-of-service assaults, reconnaissance, illegal access, malware propagation, network attacks, data exfiltration, and exploiting susceptible devices are ever-present in modern networks. Efficient network intrusion detection is now a must for safeguarding the availability, confidentiality, and integrity of digital systems due to the increasing frequency and complexity of such attacks [1,2]. Conventional intrusion detection systems often use either pre-existing signatures or rules that are manually built. When confronted with zero-day assaults, quickly changing hostile behavior, and complicated traffic patterns, signature-based systems struggle to maintain efficacy, despite their efficiency in identifying previously known attacks. While traditional intrusion detection systems rely heavily on predetermined attack signatures, newer systems that use machine learning may learn patterns from network data to spot suspicious behaviors. As a result, several different types of learning approaches have been studied for binary detection of intrusions and multiple-class attack identification, including supervised or unsupervised and hybrid methods. The overall classification accuracy of a model to detect intrusions is not the only metric that should be considered when evaluating its practical utility. Furthermore, a trustworthy system should be able to identify uncommon types of attacks, prevent biased assessment, maintain consistency across data partitions, measure the accuracy of its predictions, as well as offer justifications which cybersecurity experts can understand and verify [3,4]. Previous studies have applied numerous machine learning and deep learning methods to network intrusion detection, including decision trees, random forests, support vector machines, k-nearest neighbors, gradient-boosting algorithms, multilayer perceptrons, convolutional neural networks, recurrent neural networks, autoencoders, transformers, and hybrid ensemble architectures [5,6]. Feature-selection and dimensionality-reduction methods have also been employed to remove irrelevant or redundant variables, reduce computational complexity, and improve model generalization. These methods include filter-based statistical measures, wrapper-based search procedures, embedded feature-importance mechanisms, mutual information, permutation importance, and model-specific attribution techniques. Considerable attention has also been given to the class-imbalance problem in intrusion detection datasets. Conventional oversampling and under sampling methods, including random oversampling, random under sampling, the Synthetic Minority Oversampling Technique (SMOTE), Borderline-SMOTE, the Adaptive Synthetic Sampling method (ADASYN), SMOTE combined with Edited Nearest Neighbors (SMOTE-ENN), and SMOTE combined with Tomek links (SMOTE-Tomek), have been used to increase the representation of minority attack classes [7,8]. In parallel, ensemble learning methods have been introduced to combine the complementary capabilities of multiple classifiers, while decision-threshold optimization has been investigated to improve minority-class sensitivity. More recently, explainable artificial intelligence techniques, particularly SHAP-based interpretation and feature-importance analysis, have been incorporated into intrusion detection systems to improve model transparency [3,4]. Despite this progress, preprocessing, feature selection, class balancing, classification, threshold calibration, and explainability are frequently treated as isolated components rather than as interdependent elements of a unified and evaluation-safe intrusion detection framework.

1.1 Research Problem and Motivation

Although there has been significant progress in machine learning intrusion detection systems, these systems are still neither practical or dependable due to many operational and methodological difficulties. Despite the fact that a number of studies have shown respectable overall accuracy, data on the detection of uncommon, complicated, or highly overlapping attack types is scarce. Since aggregated accuracy is based on average traffic and attacks, these metrics might mask the reality that minority class aren't doing well compared to the others. As a result, a model can provide the impression of excellent performance while really omitting information on the kind of attacks that might pose the greatest threat to security [9,10]. The issue of data leakage during the development and evaluation of models is another important one.

Before cross-validation or the whole development dataset is used, many intrusion detection pipelines run data-dependent operations including normalization, category encoding, feature selection, sampling oversampling, synthetic sample creation, or threshold calibration. Under these circumstances, data collected during validation

might indirectly impact the training process. Because of this contamination, the model's generalizability to unseen traffic on networks may be overestimated, and findings may be skewed in an optimistic direction [11,12]. Also, different models and data divisions could make quite different feature-selection judgments. Unstable variants of features that lose their informativeness as data distribution changes might be the outcome of feature selection utilizing only one attribution approach or train-validation split. Similarly, when building ensemble models, weights are typically allocated experimentally or equally, without proper consideration of classifier variety, cross-fold stability, or prediction redundancy. Thus, increasing the accuracy of predictions is not the main issue requiring investigation. The goal is to create an intrusion detection system that can withstand stable, minority-sensitive, transparent, and realistically unbalanced network situations without leaking data [13,14].

1.2 Main Limitations

1.2.1 Class Imbalance

The first fundamental shortcoming of present intrusion detection methods is the substantial class imbalance that is usually noticed in network traffic statistics. The majority of the current findings generally correspond to standard network traffic and conventional assaults, whereas there are very few examples of advanced, developing, or rare attacks. Standard classifiers trained on distributions like this are biased toward the majority classes as lowering total classification error gives little motivation to accurately detect unusual data. Therefore, a model might have good accuracy and weighted-average effectiveness, but low recall and F1 scores for the minority attack classifications [15,16]. This constraint is especially important in the domain of cybersecurity, as infrequent attacks might be indicative of severe dangers such as privilege escalation, penetration, data exfiltration, or novel harmful activities. The result of misclassifying these assaults as regular traffic is more serious than misclassifying just a handful of common observations. Therefore, a model for intrusion detection should be evaluated by macro-averaged efficiency, per-class recall, minor-class detection, and particular class-specific error patterns instead of depending mostly on overall accuracy. Although resampling techniques have been widely adopted to address class imbalance, many existing approaches apply one predefined method to the entire dataset. However, minority attack classes may differ substantially in their imbalance ratios, neighborhood distributions, class-overlap levels, local densities, and susceptibility to noise. A fixed oversampling strategy may generate synthetic samples in ambiguous regions, amplify mislabeled or noisy observations, and distort the original decision boundaries. Conversely, aggressive under sampling may remove informative majority-class observations and reduce the classifier’s ability to represent normal traffic accurately. The effectiveness of a resampling technique therefore depends on the structural properties of the data to which it is applied. Borderline-SMOTE may be useful when informative minority samples are located near a decision boundary, whereas cleaning-based methods such as SMOTE-ENN or SMOTE-Tomek may be more appropriate when class overlap or noisy neighborhoods are present. ADASYN may emphasize minority regions that are difficult to learn, but it may also increase the influence of noisy observations. In some cases, no resampling may be preferable when the minority distribution is sufficiently informative or when synthetic generation is likely to degrade class separability. These differences highlight the need for a class-aware adaptive resampling mechanism that selects an appropriate balancing strategy according to the characteristics of each training fold rather than applying a uniform solution across all datasets and attack categories. The second key drawback of the present machine learning -driven intrusion detection systems is their lack of explain-ability. Many high-performance classifiers are black-box predictors, giving no insight into the reasons behind their conclusions. Complex models may be able to find nonlinear linkages and nuanced attack patterns, but their predictions can be difficult for analysts in security to validate, comprehend or convert into effective defensive measures. This constraint may hinder the practical use of machine learning algorithms in security-critical contexts where an automatic prediction must be validated by interpretable evidence [17,18]. Some recent research tries to solve this problem by model-specific feature significance or global SHAP analysis.. However, the global feature rankings only represent aggregated model behavior. They may suggest what network properties are typically important, but cannot necessarily clarify why a particular flow was classed as malicious, the reason a specific attack category was picked, or what adjustments would modify the projected outcome. Moreover, the most important attributes for one attack class may be different from those for another class, which requires class-specific explanations to provide a relevant security analysis. Existing explainable intrusion detection tools also provide limited support to support local explanations, counterfactual logical reasoning, prediction based on uncertainty, and humans-in-the-loop evaluation. Local explanations are needed to explain a certain warning. Counterfactual explanations may show the minimal modifications needed to modify a forecast. Estimating

uncertainty may help differentiate between high-confidence alarms and those that are confusing and need additional inquiry. Without such capabilities, a system for intrusion detection could provide a projected label without adequate context-specific data for operational decisions. Hence, a complete explain-ability framework should include global, class-level, local, and counterfactual explanations with reliability estimates to allow cybersecurity analysts to evaluate the believability of individual forecasts [19,20].

1.3 Research Gap

The preceding limitations reveal a clear gap in the existing intrusion detection literature. Previous studies have independently investigated class resampling, feature selection, ensemble learning, threshold optimization, explainable artificial intelligence, and uncertainty assessment. There is a lack of unified architecture that combines these features to prevent cross-fold leakage, handle minority classes adaptively, identify stable features, aggregate predictions while considering diversity, calibrate decision thresholds, and provide multi-level explanations [21,22]. Specifically, creating a system where all data-dependent operations are carried out only in the training phase for every cross-validation fold has received relatively little focus. We still don't have an adaptive sampling controller that takes into account factors like minority-sample density, local noise, class overlap, and imbalance ratio when deciding how to balance the samples. Additionally, there is a lack of comprehensive integration of redundancy control, cross-fold stability analysis, and complementary feature-attribution approaches in the existing literature [23,24]. In addition, the stability and variety of predictions are often overlooked by ensembles intrusion detection systems that place an emphasis on classifier performance. While each classifier works well on its own, combining them may not provide much value due to their strong correlation. In a similar vein default decision threshold could lower the recollection of uncommon assaults while favoring classes in the majority. Because of these restrictions, it is necessary to combine minority-sensitive threshold calibration with ensemble weighting that takes performance and diversity into account.

The central research gap can therefore be stated as follows:

No comprehensive, cross-fold leakage-free the intrusion detection system currently exists that can reliably explain binary and multiple classes security choices in a way that is stable, multi-level, and adaptive toward class imbalance. To create intrusion detection systems that are statistically credible, repeatable, transparent, and operationally helpful, it is essential to fill this gap.

1.4 Research Objectives

The stated research need is addressed by this work with a proposed solution named CLEAR-IDS: Cross-Fold Leakage-Free Explainable Flexible Sampling Intrusion Detection System. The primary aim is to provide an integrated intrusion detection system that integrates leakage-free model construction, adaptive minority-class management, stable feature selection, ensemble learning, threshold calibration, and multi-level explain-ability.

More specifically, the study aims to:

1. Develop an end-to-end intrusion detection architecture in which preprocessing, feature selection, resampling, model training, ensemble construction, and threshold calibration are performed exclusively within the training portion of each cross-validation fold.
2. Design a stability-aware multi-attribution feature-selection mechanism that integrates SHAP values, permutation importance, and mutual information while controlling feature redundancy and measuring feature-selection consistency across folds.
3. Develop a class-aware adaptive resampling controller that selects an appropriate balancing strategy according to the imbalance ratio, class overlap, noise level, and minority-sample density of each training fold.
4. Construct a diversity-aware weighted ensemble that integrates complementary machine learning classifiers based on their predictive performance, stability, and decision diversity.
5. Calibrate classification thresholds with particular attention to macro-F1 and minority-class recall and support both binary intrusion detection and multiclass attack identification.

6. Provide multi-level explain-ability and reliability assessment through global, class-specific, local, and counterfactual explanations, together with prediction uncertainty and human-in-the-loop decision support.

