applied sciences

MDPI

Article

CoLIME with 2D Copulas for Reliable Local Explanations on Imbalanced Network Data

Mantas Bacevicius [1,*], Kristina Sutiene [2], Lukas Malakauskas [2] and Agne Paulauskaite-Taraseviciene [1]

[1] Department of Applied Informatics, Kaunas University of Technology, 51368 Kaunas, Lithuania; agne.paulauskaite-taraseviciene@ktu.lt

[2] Department of Mathematical Modeling, Kaunas University of Technology, 51368 Kaunas, Lithuania; kristina.sutiene@ktu.lt (K.S.); lukas.malakauskas@ktu.lt (L.M.)

* Correspondence: mantas.bacevicius@ktu.edu

Abstract

Local Interpretable Model-agnostic Explanations (LIME) is a widely used technique for interpreting individual predictions of complex "black-box" models by fitting a simple surrogate model to synthetic perturbations of the input. However, its standard perturbation strategy of sampling features independently from a Gaussian distribution often generates unrealistic samples and neglects inter-feature dependencies. This can lead to low local fidelity (poor approximation of the model's behavior) and unstable explanations across different runs. This paper presents CoLIME, which is a copula-based perturbation generation framework for LIME, designed to capture the underlying data distribution and inter-feature dependencies more accurately. The framework employs bivariate (2D) copula models to jointly sample correlated features while fitting suitable marginal distributions for individual features. Furthermore, perturbation localization strategies were implemented, restricting perturbations to a defined local radius and maintaining specific property values to ensure that the synthesized samples remain representative of the actual local environment. The proposed approach was evaluated on a network intrusion detection dataset, comparing the fidelity and stability of LIME under Gaussian versus copula-based perturbations, using Ridge regression as the surrogate explainer. Empirically, for the most dependent feature pairs, CoLIME increases mean surrogate fidelity by 21.84–50.31% on the merged CIC-IDS2017/2018 dataset and by 29.28–60.24% on the UNSW-NB15 dataset. Stability is similarly improved, with mean Jaccard similarity gains of 3.78–5.45% and 1.95–2.12%, respectively. These improvements demonstrate that dependency-preserving perturbations provide a significantly more reliable foundation for explaining complex network intrusion detection models.

Keywords: LIME; Explainable AI (XAI); explanation fidelity; copula models; network intrusion

Check for updates

Academic Editors: Guoyang Liu, Weidong Zhou and Lan Tian

Received: 22 November 2025
Revised: 16 December 2025
Accepted: 17 December 2025
Published: 22 December 2025

1. Introduction

The growing complexity of cyber threats presents major challenges for network security, underscoring the importance of developing effective and accurate intrusion detection systems. Traditional approaches often struggle to handle highly imbalanced datasets, redundant or irrelevant features, and limited generalizability across diverse attack scenarios. Consequently, artificial intelligence (AI) and machine learning models have become central to modern cybersecurity, supporting intrusion detection, anomaly recognition, and threat prediction. However, intrusion detection datasets are often highly imbalanced, with benign network traffic vastly outnumbering malicious activities [1,2]. Such an imbalance

https://doi.org/10.3390/app16010119

poses significant challenges for machine learning classifiers, as minority attack classes are frequently misclassified and their decision boundaries remain poorly understood [3]. To address accuracy challenges, recent studies have proposed advanced deep learning architectures that incorporate feature fusion mechanisms that better capture relationships among specialized features [4] or hybrid intrusion detection systems combining multiple algorithms (i.e., XGBoost, Long Short-Term Memory (LSTM), Mini-VGGNet) [5] to estimate feature importance and improve detection accuracy and interpretability. Adversarial training, incorporating GANs and Siamese Neural Networks, has also been shown to improve classification performance, especially for minority classes with less clearly defined decision boundaries [6]. However, the complexity and low explainability of the results obtained from advanced AI models further exacerbate this problem. Their decision-making processes are often non-transparent, leaving security analysts and experts unable to explain why a particular network flow was flagged as suspicious. In a high-risk context, this opacity undermines trust, accountability, and operational reliability. Improving the explainability of AI-driven cybersecurity systems is therefore not only a matter of explainability but also a prerequisite for reliable and auditable decision-making.

Local Interpretable Model-agnostic Explanations (LIME) [7] has emerged as a widely used approach for post-hoc explanation of individual predictions. LIME approximates a black-box model locally by generating a synthetic neighborhood of perturbations around the instance of interest, obtaining the black-box predictions for these perturbed samples, and then fitting an interpretable surrogate (usually a weighted linear regression) to mimic the model's local behavior. The surrogate model's coefficients serve as an explanation, indicating the influence of each feature on the prediction for that instance. Despite its popularity, the original LIME framework exhibits notable limitations in terms of fidelity and stability [8–10]. In the context of XAI, fidelity and stability are two key metrics for assessing the quality of local surrogate explanations: Fidelity often refers to how well the surrogate's predictions match the black-box outputs, usually quantified by metrics such as R^2 for regression tasks and F1-score for classification; stability, in turn, measures the consistency of explanations in multiple runs or small input perturbations, typically evaluated through overlap of feature classifications or *Jaccard* similarity [3,9,11]. However, the interpretation and computation of these metrics vary across studies [8,12,13].