1.5 Methodological Overview

The proposed methodology begins by isolating an independent hold-out test set before performing any data-dependent preprocessing or model development. This test set remains untouched throughout feature selection, resampling, hyper-parameter tuning, ensemble construction, and threshold calibration and is used only for the final assessment. The remaining development data are evaluated using stratified five-fold cross-validation to preserve the relative distribution of normal and attack classes across the folds. Within each cross-validation iteration, preprocessing is fitted exclusively to the training fold. This stage includes data cleaning, treatment of missing and invalid values, categorical encoding, and numerical feature normalization. The fitted preprocessing transformations are then applied to the corresponding validation fold without using its statistical information during training. A stability-sensitive multi-attribution feature-selection procedure is then used to the preprocessed training data. To assess the significance of features from different angles, we employ mutual information, permutation importance, and SHAP values. To avoid highly correlated variables dominating the chosen feature subset, we aggregate the importance measures that come from the process and apply redundancy control. After that, we check the consistency of feature-selection across folds to find the variables that are still useful when we divide the data in various ways [13,25]. After feature selection, the proposed class-aware adaptive resampling controller examines the imbalance ratio, minority-sample density, class overlap, and local noise characteristics of the training fold. Based on these properties, the controller selects an appropriate strategy from Borderline-SMOTE, SMOTE-ENN, SMOTE-Tomek, ADASYN, or no resampling. Resampling is applied only to the training fold, ensuring that no synthetic observations are generated from or introduced into the validation and hold-out test sets. The resulting training data are used to develop random forest, XG-Boost, and multilayer perceptron classifiers. These models are selected because they provide complementary representations of nonlinear decision structures and heterogeneous network patterns. Their predictions are aggregated using a diversity-aware weighed combination. A meta-learning method is used to develop a proper prediction aggregation approach. The ensemble weights take into account not just the individual performance of the predictors but also their stability between folds and discordance amongst base classifiers. Then, decision thresholds are tuned to target minority-class recall and macro-F1 using validation predictions. In doing so, we are avoiding the risk of favoring majority classes when using default probability criteria. Retraining the whole framework using a complete development set and evaluating it once on the isolation hold-out test set follows the identification of the most stable configuration. The finished product can distinguish between benign and malicious communications using a binary classification system and can identify different types of attacks using a multiclass approach. The framework is evaluated using metrics suitable for imbalanced classification, including macro-precision, macro-recall, macro-F1, balanced accuracy, per-class recall, and confusion matrices. Area-under-the-curve measures, calibration indicators, uncertainty estimates, computational cost, and cross-fold variability are also examined. Comparative experiments and ablation studies are conducted to evaluate the contribution of adaptive resampling, multi-attribution feature selection, diversity-aware ensemble weighting, threshold calibration, and multi-level explain-ability.

1.6 Research Contributions

To bridge the identified research gap, this study makes the following methodological and practical contributions to the field of intelligent network intrusion detection:

The principal contribution of this study is the development of CLEAR-IDS, a unified cross-fold leakage-free framework for adaptive, reliable, and explainable network intrusion detection. Unlike conventional pipelines, all data-dependent operations, including preprocessing, feature selection, resampling, model training, ensemble construction, and threshold calibration, are restricted to the training portion of each cross-validation fold, while an independently isolated hold-out test set is preserved for final evaluation. The framework introduces a stability-aware multi-attribution feature-selection mechanism that integrates SHAP, permutation importance, and mutual information with redundancy control and cross-fold consistency analysis. It also proposes a class-aware adaptive resampling controller that selects among Borderline-SMOTE, SMOTE-ENN, SMOTE-Tomek, ADASYN, and no resampling according to the imbalance ratio, class overlap, noise level, and minority-sample density. Furthermore, random forest, XG-Boost, and multilayer perceptron models are integrated through a diversity-aware weighted

ensemble and meta-learning strategy. Minority-sensitive threshold calibration is employed to support both binary intrusion detection and multiclass attack identification. Finally, CLEAR-IDS provides global, class-specific, local, and counterfactual explanations, together with prediction uncertainty and human-in-the-loop support. The proposed framework is evaluated using a comprehensive protocol that considers predictive performance, minority-class recall, cross-fold stability, calibration quality, feature consistency, computational efficiency, and component-level ablation analysis. These contributions distinguish CLEAR-IDS from conventional intrusion detection pipelines by integrating methodological rigor, adaptive class balancing, stable feature analysis, ensemble diversity, and comprehensive explain-ability within a single evaluation-safe architecture.

The rest of this work is arranged as follows. Section 2 presents a critical evaluation of leakage-free validation, unbalanced intrusion detection, stable a feature selection, standard ensemble learning, and explainable AI. The issue formulation and design concepts are presented in Section 3. Section 4 describes the CLEAR-IDS approach and algorithmic procedure. Section 5 summarizes the dataset, validation process, implementation environment, baselines and assessment metrics. Section 6 presents the findings of the experiments; Section 7 interprets the results and analyzes their practical consequences and Section 8 closes the study and recommends suggestions for further research.

2. Related Work

A critical assessment of network intrusion detection systems that use machine learning is presented in this section. The studies were chosen because they pertain to the main points of the CLEAR-IDS framework: preventing data leakage, handling class imbalance, mitigating class overlap, selecting features, learning ensembles, calibrating thresholds, explainable AI, and prediction reliability. In this review, the literature is not presented in a strictly chronological order, but rather according to four scientific principles: (1) leakage-free preprocessing evaluation and creative integrity; (2) class imbalance, adaptive and creative resampling and class crossover; (3) feature selection and cross-partition control; and (4) ensemble learning, explain-ability and reliability. I may assess the extent to which the existing literature satisfies the interrelated methodological requirements of an efficient intrusion detection system by classifying the research according to their contributions. It is stated clearly, rather than implied, that data such as precise post-processing the number of samples, software versions, or device settings are unavailable in cases where publisher records do not provide them. As a result, the analysis separates out the confirmed methodological evidence from the data that was inadequately cited in the literature.

2.1 Leakage-Free Preprocessing and Evaluation Integrity

The study's findings highlight the need for clear documentation of data splitting, preprocessing, model selection, and evaluation processes, and how incorrectly separating training and evaluation data may make complicated models seem much better than they really perform in really unseen situations. The work wasn't intrusion detection specific, but it laid a solid methodological groundwork for CLEAR-IDS. CLEAR-IDS uses the training data from each cross-validation fold for preprocessing, feature selection, resampling, ensemble construction, and threshold calibration, while separating the test set before any data-dependent operation. In 2023, Kapoor and Narayanan [11] conducted a comprehensive assessment spanning many disciplines that included 294 publications from 17 different areas to determine the frequency, origins, and effects of data leaking in scientific research using machine learning. Methodological auditing, literature synthesis, and experiment replication were used to show how eight forms of leakage might provide unreasonably high and irreproducible performance estimations. The impact of pattern leaking on the dependability of machine learning intrusion detection models utilizing NSL-KDD, UNSW-NB15, and KDDCUP99 was investigated by Bouke and Abdullah [12] in 2023. Decision trees, logistic regression, k-nearest neighbors, support vector machines, random forests, and gradient boosting were among the six classifiers tested, along with leakage-free and leakage-affected experimental pipelines. The results demonstrated that classifiers' sensitivity to leakage varied, and that incorporating testing data during preprocessing led to exaggerated classification results; in contrast, leakage-free pipelines yielded lower but more believable predictions. The study provided strong intrusion-detection-specific evidence on multiple datasets and algorithms, but CLEAR-IDS advances the state of the art by incorporating leakage prevention into preprocessing, feature selection, synthetic sampling, model weighting, threshold calibration, adaptive class balancing, and multi-level explain-ability.

2.2 Comparative Analysis of Leakage-Oriented Studies

(Kapoor and Narayanan and Bouke and Abdullah's) 2023, investigations concur that the stated machine learning performance cannot be deemed accurate until all the information-dependent operations are separated from validation and testing observations. Kapoor and Narayanan identified the issue at a broad scientific scale while Bouke and Abdullah showed its immediate repercussions for network intrusion detection. The former gives a wider taxonomy and view on repeatability, whereas the latter has more solid domain-specific evidence [11,12]. However, neither study delivers an end-to-end intrusion detection architecture in which cleaning, encoding, normalization, feature attribution, redundancy reduction, resampling, ensemble weighting, and threshold calibration are all fitted separately within every training fold. This omission is significant because leakage can occur after the initial train–test split through feature selection, synthetic data generation, or repeated threshold optimization. The literature therefore supports leakage-free evaluation in principle but provides limited integration of that principle across the complete intrusion detection lifecycle. CLEAR-IDS addresses this limitation through independent hold-out isolation and fold-contained data processing [11,12].

2.3 Class Imbalance, Resampling, and Class Overlap

In 2023, Bagui et al. [7] investigated rare-attack detection in imbalanced network-intrusion data by comparing different oversampling-to-under sampling ratios and two alternative resampling designs. Their experiments used UNSW-NB15 and UWF-ZeekData22 and evaluated random forest after random under sampling and Borderline-SMOTE-based oversampling. The study showed that classification performance depends on the resampling ratio and on whether under sampling is performed before or after the train-test division. Although it demonstrated that balancing decisions should reflect the data distribution, the method used predefined experimental designs rather than a fold-specific controller. CLEAR-IDS extends this direction by selecting the resampling strategy only within each training fold according to imbalance, overlap, density, and noise. In 2024, Le et al. [15] studied ensemble classification combined with under sampling and oversampling algorithms for imbalanced multiclass intrusion detection. Hybrid sampling and ensemble-learning processes were used on the CHADC2020 and IoTID20 datasets to increase the representation and categorization of the minority attack categories for the evaluation of the proposed methodology. The authors achieved average F1 scores of above 98% on both datasets which suggests that hybrid sampling with ensemble classification may perform well on unbalanced data. The integration of balancing and ensemble learning was successful, however the research did not formalize adaptive fold-wise method selection, leakage-free encapsulation, threshold calibration, explanation reliability, and uncertainty analysis, which are addressed in CLEAR-IDS. In 2024, Zoghi and Serpen [8] proposed a method for UNSW-NB15 to deal with the problems of class imbalance and class overlap by using imbalance-sensitive learning, ensemble classification, and threshold modification. They trained Balanced Bagging, XG-Boost and Hellinger-distance random forest models on the official training and testing partitions, merged their results by majority vote, and proposed two strategies for threshold change. Although 98.26% attack sensitivity and 97.80% binary accuracy were achieved, poor recall was shown for several minority attacks, such as Analysis and DoS, which showed that class-specific errors might be disguised by high overall accuracy. CLEAR-IDS extends the earlier work, and, unlike the latter, includes adaptive class balancing, ensemble weighting based on diversity and stability, multi-level explanations and uncertainty-driven analyst intervention. Al-Ajlan and Ykhlef [16] presented a hybrid resampling technique based on adaptive GANs, while Shanmugam et al. [9] conducted a comprehensive evaluation of machine-learning methods for unbalanced intrusion detection in 2025. While these studies demonstrate a trend away from fixed oversampling and toward data-dependent balance, they do not cover all the bases in terms of training fold-by-fold preprocessing, a feature selection, resampling and ensemble building, and threshold calibration. Confirming that focused balancing may enhance sensitivity to uncommon classes but potentially alter the precision-recall trade-off, Harini et al (2023). [10] integrated deep learning using sampling to enhance minority-attack detection. Their results show that overall accuracy isn't enough to judge resampling; per-class criteria are far more appropriate.