The main reason is the LIME perturbation generation strategy for digital features: it samples each feature independently from a normal distribution $N(0,1)$ (which is then scaled and shifted) [7]. This simplistic scheme assumes features are uncorrelated and normally distributed, which is rarely true in real-world data. The problem becomes especially pronounced in network intrusion detection, where datasets such as CIC-IDS-2017 or CIC-IDS-2018 [14] are highly imbalanced and exhibit strong inter-feature dependencies as well as non-Gaussian distributions. Consequently, many of the synthetic perturbations generated by LIME fall outside the genuine data manifold, causing the surrogate model to be trained on unrealistic samples that poorly approximate the actual local decision boundary of the black-box model. This can drastically reduce the surrogate model's fidelity, stability, and other quality metrics, making the explanations highly sensitive to random initialization and sampling variations [15]. Prior studies have observed that LIME explanations can vary significantly with different random draws of perturbations [11,16].

To address these shortcomings, an improved perturbation generation methodology for LIME is proposed that is both data-driven and dependency-aware. In particular, the proposed approach uses copulas to model and sample from the joint distribution of features, thereby preserving inter-feature dependency and producing more realistic synthetic data points [17]. Copulas are functions that couple multivariate distribution functions to their one-dimensional marginals, enabling flexible modeling of dependencies separate from marginal distributions [18,19]. By fitting copulas to highly dependent feature pairs observed in the data, perturbations are generated that capture non-linear and tail dependencies overlooked by a Gaussian sampling. For features not strongly dependent, we perform univariate distribution fitting (considering Gaussian, lognormal, Gamma, Weibull, etc.) and select the best fit via a Kolmogorov–Smirnov test, instead of defaulting to Gaussian. This ensures each feature's marginal perturbation distribution aligns with the empirical data (e.g., heavy-tailed or skewed features are sampled appropriately), further increasing the fidelity of the surrogate model. Additionally, this paper introduces strategies to improve perturbation localization around the instance being explained, addressing the limitation of the original LIME approach, where numeric perturbations are merely centered on the data mean or instance value and weighted by an exponential kernel to approximate locality [7]. The proposed copula-based LIME framework is evaluated on a benchmark cybersecurity dataset (intrusion detection) where strong correlations exist between certain network features. Explanation fidelity and stability were quantified through comparative analysis of feature importance consistency across multiple runs. In particular, the conventional Gaussian perturbation approach is compared with the proposed bivariate copula model (for a pair of highly correlated features), each evaluated under various localization settings, including global, radius-constrained, and single-feature-fixed perturbations.

A key contribution of this work is the development of CoLIME, a dependency-aware extension of LIME that leverages copula-based perturbations to improve explanation quality for intrusion detection models. By preserving realistic joint feature behaviour during the sampling process, CoLIME substantially enhances both the fidelity and stability of the resulting local explanations. Empirically, for the most dependent feature pairs, CoLIME increases mean surrogate fidelity by 21.84–50.31% on the merged CIC-IDS2017/2018 dataset and by 29.28–60.24% on the UNSW-NB15 dataset. Stability is similarly improved, with mean Jaccard similarity gains of 3.78–5.45% and 1.95–2.12%, respectively. These improvements demonstrate that dependency-preserving perturbations provide a significantly more reliable foundation for explaining complex network intrusion detection models.

The cybersecurity domain was chosen as a testbed because network intrusion detection datasets typically contain a large number of interrelated statistical flow features that exhibit strong linear and non-linear dependencies. Such correlations naturally arise from the sequential and bidirectional nature of network communication and become even more pronounced during attack events, when abnormal traffic patterns cause multiple features to vary simultaneously. This makes intrusion detection data particularly suitable for evaluating dependency-aware explanation methods such as the proposed copula-based LIME. Moreover, explainability in cybersecurity is especially critical because understanding why an alert is raised helps analysts verify the cause of an attack, reduce false positives, and strengthen operational trust in AI-driven monitoring systems. Therefore, evaluating CoLIME on a representative cybersecurity dataset provides both a technically challenging and a practically relevant scenario to assess the benefits of dependency-preserving perturbations.

The rest of this paper is organized as follows. Section 2 reviews related work on enhancing LIME and similar local explanation methods, highlighting how the proposed approach differs. Section 3 describes the dataset and pre-processing steps used. In Section 4, the materials and methods are presented, including descriptions of how dependent feature subsets were identified, copula models constructed, perturbations generated and localized, and fidelity and stability quantified. Section 5 presents the results and comparative analysis, including tables and visualizations showing the trade-offs between strategies. Finally, Section 7 provides concluding remarks, summarizing the findings, and suggesting potential future directions to improve local explainability.

2. Related Works

The research community has extensively examined the limitations of LIME and proposed numerous modifications to enhance its explanatory reliability. Although recent works have introduced additional evaluation dimensions such as robustness, prescriptivity, or local concordance, the most critical indicators of performance remain fidelity and stability. The main approaches to address these challenges are summarized in Table 1, highlighting their methodological focus and their contribution to improving the reliability of local surrogate explanations.

One direction of research focuses on improvements to LIME's perturbation sampling mechanism. The original LIME framework introduced by Ribeiro et al. [7] employs Gaussian perturbations to generate synthetic samples, but this assumption often results in instability and reduced local fidelity of the surrogate model [15]. Deterministic LIME (DLIME) [16] replaces random sampling with a deterministic procedure. It clusters the training data using agglomerative hierarchical clustering and then selects perturbed samples from the cluster nearest to the instance via *k*-nearest neighbors. By drawing samples only from the relevant neighborhood in the original data distribution, DLIME achieves far more stable explanations. They report substantially higher explanation consistency (intensional stability of selected features) and improved faithfulness of the surrogate, since perturbations are actual data points rather than synthetic noise. Similarly, ALIME (Autoencoder-LIME) [20] utilizes a denoising autoencoder to generate perturbations that preserve local structure. In ALIME, an autoencoder is trained on the data and used to constrain perturbations: random noise is passed through the encoder–decoder, yielding perturbations that lie on the data manifold. This approach improved LIME's locality and interpretability by ensuring synthetic samples are realistic. It effectively acts as a learned weighting function for LIME, focusing the surrogate model on perturbations that autoencoders consider plausible reconstructions of the instance.