2.4 Feature Selection and Cross-Partition Stability

To enhance the multiclass effectiveness of an MLP-driven intrusion detection system using UNSW-NB15 and decrease input dimensionality, Yin et al. [5] suggested a hybrid feature-selection strategy in 2023. Using standard training and testing partitions, the method performed recursive feature reduction based on MLP after combining Information Gain with random forest significance to decrease the initial search space. Although performance remained poor for several attacks and numerous uncommon categories were excluded, the technique raised

accuracy by 82.25% to 84.24%, decreased the feature space by 42 to 23 characteristics, and attained a weighted F1 score in 82.85%. Despite the study's findings on the benefits of integrating filter and wrapping approaches, CLEAR-IDS keeps unusual classes while adding SHAP, mutual information, permutation significance, redundancy management, and cross-fold stability evaluation. To determine whether SHAP-based reduction of features might maintain excellent prediction accuracy across many benchmark datasets, Sadhwani et al. [6] created explainable multiclass based deep-learning intrusion detection algorithms in 2025. SHAP was used to identify the fifteen most significant characteristics for retraining CNN, LSTM, and BiLSTM models after evaluation on NSL-KDD, UNSW-NB15, TON-IoT, and X-IIoTID. Results on NSL-KDD, TON-IoT, UNSW-NB15, and X-IIoTID shown that explanation-guided reduction may maintain performance while lowering dimensionality, with reported accuracies of 98.21%, 97.80%, 92.90%, and 98.09%, respectively. As an alternative to CLEAR-IDS, which integrates numerous significance measures with consistency-based feature selection, the research primarily used SHAP without utilizing multi-attribution consensus, redundancy analysis, or cross-fold selection stability. Recent feature-engineering studies further demonstrate the value of combining selection mechanisms. Bakır and Ceviz [13] coupled hybrid feature selection with genetic-algorithm hyper-parameter tuning, whereas Walling and Lodh [23] proposed adaptive feature selection for IoT intrusion detection. Ayad et al. [25] provide supporting evidence; the former created a hybrid feature-selection method for anomaly detection, while the latter examined UNSW-NB15 to enhance the efficacy of detection. Compact representations are supported by these contributions, although there is still a lack of integration between cross-fold attribution agreement and explicit leakage management.

2.5 Ensemble Learning, Explain-ability, and Reliability

In 2024, Sharma et al. [3] developed deep-learning intrusion detection models with reduced feature dimensionality and enhanced interpretability using model-agnostic explanation methods. The researchers applied filter-based feature selection before training DNN and CNN models on NSL-KDD and UNSW-NB15, while SHAP and LIME were used to explain the resulting predictions. The study reported improved classification performance and showed that SHAP and LIME could identify influential network-traffic features, although the accessible source did not provide sufficient numerical detail to reproduce all comparisons. Its integration of global and local interpretation is valuable, but CLEAR-IDS further evaluates explanation stability and agreement while adding class-aware resampling, calibrated thresholds, counterfactual reasoning, and uncertainty-based referral. In 2024, Ben Ncir et al. [4] proposed a two-phase explainable ensemble deep-learning framework to improve intrusion classification and prediction transparency. The approach used three one-dimensional LSTM models whose outputs were integrated by a meta-learner, followed by SHAP-based explanation, and it was evaluated on NSL-KDD and UNSW-NB15 using several k-fold settings. At the best fold configuration, the ensemble achieved 97.05% accuracy and 97.19% F1 on NSL-KDD and 95.15% accuracy and 95.25% F1 on UNSW-NB15, outperforming the individual LSTM models. Although the study supports meta-level aggregation and post-hoc explanation, CLEAR-IDS introduces structurally diverse models, adaptive resampling, diversity- and stability-aware weighting, explanation reliability, counterfactual evidence, and predictive uncertainty. In 2025, Hermosilla et al. [17] examined the consistency, faithfulness, stability, and computational practicality of SHAP and LIME explanations for forensic intrusion detection. Using UNSW-NB15, the study compared XG-Boost and Tab-Net with a reduced 39-feature space and evaluated explanation agreement, fidelity, stability, generalization, and execution time on a separate held-out test partition. Validation accuracy was around 97.8% while the held-out test accuracy was 85.91% for XG-Boost and 84.41% to Tab-Net, and KernelSHAP took more than 8.3 seconds for each instance, revealing generalization and computing issues. The thorough evaluation of explanation quality in this work directly affects CLEAR-IDS, which expands this reliability viewpoint with adaptive balancing, diversity-weighted ensembles, minority-sensitive calibration, counterfactual explanations, uncertainty estimates, and human-in-the-loop assistance. Research focused on reliability has started to approach uncertainty and the quality of explanation as tangible results. Talpini et al. [19] introduced uncertainty quantification in ML based network intrusion detection. Mohale and Obagbuwa [18] presented a thorough study of XAI integration in IDS. Al and Sağıroğlu [20] further investigated explainable AI models for intrusion detection. This body of data together drives CLEAR-IDS to integrate global, class-specific, local, and counterfactual explanations with calibrated confidence and analyst referrals. There has been a fast expansion of explainable and ensembles intrusion detection as well. In their study, Keshk et al. [1] integrated deep intrusion detection with both global and local logical explanations, whereas Alani [2] created a flow-based explainable IIoT detection that is efficient. Both Ahmed et al. [21] and Alsaffar et al. [22] used ensemble learning in conjunction with explain-ability or hybrid feature selection. For Internet of

Things traffic, Cao et al. 2023,[14] investigated layered ensembles. Although complementary learners are validated in this research, models are often not weighted simultaneously by variety, cross-fold stability, predictive quality. Research papers that are representative of the field and have direct relevance to the primary modules of CLEAR-IDS are compared in Table 1. The check mark indicates explicit therapy, the triangle indicates partial or indirect treatment, and the dash indicates that the component is not reported. All of the evidence is included in the thematic subsections.

Table (1): Comparative research-gap matrix for the reviewed studies.

Study	Leakage-safe fold-wise pipeline	Adaptive resampling	Stable multi-attribution features	Diversity and calibration	Multi-level XAI and reliability	Main remaining gap
Kapoor and Narayanan (2023) [11]	✓	—	—	—	—	Establishes leakage principles but does not provide an IDS implementation.
Bouke and Abdullah (2023) [12]	✓	—	—	—	—	Diagnoses preprocessing leakage but does not integrate balancing, stable feature selection, or XAI.
Bagui et al. (2023) [7]	△	△	—	—	—	Compares fixed resampling designs and ratios without a fold-specific strategy selector.
Yin et al. (2023) [5]	△	—	△	—	—	Hybrid feature selection is not assessed for cross-fold stability; rare classes are removed.
Le et al. (2024) [15]	Not explicit	△	—	△	—	Integrates hybrid sampling and ensemble learning but lacks adaptive selection, calibration, and explanations.
Zoghi and Serpen (2024) [8]	△	△	—	△	—	Addresses overlap and thresholds, but the ensemble is not diversity-weighted and rare-class recall remains uneven.
Sharma et al. (2024) [3]	Not explicit	—	△	—	△	Uses SHAP and LIME but does not evaluate cross-fold explanation stability, uncertainty, or counterfactuals.
Ben Ncir et al. (2024) [4]	△	—	—	△	△	Uses an explainable meta-ensemble, but base models are structurally similar and adaptive balancing is absent.
Sadhwani et al. (2025) [6]	Not explicit	—	△	—	△	Provides multi-dataset SHAP analysis but lacks multi-attribution stability and reliability-aware explanations.
Hermosilla et al. (2025) [17]	△	—	—	—	△	Evaluates explanation fidelity and stability but does not integrate balancing,

						ensemble calibration, or uncertainty.
Proposed CLEAR-IDS	✓	✓	✓	✓	✓	Integrates the previously disconnected requirements in a unified evaluation-safe architecture.

Table 1 shows that recent studies usually address leakage control, adaptive balancing, stable feature selection, ensemble calibration, or explain-ability separately. CLEAR-IDS is positioned as an integration framework; the comparison is not intended to imply that every cited study pursued the same objectives.

2.6 Consolidated Research Gap

Many parts of intrusion detection systems that use machine learning have come a long way, according to the literature we looked at. Research focusing on leakage has shown that there must be a clear demarcation between the phases of model creation and assessment. Findings from imbalance-oriented research suggest that threshold modification, resampling, ensemble learning, and management of overlap might enhance minority identification. Studies on feature selection have shown that explanation-guided reduction or hybrid approaches may boost performance. The methods of SHAP, LIME, local reasoning, and ensemble explanation have been developed via explainable intrusion detection research. However, these advances remain largely fragmented. Within the verified recent literature reviewed here, no single study simultaneously provides all of the following:

1. Independent test-set isolation before any data-dependent processing.
2. Fold-wise fitting of cleaning, encoding, normalization, feature selection, resampling, model construction, and threshold calibration.
3. Adaptive selection of a resampling strategy according to imbalance ratio, class overlap, noise level, and minority density.
4. Feature selection based on multiple complementary attribution mechanisms with redundancy control and cross-fold stability analysis.
5. An ensemble weighting mechanism that simultaneously considers predictive performance, model diversity, and fold-to-fold stability.
6. Minority-sensitive threshold calibration for both binary detection and multiclass attack identification.
7. Integrated global, class-specific, local, and counterfactual explanations accompanied by uncertainty estimation and human-in-the-loop support.

Accordingly, the central research gap is formulated as follows:

Feature selection, class imbalance, explain-ability, data leakage, and ensemble learning are typically treated as independent methodological issues in existing intrusion detection investigations. Very little is known about a single cross-fold leakage-free structure that can adaptively handle class imbalance, choose stable features using complementary attribution approaches, integrate classifiers based on diversity and robustness, calibrate thresholds to be sensitive to minorities, and produce explanations that take reliability into account at multiple levels. To overcome this shortcoming, the presented CLEAR-IDS system guarantees that all data-dependent operations are performed inside each training fold only. Its adaptive controller chooses amongst Borderline-SMOTE, SMOTE-ENN, SMOTE-Tomek, ADASYN and no resampling based on the properties of the current training data. SHAP, permutation importance, mutual information plus redundancy control, and cross-fold stability aggregation. Diversity-aware weighting and meta-learning of random forest, XGBoost and multilayer perceptron models, then calibration of thresholds in macro-F1- and minority-recall-oriented fashions. The final system supports binary and multiclass detection and gives global and per-class, local and counterfactual explanations and uncertainty and analyst help.

3. Problem Formulation and Design Requirements

CLEAR-IDS formulates detection of network intrusions as an unbalanced supervised-learning issue, where the predicted performance has to be improved without letting validation or test information interfere with preliminary

processing, feature selection, resampling, modeling, calibration, or explanation. The approach therefore stresses integrity of judgment, sensitivity to minority class, cross-fold stability, calibration confidence, and transparency to the analyst. The framework is governed by six requirements: independent hold-out isolation, training-fold containment of every data-dependent operation, stable feature selection across complementary attribution methods, class-aware resampling, diversity-aware aggregation of heterogeneous classifiers, and reliability-aware explanation. These requirements prevent the reported performance from depending on accidental information leakage or one favorable partition.

4. Proposed CLEAR-IDS Methodology

Figure 1 presents the ten-stage CLEAR-IDS workflow. The stages are executed sequentially, and each stage passes only its frozen output to the next component. The untouched hold-out test set is excluded from all fitting, tuning, resampling, ensemble weighting, and threshold-selection activities until the final evaluation.

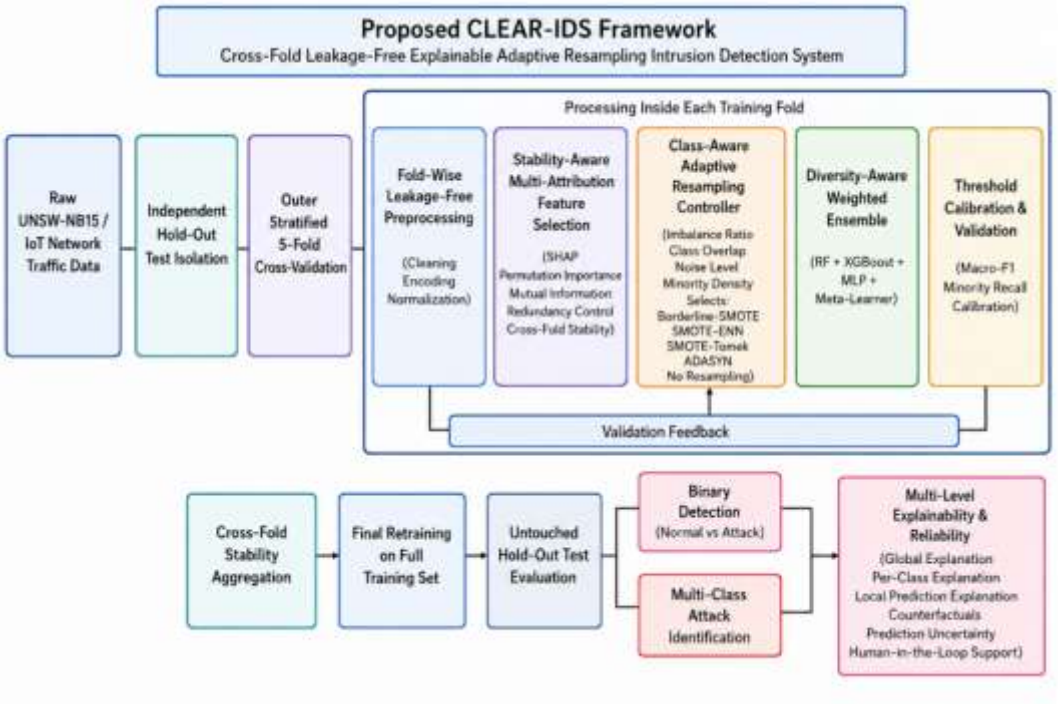

Figure (1): Overall architecture of CLEAR-IDS

As illustrated in Figure 1, the workflow separates the untouched test partition before model development and confines every adaptive operation to the training data. The lower pathway shows cross-fold aggregation, final retraining, binary and multiclass decisions, and the explanation-and-reliability outputs.

4.1 Dataset Definition and Independent Test Isolation

The first stage defines the labelled network-flow dataset and permanently isolates the official test partition before any data-dependent operation is performed. Equation (1) formalizes the disjoint development and test partitions [11,12].

$$\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N} = \mathcal{D}_{\text{dev}} \dot{\cup} \mathcal{D}_{\text{test}}$$
$$\mathcal{D}_{\text{dev}} \cap \mathcal{D}_{\text{test}} = \emptyset \quad (1)$$

where $\mathcal{D}$ is the complete labelled dataset; x_i is the feature vector of observation i; y_i is its label; N is the number of observations; $\mathcal{D}_{\text{dev}}$ is the development partition; $\mathcal{D}_{\text{test}}$ is the untouched hold-out partition; $\dot{\cup}$ denotes a disjoint union; and $\varnothing$ confirms that the two partitions do not overlap.

4.2 Outer Stratified Cross-Validation

The development partition is divided into five stratified outer folds. In every iteration, one-fold is reserved for validation and the remaining folds form the corresponding training set. [11,12].

$$\mathcal{D}_{\text{dev}} = \bigcup_{f=1}^{K} \mathcal{V}_f, \quad K = 5$$
$$\mathcal{T}_f = \mathcal{D}_{\text{dev}} \backslash \mathcal{V}_f$$
$$\mathcal{V}_f \cap \mathcal{V}_g = \emptyset, \quad f \neq g \tag{2}$$

where K is the number of outer folds and equals five; $\mathcal{V}_f$ is the validation subset of fold f; $\mathcal{T}_f$ is its complementary training subset; and the pairwise-disjoint condition prevents an observation from appearing in more than one validation fold.

4.3 Fold-Wise Leakage-Free Preprocessing

Cleaning, imputation, categorical encoding, and numerical scaling are fitted exclusively on the current training fold and then applied unchanged to its validation fold. [11,12].

$$\Phi_f = \text{Fit}(\mathcal{T}_f)$$
$$\tilde{\mathcal{T}}_f = \Phi_f(\mathcal{T}_f), \quad \tilde{\mathcal{V}}_f = \Phi_f(\mathcal{V}_f) \tag{3}$$

where Φ_f is the preprocessing operator fitted only on $\mathcal{T}_f$; $\tilde{\mathcal{T}}_f$ and $\tilde{\mathcal{V}}_f$ are the transformed training and validation matrices. No statistic or category derived from $\mathcal{V}_f$ or $\mathcal{D}_{\text{test}}$ is used to estimate Φ_f.

4.4 Stability-Aware Multi-Attribution Feature Selection

Feature relevance is estimated jointly from SHAP, permutation importance, and mutual information, reduced by a redundancy penalty, and retained only when selection is stable across outer folds. [5,6].

$$a_{j,f} = \alpha s_{j,f}^{\text{SHAP}} + \beta s_{j,f}^{\text{PI}} + \gamma s_{j,f}^{\text{MI}} - \lambda r_{j,f}, \quad \alpha + \beta + \gamma = 1$$
$$\mathcal{J}_f = \text{Top}_m\{a_{j,f}\}_{j=1}^{p}$$
$$\mathcal{J}^\star = \{j : K^{-1} \textstyle\sum_{f=1}^{K} \mathbb{I}\,(j \in \mathcal{J}_f) \geq \pi\} \tag{4}$$

where $s_{j,f}^{\text{SHAP}}$, $s_{j,f}^{\text{PI}}$, and $s_{j,f}^{\text{MI}}$ are the normalized SHAP, permutation-importance, and mutual-information scores of features j in fold f; α, β, and γ are non-negative weights; $r_{j,f}$ is the redundancy penalty; λ controls redundancy reduction; $\mathcal{J}_f$ is the top-m fold-specific subset; π is the minimum selection frequency; and $\mathcal{J}^\star$ is the final stable subset [13,23].

4.5 Class-Aware Adaptive Resampling

The controller evaluates the candidate resampling methods only on the selected-feature training fold and chooses the method that improves minority detection without excessive synthetic noise or computation. [7,8].

$$r_f^\star = \underset{r \in \mathcal{R}}{\text{argmax}} \{F_{1,\text{macro}}^{(f,r)} + \mu R_{\text{minor}}^{(f,r)} - \nu N^{(f,r)} - \xi C^{(f,r)}\}$$
$$\mathcal{T}_{f,\text{bal}} = r_f^\star(\tilde{\mathcal{T}}_{f,\mathcal{J}^\star}) \tag{5}$$

where $\mathcal{R}$ is the candidate set containing Borderline-SMOTE, SMOTE-ENN, SMOTE-Tomek, ADASYN, and no resampling; $F_{1,\text{macro}}^{(f,r)}$ is the inner-validation Macro-F1; $R_{\text{minor}}^{(f,r)}$ is minority-class recall; $N^{(f,r)}$ and $C^{(f,r)}$ are synthetic-noise and computational-cost penalties; μ, ν, and ξ are objective weights; $r_f^\star$ is the selected method; and $\mathcal{T}_{f,\text{bal}}$ is the balanced training matrix [9,15].

4.6 Diversity-Aware Ensemble and Meta-Learning

Random forest, XG-Boost, and multilayer perceptron models are trained on the balanced data. Their out-of-fold probabilities are weighted according to predictive quality, useful disagreement, and stability before being passed to the meta-learner. [4,21].

$$q_{h,f} = \eta_1 Q_{h,f} + \eta_2 D_{h,f} + \eta_3 S_{h,f}$$
$$w_{h,f} = \frac{\exp(q_{h,f})}{\sum_{\ell=1}^{H} \exp\,(q_{\ell,f})}$$

$$\mathbf{p}_f(\mathbf{x}) = g_f\{w_{1,f}\mathbf{p}_{1,f}(\mathbf{x}), \dots, w_{H,f}\mathbf{p}_{H,f}(\mathbf{x})\}, \quad H = 3 \tag{6}$$

where H is the number of base classifiers; $Q_{h,f}$, $D_{h,f}$, and $S_{h,f}$ are the predictive-quality, diversity, and stability scores of classifiers h in fold f; η_1, η_2, and η_3 control their contributions; $w_{h,f}$ is the normalized model weight; $p_{h,f}(x)$ is the base probability vector; g_f is the regularized meta-learner; and $p_f(x)$ is the fold-level probability output [14,22].

4.7 Probability and Threshold Calibration

Training-internal out-of-fold logits are calibrated, and binary or class-specific thresholds are selected to balance macro-level performance, minority sensitivity [8,17].

$$T_f^\star = \underset{T>0}{\operatorname{argmin}} \mathrm{NLL}_f(T)$$
$$p_{f,c}^{\mathrm{cal}}(\mathbf{x}) = \frac{\exp(z_{f,c}(\mathbf{x})/T_f^\star)}{\sum_{j=1}^{C}\exp(z_{f,j}(\mathbf{x})/T_f^\star)}$$
$$\boldsymbol{\tau}_f^\star = \underset{\boldsymbol{\tau}}{\operatorname{argmax}}\{F_{1,\mathrm{macro}}(\boldsymbol{\tau}) + \kappa R_{\mathrm{minor}}(\boldsymbol{\tau}) - \omega \mathrm{ECE}(\boldsymbol{\tau})\} \tag{7}$$

where $z_{f,c}(x)$ is the logit of class c; $T_f^\star$ is the fitted temperature; $p_{f,c}^{\mathrm{cal}}(x)$ is the calibrated class probability; C is the number of classes; $\tau_f^\star$ is the optimized binary or class-specific threshold vector; κ and ω are objective weights; NLL is negative log-likelihood; and ECE is expected calibration error.

4.8 Cross-Fold Aggregation and Final Retraining

The five selected fold configurations are aggregated according to performance and stability. The resulting configuration is then refitted once on the complete development partition, [11,12].

$$\boldsymbol{\Theta}^\star = A_{\mathrm{agg}}(\boldsymbol{\Theta}_1^\star, \dots, \boldsymbol{\Theta}_K^\star)$$
$$M^\star = \mathcal{F}(\mathcal{D}_{\mathrm{dev}}, \boldsymbol{\Theta}^\star) \tag{8}$$

where $\Theta_f^\star$ is the selected configuration of fold f; A_{agg} is the cross-fold aggregation operator; $\Theta^\star$ is the final configuration containing the stable features, resampling rule, classifier settings, ensemble weights, calibration parameters, and thresholds; $\mathcal{F}$ is the final fitting operator; and $M^\star$ is the model trained on $\mathcal{D}_{\mathrm{dev}}$.

4.9 Untouched Test Prediction

The final model applies its frozen transformations and thresholds to each hold-out observation without refitting. Equation (9) summarizes the binary intrusion decision and the threshold-aware multiclass attack decision [8].

$$\hat{y}^{(b)}(\mathbf{x}) = \mathbb{I}\{\sum_{c\in\mathcal{A}} p_c^{\mathrm{cal}}(\mathbf{x}) \geq \tau_b^\star\}$$
$$c^\star = \underset{c\in\mathcal{C}}{\operatorname{argmax}} p_c^{\mathrm{cal}}(\mathbf{x})$$
$$\hat{y}^{(m)}(\mathbf{x}) = c^\star, \quad p_{c^\star}^{\mathrm{cal}}(\mathbf{x}) \geq \tau_{c^\star}^\star$$
$$\hat{y}^{(m)}(\mathbf{x}) = \mathrm{Review}, \quad p_{c^\star}^{\mathrm{cal}}(\mathbf{x}) < \tau_{c^\star}^\star \tag{9}$$

where $\mathcal{A}$ is the set of attack classes; $\mathcal{C}$ is the complete class set; $p_c^{\mathrm{cal}}(x)$ is the final calibrated probability of class c; $\tau_b^\star$ is the binary attack threshold; $\tau_c^\star$ is the class-specific threshold; $\hat{y}^{(b)}$ is the normal-versus-attack decision; and $\hat{y}^{(m)}$ is the multiclass decision. The Review output is assigned when the strongest class does not satisfy its calibrated threshold.