Another line of research focuses on modifying how perturbations are selected or weighted. The S-LIME (Stabilized LIME) approach [11] addresses instability by dynamically determining the number of perturbation samples required. Rather than using a fixed 5000 samples, S-LIME employs sequential sampling and a statistical hypothesis test (based on the central limit theorem) to continue drawing perturbations until the explanation (feature importance ranking) converges with high confidence. This ensures stability of the explanation at the cost of more computation. S-LIME does not explicitly change the distribution of samples, but it guarantees that enough samples are drawn to average out the randomness.

Alternatively, US-LIME (Uncertainty Sampling LIME) [21] seeks to improve surrogate fidelity by prioritizing perturbations in regions of higher informational value. This method proposes uncertainty sampling, which involves preferentially generating perturbed samples near the decision boundary of the black-box model (where the model is most uncertain) rather than uniformly in the local neighborhood. This focused sampling yields perturbations that are more relevant for approximating the model locally, thus improving the surrogate's fidelity. Their experiments on tabular data showed higher R^2 for the surrogate and modest gains in stability as well, since redundant far-off points are reduced.

Beyond sampling strategies, modifications to the surrogate model or loss function have been explored. For example, GLIME (General, Stable and Local LIME) [15], revises LIME's objective function. They derive an equivalent formulation of LIME's weighted least squares problem that accelerates convergence and reduces variance. GLIME also unifies several earlier ideas. It employs an unbiased, purely local sampling distribution (centered precisely on the instance with a proper kernel) instead of LIME's original biased sampling around a reference point. This change led to improved local fidelity and made explanations independent of the centering of data points. Moreover, GLIME allows the user to plug in different sampling distributions (e.g., Gaussian kernel, uniform in a sphere) based on needs. It mitigates the effect of extremely small sample weights, which was identified as a cause of instability (as tiny weights cause the ridge surrogate to rely on the regularization term rather than fitting data, resulting in random coefficients [15]). By ensuring a sufficient number of high-weight samples (through the local unbiased sampling), GLIME achieved both faster surrogate fitting and more stable explanations compared to LIME. Instead of linear models, some works investigated alternative surrogates. In particular, decision trees have been used as surrogate models to yield rule-based explanations that might be more intuitive in some cases. For instance, the Tree-LIME variation (using a decision tree in place of Ridge) can enhance interpretability but might sacrifice some fidelity if the decision boundary is not well approximated linearly [16]. Comparatively, the approach proposed in this research employs Ridge regression as the surrogate to isolate the impact of the perturbation strategy. Recently, generative models beyond autoencoders have also been proposed. Conditional Variational Autoencoder, namely CVAE, was introduced [22] to generate perturbations conditioned on the instance's class label, in an effort to improve local fidelity and maintain interpretability. In a suggested approach, CVAE-LIME modifies only the sampling part of LIME: instead of random noise, they sample from a CVAE that is trained to model the distribution of feature vectors given a class. By doing so, the perturbations are both plausible and more focused on regions that matter for the prediction. A notable increase in R^2 of explanations and evaluated stability under noise were observed, finding that while CVAE-LIME was not as robust as methods explicitly targeting stability (like S-LIME), it still performed reasonably well. This reinforces the trend that data-driven sampling can boost fidelity substantially.

A summary of the key LIME variants is provided in Table 1. Compared to these works, the proposed approach is novel in employing copula functions to capture multi-feature dependencies. Copula-LIME (CoLIME) contributes to the ongoing efforts to improve LIME's perturbation sampling mechanism by introducing a dependency-aware, copula-based approach that preserves inter-feature relationships and enhances local fidelity. The method combines copula-based sampling with localization constraints, thereby improving the accuracy and consistency of explanations while ensuring that the generated perturbations remain faithful to the underlying data distribution.

Table 1. Comparison of LIME enhancement methods and their impact on explanation fidelity and stability.

Year, [Reference]	Method	Core Approach	Focus of Improvement
2016 [7]	Original LIME	Random Gaussian perturbations, weighted linear surrogate.	The baseline.
2021 [16]	DLIME	Deterministic sampling from nearest real-data cluster (AHC + KNN).	Stability (reproducible explanations.)
2019 [20]	ALIME	Denoising autoencoder for manifold-constrained perturbations.	Fidelity (local weighting) and stability.
2021 [11]	S-LIME	Adaptive sampling until feature importance convergence (CLT-based)	Stability (convergence of feature importance).
2024 [21]	US-LIME	Uncertainty sampling near model decision boundary.	Fidelity (focus on informative perturbs).
2025 [22]	CVAE-LIME	Conditional VAE-based perturbation generation.	Fidelity and interpretability
2023 [15]	GLIME	Unbiased local sampling with reformulated loss.	Fidelity and stability.
Our proposed method	**CoLIME**	Fit bivariate copulas for correlated features; KS-fit marginals for others; add locality constraints (radius/fix).	Fidelity (realistic joint samples) and stability (localized sampling).