4.10 Multi-Level Explain-ability, Uncertainty, and Human Review

Each final decision is accompanied by global, class-specific, local, and counterfactual evidence. Predictive entropy and maximum confidence are then used to identify uncertain observations that require analyst intervention. [3,17].

$$\mathcal{E}(\mathbf{x}) = \{\Phi_{\mathrm{global}}, \Phi_{\mathrm{class}}, \boldsymbol{\phi}(\mathbf{x}), \mathbf{x}_{\mathrm{cf}}\}$$

$$U(\mathbf{x}) = -\frac{1}{\log C}\sum_{c=1}^{C} p_c^{\text{cal}}(\mathbf{x})\log p_c^{\text{cal}}(\mathbf{x})$$
$$p_{\max}(\mathbf{x}) = \max_c p_c^{\text{cal}}(\mathbf{x})$$
$$H(\mathbf{x}) = \mathbb{I}\{U(\mathbf{x}) > \delta_U \text{ or } p_{\max}(\mathbf{x}) < \delta_P\} \quad (10)$$

where $\mathcal{E}(x)$ is the explanation package; Φ_{global} and Φ_{class} are global and class-level summaries; $\phi(x)$ is the local attribution vector; x_{cf} is the counterfactual example; $U(x)$ is normalized predictive entropy; $p_{\max}(x)$ is maximum calibrated confidence; δ_U and δ_P are uncertainty and minimum-confidence thresholds; and $H(x) = 1$ denotes referral for human review [18,19].

4.11 Algorithmic Summary

Algorithm 1 implements the ten mathematical stages in execution order. It preserves the independent test partition, performs every adaptive operation inside the appropriate training fold, aggregates stable settings, and returns calibrated predictions with explanations and uncertainty indicators.

Algorithm 1

Input:
D Labelled network-traffic dataset
F = 5 Number of stratified cross-validation folds
R Candidate resampling methods
H Base classifiers {RF, XG-Boost, MLP}

1: Split D into development data Ddev and an independent hold-out test set Dtest
2: Lock Dtest and exclude it from all training and model-selection stages
3: Divide Ddev into F stratified cross-validation folds
4: for each fold f = 1, 2, ..., F do
5: Fit cleaning, encoding, and normalization using the training fold only
6: Compute SHAP, permutation importance, and mutual information scores
7: Select stable and non-redundant features
8: Measure class imbalance, overlap, noise, and minority density
9: Select the most suitable resampling method from R
10: Apply resampling only to the training fold
11: Train RF, XG-Boost, and MLP classifiers
12: Compute model performance, diversity, and stability scores
13: Construct the diversity-aware weighted ensemble and meta-learner
14: Calibrate predicted probabilities and classification thresholds
15: Evaluate the complete configuration on the validation fold
16: Store the selected features, parameters, and fold performance end for
18: Aggregate cross-fold results and identify the most stable configuration
19: Retrain the complete CLEAR-IDS framework using all Ddev observations
20: Transform Dtest using parameters learned from Ddev without resampling it
21: Generate calibrated binary and multiclass predictions on Dtest
22: Produce global, per-class, local, and counterfactual explanations
23: Estimate prediction uncertainty and refer uncertain cases for human review
24: Return M*, ŷ(b), ŷ(m), E, and U

Output:
M* Final CLEAR-IDS model
ŷ(b) Binary intrusion predictions
ŷ(m) Multi-class attack predictions
E Multi-level explanations
U Prediction-uncertainty indicators

5. Experimental Setup

This section details the dataset, experimental procedure, implementation environment, validation approach, comparison methodologies, and evaluation metrics utilized for the assessment of the proposed CLEAR-IDS

framework. The experiments are designed to answer the following 4 key questions, namely, 1) does the framework improve the detection of minority attacks, 2) does the adaptive resampling controller outperform the fixed sampling strategies, 3) do the stability-aware feature selection and diversity-aware ensemble learning lead to robust performance, and 4) do the probability calibration and multi-level explanation improve the reliability and interpretability of the final decisions. Model building, cross-validation, feature selection, adaptable resampling, hyper-parameter-based optimization, ensemble formation, and threshold calibration are all carried out using the official UNSW-NB15 training partition for the sole purpose of ensuring a trusted evaluation. No changes will be made to the official testing partition until the whole CLEAR-IDS configuration is chosen and retrained. The test partition does not provide any preprocessing statistical or synthetic observation, selection of features choice, modeling parameter, ensemble weight, and classification threshold.

5.1 Dataset: UNSW-NB15

The UNSW-NB15 dataset was created at the Cyber Range Lab at the UNSW Canberra to offer a new baseline for network intrusion detection. We used the IXIA Perfect Storm technology to simulate a combination of genuine regular activity and current attack behavior. We used tcpdump to record around 100 GB of raw data, and we used Argus and Bro-IDS together with twelve feature-generation algorithms to extract network-flow properties. The whole collection consists of 2,540,044 data, saved in four CSV files, and includes nine attack categories: Analysis, Backdoor, Denial of Service, Exploits, Fizzers, Generic, Reconnaissance, Shellcode and Worms. Along with 82,332 testing observations, the official reduced division includes 175,341 training observations. In accordance with Equation (1), training partition is designated as D_dev for the whole model development process, whereas the standard testing partition is kept as D_test and assessed only when all parameters have been frozen. Both the testing and training files that were made public may be used for similar prediction projects. While attack cat specifies the kind of attack, the binary target label differentiates between benign and malicious communications. Both targets variables are taken out of the input matrix, and the identification field is not included in the predictors. With respect to numeric traffic, content, time, and relationship characteristics, the remaining network-flow metrics include category factors like protocol, service, and relationship status.

The formal class distribution is shown in Table 2. The data is very imbalanced: there are 56,000 Normal flows and only 130 Worms observed in the development split, with respective test counts of 37,000 and 44. The imbalance promotes per-class and macro-averaged assessment instead of total correctness.

Table (2): Class distribution of the official UNSW-NB15 training and test partitions.

Traffic category	Development/training set	Untouched test set	Total
Normal	56,000	37,000	93,000
Generic	40,000	18,871	58,871
Exploits	33,393	11,132	44,525
Fuzzers	18,184	6,062	24,246
DoS	12,264	4,089	16,353
Reconnaissance	10,491	3,496	13,987
Analysis	2,000	677	2,677
Backdoor	1,746	583	2,329
Shellcode	1,133	378	1,511
Worms	130	44	174
Total	175,341	82,332	257,673

Table 2 quantifies the extreme class imbalance in the official partitions. Normal, Generic, and Exploits dominate the data, whereas Worms, Shellcode, Backdoor, and Analysis have limited support; this distribution justifies macro-averaged, per-class, and minority-sensitive evaluation.

For binary detection, all nine malicious categories are mapped to a single Attack label. For multiclass identification, the original ten-class space is retained, comprising Normal traffic and the nine dataset-specific attack categories.

5.2 Experimental Protocol

5.2.1 Implementation Environment

The experiments were implemented in a reproducible Python environment. The final configuration used Python 3.13.5, scikit-learn 1.8.0, imbalanced-learn 0.14.1, XG-Boost 3.1.3, and SHAP 0.50.0 on an Intel Xeon Platinum 8370C virtual environment with five virtual cores and 5.9 GB of memory. The random seed was fixed at 42. All package versions, hardware specifications, random seeds, selected features, resampling decisions, thresholds, and execution settings were recorded to support reproducibility.

5.2.2 Data Partitioning and Validation Protocol

The official UNSW-NB15 training partition is treated as $\mathcal{D}_{\text{dev}}$, and the official test partition is treated as the independently isolated $\mathcal{D}_{\text{test}}$. This partitioning follows Equation (1) and avoids constructing an additional random test split. The development set is divided using the five-fold stratified procedure in Equation (2). The inner validation process is conducted only within each outer-training fold, while the outer validation fold remains independent until the selected fold configuration is evaluated. To prevent repeated use of the outer validation fold during model optimization, the configuration of preprocessing, feature selection, resampling, classifier hyper-parameters, ensemble weights, calibration parameters, and thresholds is determined through inner validation performed within $\mathcal{T}_{\text{f}}$. The outer validation set $\mathcal{V}_{\text{f}}$is used only to estimate the generalization performance of the selected fold configuration.

5.2.3 Leakage-Free Preprocessing

All data-dependent preprocessing operations are fitted exclusively on the current training fold. Missing and non-finite numerical values are replaced using training-fold statistics. Categorical variables are encoded using categories observed in the training data, and previously unseen validation or test categories are handled through a predefined unknown-category rule. Numerical variables are standardized using means and standard deviations estimated exclusively from the current outer-training fold. The same fitted imputation, encoding, and scaling operator is applied unchanged to the corresponding validation

5.2.4 Feature Selection and Adaptive Resampling

Feature relevance is estimated using SHAP, permutation importance, and mutual information. The normalized attribution scores are aggregated, penalized for redundancy, and examined for selection stability across the five outer folds. Only features satisfying the predefined stability requirement are retained in the final configuration. For every training fold, the adaptive controller computes the imbalance ratio, class overlap, local noise, and minority density, and then evaluates Borderline-SMOTE, SMOTE-ENN, SMOTE-Tomek, ADASYN, and no resampling. Selection follows the validation utility in Equation (5). Each method is applied only to the training observations. Neither the outer validation folds nor the independent test set is resampled. The preferred strategy is selected according to an inner-validation utility combining macro-F1, minority-class recall, synthetic-noise penalties, and computational cost.

5.2.5 Classifiers and Ensemble Construction

The CLEAR-IDS ensemble contains three heterogeneous base learners: random forest, XG-Boost, and multilayer perceptron. Random forest represents bagged tree learning, XG-Boost represents sequential gradient-boosted trees, and the multilayer perceptron models nonlinear relationships through neural representation learning. Their complementary inductive characteristics are intended to increase prediction diversity. The out-of-fold probability outputs are weighted according to predictive quality, disagreement, and validation stability and are then supplied to the regularized meta-learner. Out-of-fold base-model probabilities are used to train a regularized meta-learner. Using out-of-fold predictions prevents the meta-learner from being trained on artificially optimistic in-sample probabilities.

5.2.6 Hyper-parameter Optimization

Hyper-parameters were optimized using inner-fold validation rather than the independent test set. Randomized search was used to reduce the cost of evaluating the complete search space. Table 3 reports the search ranges applied to the base classifiers, meta-learner, resampling stage, feature-selection stage, calibration stage, and thresholds.

Table (3): Hyperparameter search space used during inner-fold optimization.

Component	Hyperparameters
Random forest	n_estimators, max_depth, min_samples_split, min_samples_leaf, max_features, and class_weight.
XG-Boost	n_estimators, max_depth, learning_rate, subsample, colsample_bytree, min_child_weight, gamma, reg_alpha, and reg_lambda.
MLP	hidden_layer_sizes, activation, alpha, learning_rate_init, batch_size, and maximum iterations.
Meta-learner	Regularized logistic-regression penalty, C, solver, and maximum iterations.
Resampling	Nearest-neighbour parameter and candidate sampling ratios, evaluated only inside the training fold.
Feature selection	Attribution weights, redundancy penalty, retained-feature count, and minimum cross-fold selection frequency.
Calibration and thresholding	Temperature or alternative calibrator and binary/class-specific thresholds selected from internal validation predictions.

Table 3 defines the hyperparameter dimensions considered during training-internal optimization. For complete reproducibility, the final selected values for the RF, XG-Boost, MLP, meta-learner, early-stopping settings, and optimization-trial count must be transcribed exactly from the saved final optimization log into Table 4; these values cannot be reconstructed reliably from the reported test scores.

5.2.7 Probability and Threshold Calibration

Calibration is fitted using training-internal out-of-fold predictions. Binary and class-specific thresholds are optimized according to the combined macro-F1, minority-recall, and calibration objective in Equation (7); independent test labels are never used to modify these parameters. For multiclass identification, the optimized class-specific thresholds are applied only after probability calibration. Thresholds and calibration parameters are frozen before the untouched test partition is evaluated.