3. Dataset

The proposed approach was first evaluated on the CIC-IDS2017 and CSE-CIC-IDS2018 intrusion detection datasets, which were combined and preprocessed into a single dataset for experimentation. CIC-IDS2017 [23] contains around 2.8 million network flow records (with 79 features) collected over a week, encompassing 15 classes of traffic (normal and various attacks like DoS, brute force, etc.). CSE-CIC-IDS2018 expands on this with over 16 million flows and features for defining additional attack types (including web attacks such as SQL injection, XSS). Building on prior studies, related categories from both datasets were unified, resulting in 28 consolidated classes. The merged dataset, namely CIC-IDS2017/2018, comprises approximately 18 million instances and 80 features (including the class label), which were further organized into five functional classes, summarized in Table 2. The merged dataset provides a comprehensive representation of modern network intrusion patterns.

Table 2. Summary of feature classes in the CIC-IDS2017/2018 dataset (five-class grouping).

No.	Feature Group	Count	Description/Example Features
1	Basic flow identifiers and durations	9	Attributes describing session-level properties such as *Destination Port, Flow Duration, Flow Packets/s, Flow Bytes/s, Down/Up Ratio, Subflow Fwd Bytes, etc*. These features define the basic structural characteristics of each network flow.
2	Packet length and statistical descriptors	20	Metrics summarizing packet-size distribution and variability: *Fwd/Bwd Packet Length (Mean, Std, Max, Min), Total Length of Fwd/Bwd Packets, Average Packet Size, etc.*
3	Inter-arrival time (IAT) features	14	Features reflecting temporal spacing between packets, including *Flow IAT Mean, Std, Min, Max, Fwd/Bwd IAT Mean, Std, Active/Idle Mean, Min, Max, Std, etc.* These characterize the temporal rhythm of communication.
4	Protocol and flag-based indicators	17	Binary or count-based attributes representing control and protocol-level signaling patterns such as *SYN, ACK, FIN, URG, ECE, PSH, RST Flag Counts, Fwd/Bwd Header Length, etc.*
5	Bulk, subflow, and activity metrics	20	Aggregated metrics describing session intensity and transmission behavior, including *Subflow Fwd/Bwd Packets and Bytes, Avg Fwd/Bwd Bulk Rate, Avg Bytes/Bulk, Avg Packets/Bulk, etc.*

Additionally, for comparative purposes, the proposed approach was also evaluated using the CIC UNSW-NB15 [24] intrusion detection dataset. This dataset was collected over two days and contains 3,540,241 entries, encompassing 10 classes of benign and malicious network traffic and 84 features (including the label), which were further grouped into five summarized feature classes, as shown in Table 3. Compared with CIC-IDS2017/CSE-CIC-IDS2018, the network traffic classes in CIC UNSW-NB15 are more generic in nature; however, the issue of class imbalance remains present in both datasets.

Table 3. Summary of feature classes in the CIC-UNSW-NB15 dataset (five-class grouping).

No.	Feature Group	Count	Description/Example Features
1	Basic flow identifiers and durations	11	Attributes describing session-level properties such as *Source/Destination IP and Port, Protocol, Flow Duration, Total Fwd/Bwd Packets, Total Fwd/Bwd Bytes*, and related fields. These features define the basic structural characteristics of each network flow.
2	Packet length and statistical descriptors	20	Metrics summarizing packet-size distribution and variability, including *Fwd/Bwd Packet Length (Mean, Std, Max, Min), Total Length of Fwd/Bwd Packets, Packet Length Variance, Average Packet Size*, and similar statistics capturing payload volume per flow.
3	Inter-arrival time (IAT) features	14	Features reflecting temporal spacing between packets, such as *Flow IAT Mean, Std, Min, Max, Fwd/Bwd IAT Mean and Std, Active/Idle Mean, Min, Max, Std*, and related timing descriptors. These characterize the temporal rhythm of communication.
4	Protocol, state, and flag-based indicators	18	Binary or count-based attributes representing control and protocol-level signaling, including TCP flag statistics (*SYN, ACK, FIN, URG, ECE, PSH, RST Flag Counts*), header-length measures (*Fwd/Bwd Header Length*), and UNSW-specific connection descriptors such as *state* or related protocol-status fields.
5	Bulk, subflow, and activity metrics	20	Aggregated metrics describing session intensity and transmission behavior, such as *Subflow Fwd/Bwd Packets and Bytes, Avg Fwd/Bwd Bulk Rate, Avg Bytes/Bulk, Avg Packets/Bulk, Fwd/Bwd Packets/s, Fwd/Bwd Bytes/s*, and other indicators of burstiness and flow activity.

4. Methods

In this section, RF (Random Forest) and XGBoost (eXtreme Gradient Boosting) models for intrusion detection are outlined. LIME is used as the baseline method for generating local explanations. Then, the proposed CoLIME methodology, which is an extension of LIME that improves explanation quality by combining copula-based sampling with multiple localization strategies, is described. In particular, copula models were constructed for bivariate cases to capture complex dependencies among features. Finally, the metrics used to evaluate the fidelity and stability of the generated explanations are defined.

4.1. Intrusion Detection Model

In this study, two widely adopted ensemble classifiers were used, namely Random Forest (RF) and XGBoost, as representative high-performance baselines for network intrusion detection. Their role here is to provide the black-box predictions that CoLIME aims to explain.

Random Forest (RF) is an ensemble learning algorithm that builds multiple decision trees and aggregates their predictions to achieve high accuracy and generalization performance [25]. Owing to its ensemble structure, RF effectively reduces variance and overfitting, making it robust across diverse types of data distributions [26]. The method has demonstrated excellent performance when applied to large-scale and high-dimensional datasets, as it can efficiently handle complex feature interactions and noisy data [27,28]. Furthermore, RF is known to perform well on imbalanced datasets due to its bagging-based sampling and aggregation strategies [29]. These properties make RF a suitable choice for this study, which involves a large, imbalanced dataset requiring reliable and scalable classification performance.

Extreme Gradient Boosting (XGBoost) is a high-performance ensemble algorithm that builds successive decision trees, each correcting the errors of its predecessors, and thereby delivers strong predictive accuracy on structured, tabular data [30,31]. A key strength of XGBoost lies in its computational efficiency: it leverages optimized parallelization, cache-aware data structures and out-of-core computations to scale to large-scale datasets with hundreds of millions of records [32]. It also incorporates explicit regularization terms (both L1 and L2) and a shrinkage framework that help to control overfitting and maintain generalization, particularly in scenarios with noisy, large or heterogeneous feature spaces [31,33]. In addition, XGBoost offers flexible loss-functions and supports learning on weighted samples, which is beneficial when dealing with imbalanced classes or cost-sensitive classification tasks. Owing to these attributes, XGBoost was selected in our study to provide a computationally efficient, scalable and robust benchmark for classification on a large, imbalanced dataset.

The optimal hyperparameter search [34] for the Random Forest and XGBoost models was performed using grid search, a systematic method to identify optimal hyperparameters of a learning algorithm by exhaustively evaluating a predefined set of candidate values. Firstly, a search space was defined that was composed of discrete hyperparameter combinations selected on theoretical and empirical grounds. Each combination was fitted to the training data. The performance of every configuration was then quantified using accuracy, F1-score metrics, their weighted and macro averages. After all candidate configurations have been assessed, the hyperparameter set that achieves the highest validation performance is selected as optimal. Grid Search as a hyper-parameter search strategy was chosen because it offers the highest degree of granularity and reliability for thoroughly exploring the defined hyperparameter space. Unlike probabilistic methods, the deterministic Grid Search approach guarantees checking every possible combination within the grid, which was critical for ensuring the globally optimal parameter set was found within the specified search boundaries for RF and XGBoost. While the datasets are large, the associated computational time was considered to be an acceptable and necessary trade-off to maximize the confidence and robustness of the final model configuration used in the study. By applying a grid search strategy, the optimal hyperparameters for each machine learning model are reported in Table 4. It is important to note that these hyper-parameters were used for model training with both the merged CIC-IDS-2017/2018 dataset and UNSW-NB15 dataset.

Table 4. Optimal random forest and XGBoost hyper-parameters.

Random Forest		XGBoost	
Hyper-Parameter	**Value**	**Hyper-Parameter**	**Value**
n_estimators	100	n_estimators	200
max_depth	30	max_depth	6
min_samples_split	2	learning_rate	0.1
min_samples_leaf	1	subsample	0.8
max_features	sqrt	colsample_bytree	0.8
random_state	42	min_child_weight	1

For the training of both CIC-IDS-2017/2018 and UNSW-NB15 datasets, validation and test sets were also prepared with a ratio of 70-20-10, respectively. Meanwhile, For LIME and CoLIME performance evaluation, only a subset of test dataset was used. To better understand the LIME and CoLIME capabilities and limitations, while working with a highly imbalanced dataset, only data points that belong to “Benign” (83%), “DDoS attacks-LOIC-HTTP” (3%) and “Brute Force -Web” (0.003%) classes were selected as points from the CIC-IDS2017/2018 dataset for which the explanations were generated. This allowed us to perform fine-grained analysis of classifiers and explanation models’ performance with predominant, moderate and minority data classes. By contrast, for the UNSW-NB15 dataset, data points from all 10 classes were used, taking advantage of its smaller size and the resulting lower computational requirements.

4.2. coLIME

Local Interpretable Model-agnostic Explanations (LIME) is a model-agnostic method used to provide explanations of a black-box model by approximating it locally with some interpretable model [7]. Suppose $x \in \mathbb{R}^d$ is the instance to explain. Formally, LIME minimizes the objective function given as:

$$\arg\min_{g \in G} \mathcal{L}(f, g, \pi_x) + \Omega(g)$$

where $\mathcal{L}(f, g, \pi_x)$ determines the local fidelity between the explanation model g and the black-box model f, $\pi_x(z)$ is a proximity measure (kernel function) that defines how close z is to x, and $\Omega(g)$ is a complexity penalty ensuring g remains simple and interpretable. Typically, g is chosen from the family G of interpretable models such as linear regression of decision tree. In this context, z is a perturbed instance, i.e., a synthetic data point generated by slightly modifying x. As such, we get $Z = \{z_1, z_2, \ldots, z_n\}$, which is the set of all perturbed samples generated around x. Then, some interpretable model g is fitted on these $(z_i, f(z_i))$ pairs, with higher weights for z_i close to x.

In case of default LIME [7], for a continuous feature j in the tabular date, the synthetic local observations z_i of the instance x are generated using a normal distribution

$$z_{ij} \sim \mathcal{N}(x_j, \sigma_j^2)$$

where σ_j^2 is a scaled value of feature's j global standard deviation. If feature j is categorical, the empirical categorical distribution of that feature is used to sample synthetic observations, however usually with higher probability of keeping the same value as in x_j.