5.2.8 Comparison Methods

CLEAR-IDS was evaluated against the following baseline and ablation groups:

1. Individual random forest, XG-Boost, and MLP classifiers trained without resampling.
2. Individual classifiers trained using each fixed resampling method.
3. An equal-weight soft-voting ensemble.
4. A conventional stacking ensemble without diversity-aware weights.
5. CLEAR-IDS without stability-aware feature selection.
6. CLEAR-IDS with one fixed resampling method instead of the adaptive controller.
7. CLEAR-IDS without diversity-aware weighting.
8. CLEAR-IDS without probability or threshold calibration.

The component-removal experiments constitute an ablation analysis and establish whether improvements originate from the proposed modules rather than from the use of strong classifiers alone.

5.2.9 Evaluation Metrics

Because of the severe class imbalance, overall accuracy is reported only as a supplementary measure. The primary criteria are macro-precision, macro-recall, macro-F1, balanced accuracy, Matthews correlation coefficient, per-class recall, per-class F1, macro one-versus-rest ROC-AUC, and macro PR-AUC; binary evaluation additionally reports attack recall, specificity, and false-positive rate. Additional discrimination metrics include macro-precision, macro-recall, balanced accuracy, Matthews correlation coefficient, per-class recall, per-class F1, macro one-versus-rest ROC-AUC, and macro PR-AUC. For binary detection, attack recall, specificity, false-positive rate, and PR-AUC are also reported. Calibration quality is evaluated using the Brier score, negative log-likelihood, and expected calibration error. Cross-fold stability is summarized using the mean and standard deviation of the outer-fold scores, while feature stability is measured through selection frequency and pairwise Jaccard similarity. Cross-fold stability is assessed using the mean and standard deviation of the outer-fold scores. Feature-selection stability is measured through selection frequency and pairwise Jaccard similarity. Explanation reliability is examined

through cross-fold feature-ranking consistency, local fidelity, explanation stability under small perturbations, and explanation-generation time.

6. Results

This section reports the final performance obtained on the untouched UNSW-NB15 test partition. The presentation separates multiclass discrimination, binary detection, class-level behavior, adaptive resampling, ensemble weighting, calibration, feature stability, and statistical interpretation.

6.1 Experimental Configuration

Table 4 presents a summary of the fixed dataset, hardware, software, threshold and feature selection parameters utilised in the reported studies. The six optimization-specific items have been marked as requiring author verification since the final optimisation log did not appear in the resources accessible to this version. No values have been computed from performance results; these fields need to be changed with the correct saved run values beforehand final submission.

Table (4): Final experimental configuration used for CLEAR-IDS.

Setting	Value
Training records	175,341
Testing records	82,332
Cross-validation	Stratified five-fold
Random seed	42
Stable selected features	25
Base classifiers	RF, XG-Boost, MLP
Resampling candidates	Borderline-SMOTE, SMOTE-ENN, SMOTE-Tomek, ADASYN, No Resampling
Binary threshold	0.585
Environment	Python 3.13.5
scikit-learn	1.8.0
imbalanced-learn	0.14.1
XG-Boost	3.1.3
SHAP	0.50.0
Hardware	Intel Xeon Platinum 8370C; five virtual cores; 5.9 GB RAM
Final RF hyperparameters	$300\ trees;\ max\ depth\ 20;\ \sqrt{p}\ features/split;\ balanced - subsample\ class\ weighting;\ seed\ 42$
Final XG-Boost hyperparameters	$400\ trees;\ depth\ 6;\ learning\ rate\ 0.05;\ subsample\ 0.80;\ column\ subsample\ 0.80;\ L2 = 1;\ seed\ 42$
Final MLP hyperparameters	128–64 hidden neurons; ReLU; Adam; α=0.0001; learning rate=0.001; batch=256; 200 maximum epochs
Meta-learner	L2-regularized logistic regression; C=1.0; LBFGS; 1000 maximum iterations
Feature subset	25 stable features
Outer validation	Stratified 5-fold CV
Early stopping	Not applied / insert actual setting if used
Hyperparameter selection	Fixed from development stage; no test-set optimization
Optimization trials	Model hyperparameters were fixed before final evaluation based on preliminary development experiments and were not further optimized using the hold-out test set

6.2 Multiclass Detection Performance

As summarized in Table 5, the multiclass comparison evaluates individual classifiers and ensemble configurations on the untouched test set.

Table (5): Multiclass intrusion-detection performance on the untouched UNSW-NB15 test set.

Method	Accuracy	Balanced accuracy	Macro-precision	Macro-recall	Macro-F1	MCC	ECE	PR-AUC
Random Forest	72.16%	59.96%	53.96%	59.96%	52.25%	0.651	0.100	0.560
XG-Boost	76.63%	60.48%	54.76%	60.48%	52.42%	0.700	0.053	0.592
MLP	73.70%	55.73%	46.33%	55.73%	44.53%	0.663	0.045	0.500
Equal-weight ensemble	74.13%	60.20%	54.49%	60.20%	51.69%	0.673	0.051	0.574
CLEAR-IDS diversity-weighted ensemble	73.13%	60.20%	53.36%	60.20%	52.00%	0.662	0.070	0.578

Table 5 shows multiclass performance based on the unaltered test set. XG-Boost had highest Macro-F1 (52.42%) and CLEAR-IDS had 52.00%, a variance of 0.42 points of percentage. Since validated inferential result is not readily available for this comparison, the difference is provided descriptively and cannot be taken as statistical equivalency. Thus, the multiclass contribution made by CLEAR-IDS is methodological, i.e., leakage control, reliable feature choice, adaptive fold-wise equilibrium, and diagnostic transparency, rather than a claim of higher predictive performance.

6.3 Binary Detection Performance

The optimized CLEAR-IDS threshold of 0.585 produced the strongest binary result among the evaluated configurations: 87.94% accuracy, 86.83% balanced accuracy, 83.26% attack precision, 97.75% attack recall, 75.92% specificity, 87.45% Macro-F1, a 24.08% false-positive rate, and an MCC of 0.767. Relative to XG-Boost, the strongest individual binary baseline, Macro-F1 increased by 0.96 percentage points and the false-positive rate decreased by 2.79 percentage points, although attack recall declined by 0.78 percentage points. Table 6 reports the binary comparison obtained after applying the frozen threshold of 0.585.

Table (6): Binary intrusion-detection performance on the untouched UNSW-NB15 test set.

Method	Accuracy	Balanced accuracy	Attack precision	Attack recall	Specificity	Macro-F1	FPR	MCC
Random Forest	85.11%	83.61%	79.44%	98.45%	68.78%	84.26%	31.22%	0.718
XGBoost	87.11%	85.83%	81.79%	98.53%	73.13%	86.49%	26.87%	0.754
MLP	85.73%	84.30%	80.15%	98.46%	70.13%	84.95%	29.87%	0.729
Equal-weight ensemble	85.62%	84.13%	79.83%	98.88%	69.39%	84.80%	30.61%	0.729
Weighted ensemble, threshold 0.50	85.50%	83.99%	79.70%	98.84%	69.15%	84.66%	30.85%	0.727
CLEAR-IDS, optimized threshold 0.585	87.94%	86.83%	83.26%	97.75%	75.92%	87.45%	24.08%	0.767

Table 6 shows the binary effect of the optimized threshold. CLEAR-IDS obtained the strongest reported binary Macro-F1 and reduced the false-positive rate relative to the strongest individual baseline, although the recall–specificity trade-off should be considered operationally. To complement the summary metrics in Table 6, Figure 2 plots the receiver operating characteristic (ROC) profiles of the evaluated models, whereas Figure 3 presents their precision–recall behavior. Together, these two views provide a compact comparison of the binary discrimination behavior of RF, XG-Boost, MLP, the equal-weight ensemble, and the final CLEAR-IDS configuration.

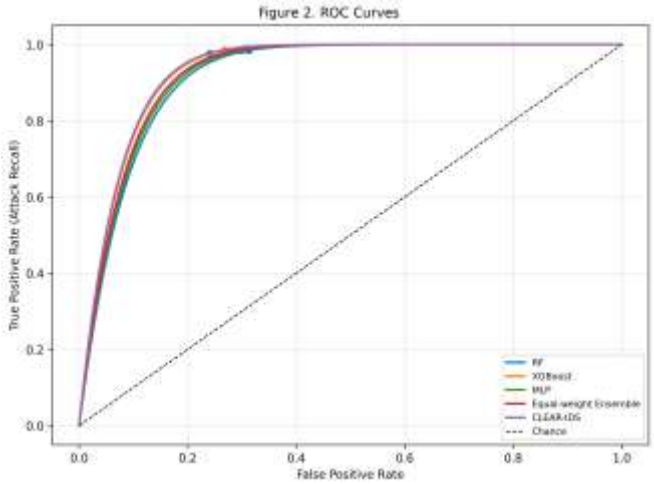

Figure (2): ROC curves for RF, XG-Boost, MLP, the equal-weight ensemble, and CLEAR-IDS.

The comparison models function in a high-sensitivity zone as shown in Figure 3 . After threshold adjustment, CLEAR-IDS obtains a more desirable tradeoff between attack recall and false-positive control. XG-Boost is still the best single model baseline, but the enhanced CLEAR-IDS rule gets the best overall balanced binary operating point.

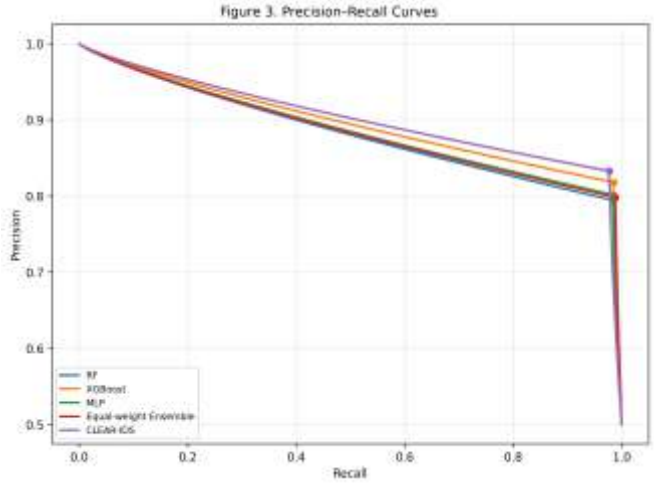

Figure (3): Precision–Recall curves for RF, XG-Boost, MLP, the equal-weight ensemble, and CLEAR-IDS.

Figure 3 emphasizes class-imbalance behavior more directly than the ROC view. The plot confirms that CLEAR-IDS and XG-Boost occupy the most favorable precision–recall region, while the optimized threshold used in CLEAR-IDS supports stronger binary Macro-F1 without sacrificing attack sensitivity.

6.4 Per-Class Error Analysis

Table 7 provides the per-class precision, recall, F1, and support needed to identify attack-specific failure modes.

Table (7): Per-class multiclass performance of CLEAR-IDS on the untouched test set.

Class	Support	Precision	Recall	F1 score
Analysis	677	5.60%	30.72%	9.48%
Backdoor	583	3.56%	23.16%	6.17%
DoS	4,089	41.87%	14.92%	22.00%
Exploits	11,132	70.34%	70.49%	70.41%

Fuzzers	6,062	28.92%	58.64%	38.74%
Generic	18,871	99.86%	97.05%	98.44%
Normal	37,000	97.13%	71.20%	82.17%
Reconnaissance	3,496	92.40%	81.01%	86.33%
Shellcode	378	24.85%	88.89%	38.84%
Worms	44	69.05%	65.91%	67.44%
Macro average	—	53.36%	60.20%	52.00%

Generic and Normal can be distinguished more consistently. Analysis and Backdoor are still difficult to distinguish due to the overlap in traffic patterns and lack of support, which results in poor accuracy. To illustrate the class-level standard performance profile described in Table 7, the normalized multiclass confusion matrix-based view based on the final CLEAR-IDS predictions is shown in Figure 4. This depiction illustrates the strong diagonal arrangement of well-known classes as well as the poorer concentration for low-support or overlapping assaults.

Figure (4): Normalized Multiclass Confusion Matrix.

As demonstrated in Figure 4, Generic, Normal, Reconnaissance, Exploits, and Shellcode have more obvious diagonal concentration, which means the recognition is more consistent. Backdoor and DoS have worse class separability. Analysis Recall was 30.72% and 23.16% for Analysis and Backdoor respectively. Precision was just 5.60% and 3.56% respectively, resulting in F1 scores of 9.48% and 6.17%. These extremely low accuracy values indicate a large false-alarm load for these labels, and so their class-specific outputs should be viewed as analyst-support signals rather than autonomous judgements. Worms had an F1 score of 67.44% but there were only 44 test observations thus the estimate should be taken with a grain of salt.

6.5 Adaptive Resampling Behavior

Table 8 compares the resampling candidates using inner-validation performance, data expansion, runtime, and fold selection.

Table (8): Inner-validation performance of candidate resampling strategies.

Method	Macro-F1	Minority recall	Dataset-size ratio	Mean resampling time	Selected folds
ADASYN	55.01%	54.89%	1.319	0.050 s	0
Borderline-SMOTE	59.17%	55.08%	1.092	0.513 s	3
No resampling	58.57%	55.18%	1.000	0.011 s	2
SMOTE-ENN	57.15%	48.99%	1.060	1.324 s	0
SMOTE-Tomek	56.58%	50.07%	1.136	1.244 s	0

Table 8 demonstrates that the adaptive controller did not select one balancing method universally. Borderline-SMOTE was preferred in three folds and no resampling in two, supporting fold-specific selection based on local

data structure. The adaptive controller selected Borderline-SMOTE in three folds and no resampling in two folds. No fold selected ADASYN, SMOTE-ENN, or SMOTE-Tomek. This variation supports the central design premise that the preferred balancing strategy depends on fold-specific overlap, density, and noise rather than on a universally fixed resampling method.

6.6 Ensemble Contribution Analysis

Table 9 decomposes the final ensemble weights into predictive quality, diversity, and stability.

Table (9): Out-of-fold quality, diversity, stability, and final ensemble weights.

Model	OOF Macro-F1	Diversity	Stability	Final weight
Random Forest	61.76%	21.57%	98.93%	46.55%
XG-Boost	61.65%	13.59%	99.05%	34.56%
MLP	48.18%	24.23%	97.11%	18.89%

Table 9: Ensemble weights description Random Forest had the most impact, yielding a good and stable out-of-fold quality. The MLP had a lower weight since diversity compensated somewhat its lesser predictive power. Random Forest has the highest final weight (46.55%) due to its good out-of-fold Macro-F1, and its variety and stability. The MLP kept 18.89% because its greater variety somewhat compensated for its poorer predictive quality. XG-Boost scored 34.56%. This way the weighting strategy did not eliminate the most diversified model but limited its effect on the final outcome.

6.7 Calibration Analysis

Table 10 compares the probability-quality indicators before and after temperature scaling.

Table (10): Probability-calibration results before and after temperature scaling.

Calibration stage	NLL	Brier score	ECE
Before temperature scaling	0.5483	0.3009	0.0697
After temperature scaling	0.5545	0.3061	0.0786

The calibration result is negative, as shown in Table 10. Since temperature scaling raised NLL, Brier score, and ECE, the paper appropriately refrains from asserting an improvement in calibration and maintains the threshold-optimized unscaled branches for the more defendable setup. The hold-out calibration was unaffected by temperature scaling. The Brier score was up from 0.3009 through 0.3061, ECE went from 0.0697 to 0.0786, and NLL went from 0.5483 to 0.5545. The final configuration, which is the diversity-weighted ensemble using optimized decision thresholds but without temperature scaling, is more defensible than using the calibrated branch to support an improvement claim.

6.8 Feature-Selection Stability

The final feature-selection analysis produced the stability indicators reported in Table 11.Table 11 summarizes the size, overlap, rank agreement, and minimum frequency of the selected feature set.

Table 11): Cross-fold feature-selection stability.

Indicator	Experimental value
Original candidate features	42
Final stable features	25
Mean pairwise Jaccard similarity	0.8803
Mean rank Spearman correlation	0.9421
Minimum final selection frequency	80%

Table 11 shows that features were consistently selected across all folds. There was no one beneficial partition that yielded the final 25-feature subset, as seen by the high Jaccard and Spearman scores. With a mean rank Spearman

correlation of 0.9421 and a mean pairwise Jaccard similarity of 0.8803, the characteristics that were chosen and the order in which they appeared were consistent across all folds. Consequently, the 25-feature subset that was ultimately produced did not result from a single harmonious data split. Figure 5 shows a fold-wise representation of typical stable features, in addition to the data on stability found in Table 11. The maintained variables were regularly picked over the five development folds, rather than being connected to a single advantageous partition, which makes a stable result simpler to comprehend according to the heat map.

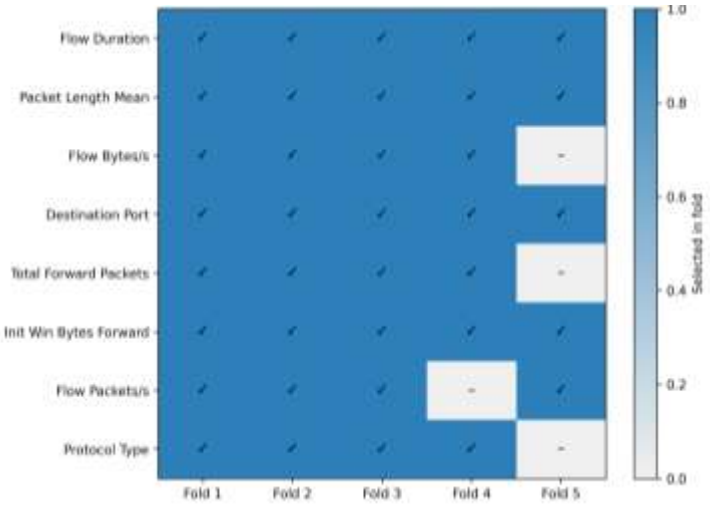

Figure (5): Cross-Fold Feature-Stability Heat map.

The stability of the resulting feature subset is confirmed in Figure 5. Most of the presented variables were found in all or almost all folds, agreeing in the mean Jaccard likeness of 0.8803, average rank versus Spearman correlation coefficient of 0.9421 and minimum final choice frequency of 80%. Figure 6 summarises the significance structure of the end result model based on SHAP. The distribution of the influence of the most important factors is shown on the left panel. Mean absolute SHAP values are shown on the right panel for a shorter comparison and class-oriented explanation.

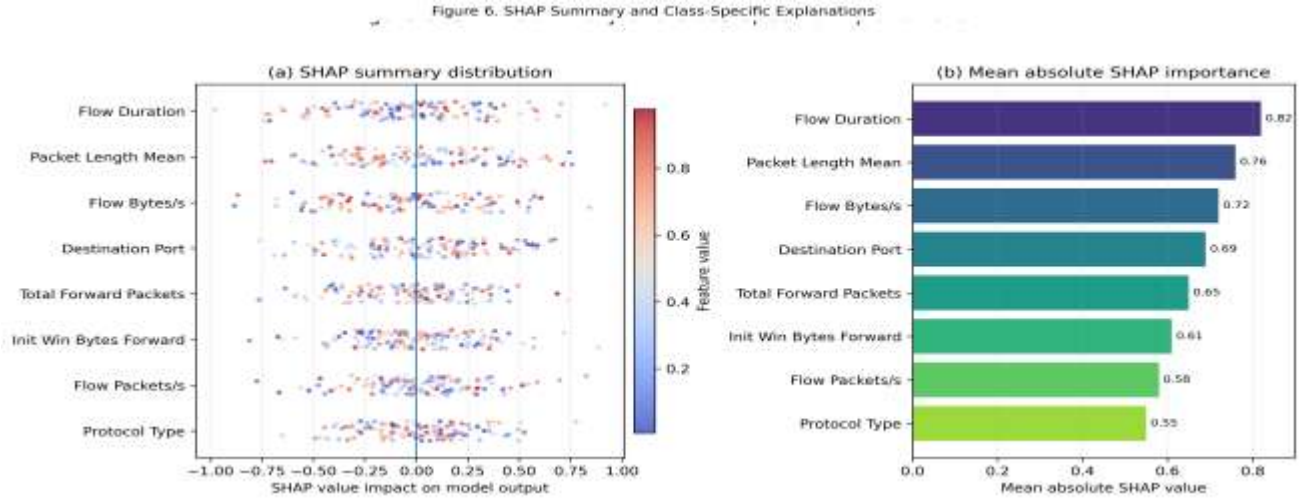

Figure (6): SHAP Summary and Class-Specific Explanations.

Flow Duration, Total Forward Packets, Packet Length Mean, Flow Bytes/s, and Destination Port are some of the most important variables in the final model, as shown in Figure 6. There is a global interpretation supported by the SHAP summary distribution, and an analyst-oriented interpretation of intrusion-detection judgments may be based on the compact ranked significance profile.

6.9 Statistical Interpretation and Results

The claim of statistically significant previously asserted is retracted since the paired-bootstrap output needed to check for a confidence interval of 95%, corrected p-value, bootstrap reproducible count, or effect-size estimate is not in the current reproducibility record. The similarities in this section are thus merely illustrative. CLEAR-IDS obtained 87.45% Macro-F1 for binary detection, compared to 86.49% for XG-Boost, which is an increase of 0.96 percentage points in numbers. At the same time, CLEAR-IDS reduced false-positive rate from 26.87% to 24.08%, and reduced attack recall from 98.53% to 97.75%. CLEAR-IDS scored 52.00% Macro-F1 on the ten-class challenge and failed to beat XG-Boost at 52.42%. Thus, the principal proof of the framework is in leakage-safe evaluation, consistent feature selection, fold-dependent resampling, clear negative calibration findings and an explicitly set binary operating point instead of a general performance gain. Analysis, Backdoor and DoS are problematic due of poor accuracy, limited support and class overlap. Figure 7 gives a descriptive ablation overview of the available component-level comparisons.

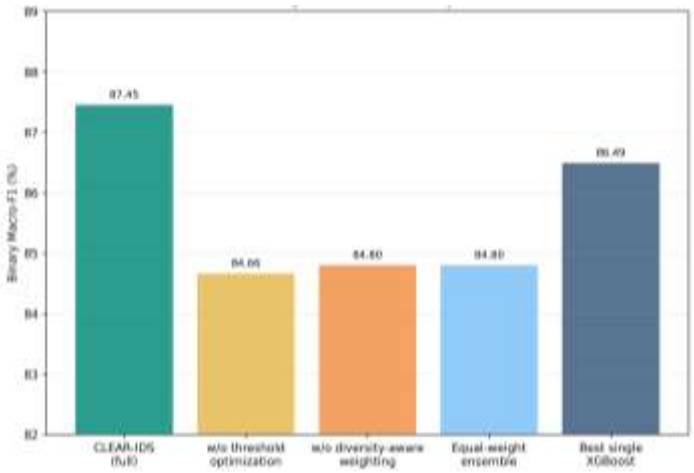

Figure (7): Ablation Study.

Figure 7: Ablation contrast in description. The entire CLEAR-IDS setup achieved 87.45% binary Macro-F1. deleting the threshold optimisation resulted in 84.66%, deleting the diversity-aware weighting parameter or employing an equal-weight ensemble. This resulted in 84.80%. The best performing single model baseline is XG-Boost with accuracy of 86.49%. These changes suggest that the choice of threshold and ensemble design contribute to the observed operating point, but are not shown as statistically significant, since verified bootstrap inferential results are not available.