Notably, when generating the synthetic samples z_i for a given instance x, each feature j is usually sampled independently from the others, i.e., for a feature vector $x = [x_1, x_2, \ldots, x_d]$, the points z_{ij} are perturbed not preserving any dependence between features. However, in real-world cases, it might not be true. Consequently, perturbing them independently, the local surrogate g might be distorted after those unrealistic combinations of z_i are fed into the black-box model f to get unreliable $f(z_i)$.

In this paper, instead of sampling each feature independently, we employ Copula functions to model the dependencies between variables and to generate more realistic perturbations. Copulas enable separating marginal distributions from their dependency structure, allowing flexible modeling of multivariate relationships. In this study, we consider four common bivariate copula families: Gaussian, Clayton, Gumbel, and Frank. The Gaussian copula captures symmetric linear dependence similar to correlation in normal distributions. The Clayton copula emphasizes lower-tail dependence, making it suitable for modeling variables that jointly take small values. The Gumbel copula captures upper-tail dependence, describing situations where extreme high values occur together. The Frank copula, on the other hand, models symmetric dependence without particular emphasis on tails, providing a versatile alternative when tail behavior is not dominant.

The workflow of the proposed CoLIME approach is demonstrated in Figure 1. It begins with data preprocessing, where numerical and categorical features are cleaned, encoded, and prepared for subsequent modeling. The dependency detection is then performed by identifying numerical features that exhibit strong monotonic relationships, based on rank correlations, and selecting non-overlapping pairs suitable for joint modeling. The next stage includes a copula construction: marginal distributions are fitted and several candidate bivariate copula families are evaluated to select the best-fitting model for each dependent feature pair. These copulas are then used in the fourth stage for perturbation generation, where synthetic neighborhoods are created through dependency-preserving sampling, followed by mapping samples back to the original feature space and enforcing locality constraints. In the next stage, surrogate modeling, the black-box classifier (here, XGBoost or RF) is queried on the synthetic neighborhood, proximity weights are computed, and a sparse linear surrogate is fitted to approximate the model's local behavior. Finally, the methodology concludes with evaluation, where fidelity and stability are assessed to quantify how accurately and consistently the surrogate captures the underlying decision boundary. Together, these stages form a coherent pipeline that integrates dependency detection, copula-based generative modeling, and localized surrogate fitting to enhance the reliability of LIME explanations.