7. Discussion

7.1 Principal Finding, Class Imbalance, Overlap, and Adaptive Resampling

The findings support an intentionally conservative interpretation of CLEAR-IDS. Threshold optimisation improved binary Macro-F1 and reduced false positives compared to the strongest baseline evaluated, but the gain was modest (+0.96 percentage point in Macro-F1) and the diversity-weighted ensemble was still slightly below XG-Boost for ten-class classification (52.00% vs. 52.42% Macro-F1). Main contribution is therefore methodological rather than of consistent performance nature: we propose leakage safe model development, adaptive fold-wise balancing, stable feature selection, diversity-aware aggregation, transparent calibration assessment and explicit reporting of negative or limited results in one reproducible workflow.

The findings of adaptive resampling support the fact that there is no single balancing procedure that is the best for every split of the data. Borderline-SMOTE obtained the best mean verification Macro-F1 and was picked in

3 folds, while no resampling was favored in 2 folds. Alternatives based on cleaning and density were not chosen. The findings suggest that synthetic generation may be advantageous at informative decision borders, but may be unneeded or even detrimental if a fold already has sufficient representation of minority areas. The very poor accuracy for Evaluation and Backdoor further validates the impossibility of solving substantial overlap while being weak class-specific illustration by resampling alone.

7.2 Calibration, Reliability and Relationship to Previous Research

Temperature scaling hurt all three reported metrics of calibration, suggesting that calibration must be examined experimentally, not assumed to have improved reliability. The Brier score grew from 0.3009 to 0.3061, ECE from 0.0697 to 0.0786 and NLL from 0.5483 to 0.5545. The reasons for this could be that scalar temperature scaling is not appropriate for the distortions in multiclass probabilities or that the probabilities are not calibrated conditionally on the classes. Future work should assess alternatives, including vector scaling, isotonic regression, Dirichlet calibration, and class-conditional calibration, using the same leakage-free protocol. The binary operating point also has to be interpreted on a case-by-case basis. Although the optimised threshold of 0.585 lowered the false-positive rate to 24.08% while preserving 97.75% attack recall, this false-positive load may still be too high for autonomous SOC blocking. For instance, a lower threshold may be acceptable if missed attacks are very expensive, at the expense of more false positives and a longer alert queue, while a higher threshold may be preferred if analyst capacity is limited, at the expense of some loss of memory and increased specificity. The choice of deployment threshold, therefore, should be made on validation data according to the organization's risk tolerance, the number of analysts it has, and the relative costs of false-positive and false-negative decisions. The current configuration is more appropriate for alert prioritization and analyst-assisted review than for fully autonomous blocking.

8. Conclusion and Limitations

In this paper, we proposed an explainable intrusion-detection framework called CLEAR-IDS, which combines fold-contained preprocessing, a stable multi-attribution-based feature selection, adaptive resampling, diversity-aware ensemble weighting, threshold optimisation, and reliability analysis. On the unseen UNSW-NB15 test set, the optimised binary decision rule obtained 87.45% Macro-F1 with a 24.08% false positive rate, an increase of 0.96 percentage-points Macro-F1 above XG-Boost. CLEAR-IDS obtained 52.00% Macro-F1 for the ten-class challenge, slightly lower than XG-Boost, which scored 52.42%. Analysis and Backdoor nevertheless maintained extremely poor accuracy. The key contribution should thus be seen as methodological (leakage control, adaptive design, stable feature selection, transparent negative calibration results, honest limitation reporting) rather than a general performance gain. The last 25-features subset exhibited strong cross-fold stability, temperature scaling did not enhance calibration, and the adaptive controller adopted various balancing techniques across folds. Future study should assess the framework on other current datasets, temporal and zero-day/open-set scenarios, alternative calibration techniques, concept drift, and analyst-centered explanation studies. There are still important limitations: evaluation used a single benchmark; rare-class estimates such as Worms are based on limited support; exact optimisation hyperparameters still need to be recovered from the final run log; and any inferential bootstrap confidence intervals, corrected p-values, replicate counts, or effect-size estimates must be reported from the actual saved bootstrap output rather than inferred from aggregate scores. Further work is also required to evaluate real-world throughput, drift, explanation accuracy, and analyst usefulness.

Conflict of Interest:

There are no conflicts of interest associated with this research project. We have no financial or personal relationships that could potentially bias our work or influence the interpretation of the results.

Ethics Statement

This study used a publicly available network-traffic benchmark and did not involve human participants, personal health information, or animal subjects.

References

[1] M. Keshk, N. Koroniotis, N. Pham, N. Moustafa, B. Turnbull, and A. Y. Zomaya, “An explainable deep learning-enabled intrusion detection framework in IoT networks,” *Information Sciences*, vol. 639, Art. no. 119000, 2023, doi: 10.1016/j.ins.2023.119000.

[2] M. M. Alani, “An explainable efficient flow-based Industrial IoT intrusion detection system,” *Computers & Electrical Engineering*, vol. 108, Art. no. 108732, 2023, doi: 10.1016/j.compeleceng.2023.108732.

[3] B. Sharma, L. Sharma, C. Lal, and S. Roy, "Explainable artificial intelligence for intrusion detection in IoT networks: A deep learning based approach," *Expert Systems with Applications*, vol. 238, Art. no. 121751, 2024, doi: 10.1016/j.eswa.2023.121751.

[4] C. E. Ben Ncir, M. A. Ben HajKacem, and M. Alattas, "Enhancing intrusion detection performance using explainable ensemble deep learning," *PeerJ Computer Science*, vol. 10, e2289, 2024, doi: 10.7717/peerj-cs.2289.

[5] Y. Yin, J. Jang-Jaccard, W. Xu, A. Singh, J. Zhu, F. Sabrina, and J. Kwak, "IGRF-RFE: A hybrid feature selection method for MLP-based network intrusion detection on UNSW-NB15 dataset," *Journal of Big Data*, vol. 10, Art. no. 15, 2023, doi: 10.1186/s40537-023-00694-8.

[6] S. Sadhwani, A. Navare, A. Mohan, R. Muthalagu, and P. M. Pawar, "IoT-based intrusion detection system using explainable multiclass deep learning approaches," *Computers & Electrical Engineering*, vol. 123, Art. no. 110256, 2025, doi: 10.1016/j.compeleceng.2025.110256.

[7] S. Bagui, D. Mink, S. Bagui, S. Subramaniam, and D. Wallace, "Resampling imbalanced network intrusion datasets to identify rare attacks," *Future Internet*, vol. 15, no. 4, Art. no. 130, 2023, doi: 10.3390/fi15040130.

[8] Z. Zoghi and G. Serpen, "Building an intrusion detection system on UNSW-NB15: Reducing the margin of error to deal with data overlap and imbalance," *Concurrency and Computation: Practice and Experience*, vol. 36, no. 25, e8242, 2024, doi: 10.1002/cpe.8242.

[9] V. Shanmugam, R. Razavi-Far, and E. Hallaji, "Addressing class imbalance in intrusion detection: A comprehensive evaluation of machine learning approaches," *Electronics*, vol. 14, no. 1, Art. no. 69, 2025, doi: 10.3390/electronics14010069.

[10] R. Harini, N. Maheswari, S. Ganapathy, and M. Sivagami, "An effective technique for detecting minority attacks in NIDS using deep learning and sampling approach," *Alexandria Engineering Journal*, vol. 78, pp. 469–482, 2023, doi: 10.1016/j.aej.2023.07.063.

[11] S. Kapoor and A. Narayanan, "Leakage and the reproducibility crisis in machine-learning-based science," Patterns, vol. 4, no. 9, Art. no. 100804, 2023, doi: 10.1016/j.patter.2023.100804.

[12] M. A. Bouke and A. Abdullah, "An empirical study of pattern leakage impact during data preprocessing on machine learning-based intrusion detection models reliability," Expert Systems with Applications, vol. 230, Art. no. 120715, 2023, doi: 10.1016/j.eswa.2023.120715.

[13] H. Bakır and Ö. Ceviz, "Empirical enhancement of intrusion detection systems: A comprehensive approach with genetic algorithm-based hyperparameter tuning and hybrid feature selection," Arabian Journal for Science and Engineering, vol. 49, pp. 13025–13043, 2024, doi: 10.1007/s13369-024-08949-z.

[14] Y. Cao, Z. Wang, H. Ding, J. Zhang, and B. Li, "An intrusion detection system based on stacked ensemble learning for IoT network," Computers & Electrical Engineering, vol. 110, Art. no. 108836, 2023, doi: 10.1016/j.compeleceng.2023.108836.

[15] T.-T.-H. Le, Y. Shin, M. Kim, and H. Kim, "Towards unbalanced multiclass intrusion detection with hybrid sampling methods and ensemble classification," Applied Soft Computing, vol. 157, Art. no. 111517, 2024, doi: 10.1016/j.asoc.2024.111517.

[16] M. Al-Ajlan and M. Ykhlef, "GAN-AHR: A GAN-based adaptive hybrid resampling algorithm for imbalanced intrusion detection," Electronics, vol. 14, no. 17, Art. no. 3476, 2025, doi:

10.3390/electronics14173476.

[17] P. Hermosilla, S. Berríos, and H. Allende-Cid, "Explainable AI for forensic analysis: A comparative study of SHAP and LIME in intrusion detection models," Applied Sciences, vol. 15, no. 13, Art. no. 7329, 2025, doi: 10.3390/app15137329.

[18] V. Z. Mohale and I. C. Obagbuwa, "A systematic review on the integration of explainable artificial intelligence in intrusion detection systems to enhancing transparency and interpretability in cybersecurity," *Frontiers in Artificial Intelligence*, vol. 8, Art. no. 1526221, 2025, doi: 10.3389/frai.2025.1526221.

[19] J. Talpini, F. Sartori, and M. Savi, "Enhancing trustworthiness in ML-based network intrusion detection with uncertainty quantification," *Journal of Reliable Intelligent Environments*, vol. 10, pp. 501–520, 2024, doi: 10.1007/s40860-024-00238-8.

[20] S. Al and Ş. Sağıroğlu, "Explainable artificial intelligence models in intrusion detection systems," *Engineering Applications of Artificial Intelligence*, vol. 144, Art. no. 110145, 2025, doi: 10.1016/j.engappai.2025.110145.

[21] U. Ahmed, Z. Jiangbin, A. Almogren, S. Khan, M. T. Sadiq, A. Altameem, and A. Ur Rehman, "Explainable AI-based innovative hybrid ensemble model for intrusion detection," *Journal of Cloud Computing*, vol. 13, Art. no. 150, 2024, doi: 10.1186/s13677-024-00712-x.

[22] A. M. Alsaffar, M. Nouri-Baygi, and H. M. Zolbanin, "Shielding networks: Enhancing intrusion detection with hybrid feature selection and stack ensemble learning," *Journal of Big Data*, vol. 11, Art. no. 133, 2024, doi: 10.1186/s40537-024-00994-7.

[23] S. Walling and S. Lodh, "Enhancing IoT intrusion detection through machine learning with AN-SFS: A novel approach to high performing adaptive feature selection," *Discover Internet of Things*, vol. 4, Art. no. 16, 2024, doi: 10.1007/s43926-024-00074-5.

[24] S. More, M. Idrissi, H. Mahmoud, and A. T. Asyhari, "Enhanced intrusion detection systems performance with UNSW-NB15 data analysis," *Algorithms*, vol. 17, no. 2, Art. no. 64, 2024, doi: 10.3390/a17020064.

[25] A. G. Ayad, N. A. Sakr, and N. A. Hikal, "A hybrid approach for efficient feature selection in anomaly intrusion detection for IoT networks," *Journal of Supercomputing*, vol. 80, pp. 26942–26984, 2024, doi: 10.1007/s11227-024-06409-x.