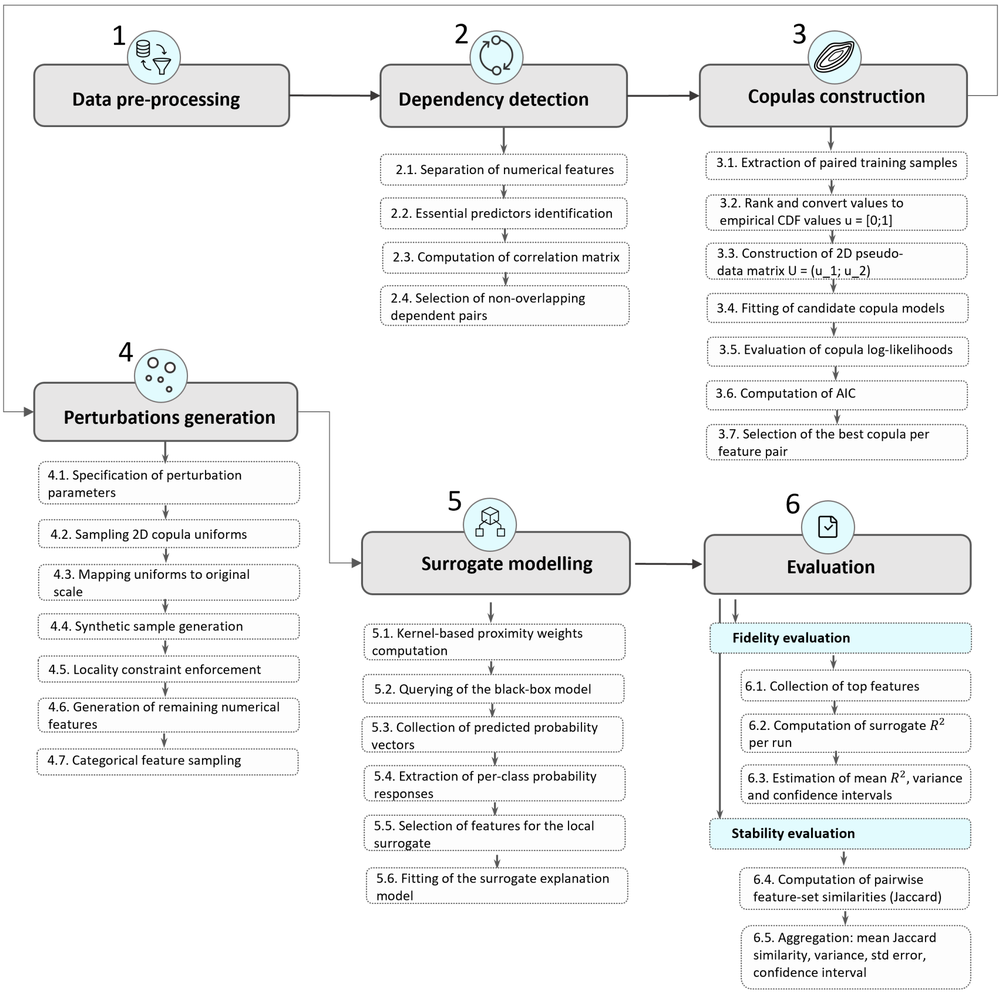

1
Data pre-processing
2
Dependency detection
2.1. Separation of numerical features
2.2. Essential predictors identification
2.3. Computation of correlation matrix
2.4. Selection of non-overlapping dependent pairs
3
Copulas construction
3.1. Extraction of paired training samples
3.2. Rank and convert values to empirical CDF values u = [0;1]
3.3. Construction of 2D pseudo-data matrix U = (u_1; u_2)
3.4. Fitting of candidate copula models
3.5. Evaluation of copula log-likelihoods
3.6. Computation of AIC
3.7. Selection of the best copula per feature pair
4
Perturbations generation
4.1. Specification of perturbation parameters
4.2. Sampling 2D copula uniforms
4.3. Mapping uniforms to original scale
4.4. Synthetic sample generation
4.5. Locality constraint enforcement
4.6. Generation of remaining numerical features
4.7. Categorical feature sampling
5
Surrogate modelling
5.1. Kernel-based proximity weights computation
5.2. Querying of the black-box model
5.3. Collection of predicted probability vectors
5.4. Extraction of per-class probability responses
5.5. Selection of features for the local surrogate
5.6. Fitting of the surrogate explanation model
6
Evaluation
Fidelity evaluation
6.1. Collection of top features
6.2. Computation of surrogate R^2 per run
6.3. Estimation of mean R^2, variance and confidence intervals
Stability evaluation
6.4. Computation of pairwise feature-set similarities (Jaccard)
6.5. Aggregation: mean Jaccard similarity, variance, std error, confidence interval

Figure 1. Flowchart of CoLIME methodology pipeline.

More specifically, when two features X_1 and X_2, for some instance $x = (x_1, x_2)$, the synthetic samples $z_i = (z_{i1}, z_{i2})$ that are close to x should be created following some dependence structure defined via a bivariate copula. More specifically, the joint distribution using a copula $C_\theta(u_1, u_2)$ is created, where $u_1 = F_1(x_1)$, $u_2 = F_2(x_2)$, with F_1 and F_2 defining marginal CDFs. Then, the dependence structure of (x_1, x_2) is defined as

$$H(x_1, x_2) = C_\theta(F_1(x_1), F_2(x_2)).$$

In order to generate samples of X_2 conditional on $X_1 = x_1$, the Algorithm 1 is proposed.

Algorithm 1 Conditional Sampling from a Bivariate Copula in a Neighborhood of $u^\star$

```
Number of draws $N$; copula $C(u, v; \theta)$; center point $u^\star \in (0,1)$; range width $\delta > 0$; parameter $\theta$.
Simulated realizations $\{v_i\}_{i=1}^{n}$ conditional on $u \in [u^\star - \delta,\ u^\star + \delta]$.
for $i = 1$ to $n$ do
    Draw $u_i \sim \text{Uniform}(u^\star - \delta,\ u^\star + \delta)$
    Draw $w_i \sim \text{Uniform}(0,1)$
    Compute conditional CDF:   $H(v; u_i) = \dfrac{\partial C(u_i, v; \theta)}{\partial u_i}$
    Obtain $v_i = H^{-1}(w_i; u_i)$
end for
return $\{v_i\}_{i=1}^{n}$
```

In algorithm, $u_i \in [u^\star - \delta,\ u^\star + \delta]$ and $v_i \sim V|U = u_i$ are dependence-preserving perturbations in a copula space. In the next step, they are used to fit a local surrogate (see Algorithm 2).

Algorithm 2 CoLIME with Bivariate Copula Samples

```
Require: Copula samples $\{(u_i, v_i)\}_{i=1}^{N}$; black-box $f$; interpretable map $\phi(\cdot)$; distance $D(\cdot,\cdot)$; kernel width $\sigma$; default LIME perturbation DefaultLIMEPerturb$(\cdot)$ for $j \notin \{1,2\}$.
Ensure: Local surrogate $g$ explaining $f$ near $\mathbf{x}$.
    Step: Map copula samples back to feature space
for $i = 1$ to $N$ do
    $x_1^{(i)} \leftarrow F_1^{-1}(u_i)$
    $x_2^{(i)} \leftarrow F_2^{-1}(v_i)$
end for
    Step: Build LIME neighborhood (neighborhood conditioning)
Initialize $\mathcal{Z} \leftarrow \varnothing$
for $i = 1$ to $N$ do
    $\mathbf{z} \leftarrow \mathbf{x}$
    $z_1 \leftarrow x_1^{(i)}$;   $z_2 \leftarrow x_2^{(i)}$
    for each $j \in \{1, \ldots, d\} \setminus \{1,2\}$ do
        $z_j \leftarrow$ DefaultLIMEPerturb$(x_j)$
    end for
    $y_i \leftarrow f(\mathbf{z})$
    $w_i \leftarrow \exp\Big( - \dfrac{D\big(\phi(\mathbf{z}), \phi(\mathbf{x})\big)^2}{\sigma^2} \Big)$
    $\mathcal{Z} \leftarrow \mathcal{Z} \cup \{(\phi(\mathbf{z}), y_i, w_i)\}$
end for
    Step: Fit the local surrogate
Learn $g$ by solving
        $$\min_{g \in G} \sum_{(\mathbf{u}, y, w) \in \mathcal{Z}} w\, \big(y - g(\mathbf{u})\big)^2 + \Omega(g),$$
    where $G$ is an interpretable model class
return $g$
```

Moving to the higher dimension, a multivariate dependence structure is constructed using pair-copula decomposed models, proposed in [35]. To organize those pairwise dependencies efficiently, the concept of vines was introduced. This is a graphical framework that describes the hierarchical structure of conditional dependencies through a sequence of linked trees. In the works of [36,37], a graphical model denoted as the regular vine (R-vine) was proposed, which suggests the highest flexibility. Later, two special cases of R-vine, such as C-vine and D-vine were developed [38] to enable a specific way of decomposing the density. However, the selection of an appropriate structure of the construction of a vine copula model is a crucial step, as it includes the ordering of variables and the configuration of trees. Following [35], pairwise measures are typically computed to quantify the strength of association between variables of dependence, as for example, the absolute values of Kendall's τ or Spearman's ρ. The first tree of the vine is then obtained by constructing a maximum spanning tree (MST), where the edge weights correspond to these absolute dependence measures. For higher-order trees, the process is repeated using values that are obtained from conditional copula derived from the previously estimated trees. At each subsequent level, the algorithm again selects edges that maximize the overall dependence strength. As the result of this approach, a fully specified vine structure is determined, consisting of a sequence of trees that systematically represent the hierarchical dependence relationships among all variables.

4.3. Measuring Stability and Fidelity

Two quantitative metrics were used to evaluate the quality of LIME explanations: fidelity is measured by R^2 and stability.

More specifically, for each explanation, the fidelity of the surrogate model is quantified by computing the coefficient of determination (R^2) between the surrogate's predictions and the black-box model's predictions on the locally perturbed samples. Let $f(x)$ denote the output of the black-box model for a sample x, and $g(x)$ denote the corresponding prediction of the surrogate explanation model. Given a set of N sampled perturbations in the local neighborhood of the instance being explained, the R^2 score is defined as:

$$R^2 = 1 - \frac{\sum_{i=1}^{N} \pi_x(x_i)(f(x_i) - g(x_i))^2}{\sum_{i=1}^{N} \pi_x(x_i)\left(f(x_i) - \overline{f(X)}\right)^2}, \tag{1}$$

where $\overline{f(X)}$ is the mean of the black-box model's outputs $f(x_i)$ over the sampled perturbations. The numerator corresponds to the residual sum of squares (the discrepancy between the surrogate and the black-box), while the denominator is the total sum of squares (the variance of the black-box model's predictions in the local region). An R^2 value of 1 indicates perfect fidelity (the surrogate matches the black-box predictions exactly), whereas $R^2 = 0$ indicates that the surrogate performs no better than predicting a constant value equal to the local mean. Negative values may also occur, signaling that the surrogate is performing worse than such a constant predictor. As in standard LIME-based evaluations, the R^2 score is computed separately for each explained instance, and the mean R^2 over all considered instances is then reported as the overall fidelity measure.

Meanwhile, stability, in the context of explanation methods, refers to how consistently an explanation is reproduced when small changes are introduced, either through repeated executions of the method or through slight perturbations of the input data or model. A stable explanation method should yield similar explanations across such variations; otherwise, confidence in the interpretability technique is diminished. A common way to evaluate stability is to examine the overlap between the sets of features (or factors) that are deemed important across multiple explanation runs. One widely used measure for this purpose is the Jaccard similarity coefficient, which quantifies the similarity between two sets based on the size of their intersection relative to their union.

Formally, let $E_r(f,x)$ and $E_s(f,x)$ denote the sets of features provided by the explanation method E (e.g., LIME) applied to the same model f and input x during two independent runs r and s. The Jaccard similarity between these two explanation runs is then defined as:

$$J(E_r(f,x), E_s(f,x)) = \frac{|E_r(f,x) \cap E_s(f,x)|}{|E_r(f,x) \cup E_s(f,x)|} \tag{2}$$

where $|E_r(f,x) \cap E_s(f,x)|$ denotes the number of features shared between the two explanation runs, and $|E_r(f,x) \cup E_s(f,x)|$ denotes the total number of distinct features identified across both runs. The Jaccard similarity coefficient J takes values in the range $[0,1]$, where $J = 1$ indicates perfect agreement between the two explanations (i.e., identical sets of important features), and $J = 0$ indicates complete dissimilarity (no overlap in identified features). Higher values of J therefore correspond to more stable and consistent explanations, while a perfectly stable explanation method would yield identical feature sets across all runs.

5. Results

5.1. Evaluation of RF and XGBoost for Intrusion Detection

Examination of the confusion matrices provided in Figure 2 revealed that for “Benign” class, XGBoost correctly predicted 1,575,304 samples with only 105 misclassifications, whereas Random Forest misclassified 106 samples in the same class. For “DDoS” class, XGBoost achieved a perfect classification of 12,800 instances, while Random Forest recorded two misclassifications. Similarly, for class “PortScan”, XGBoost correctly classified 15,886 samples with only 1–2 misassignments, while Random Forest misclassified 19 samples in this class. A notable difference was observed in class “Infilteration”, where XGBoost produced 331 false positives compared to 1568 in Random Forest, indicating markedly better specificity in this minority class. For underrepresented classes such as “Web Attack-Brute Force” and “Web Attack-XSS”, XGBoost misclassified 18 and 12 samples, respectively, while Random Forest exhibited slightly higher misclassification counts (18 for both), especially for “Web Attack-XSS”. The proportion of correctly classified samples in the dominant classes consistently exceeded 99.9% for both models, but XGBoost maintained a modest advantage across low-frequency classes by reducing accumulation of classification errors. These findings indicate that while both ensemble approaches are highly effective, XGBoost offers slightly superior robustness in handling minority classes and yields overall cleaner diagonal dominance in the confusion matrix across classes.

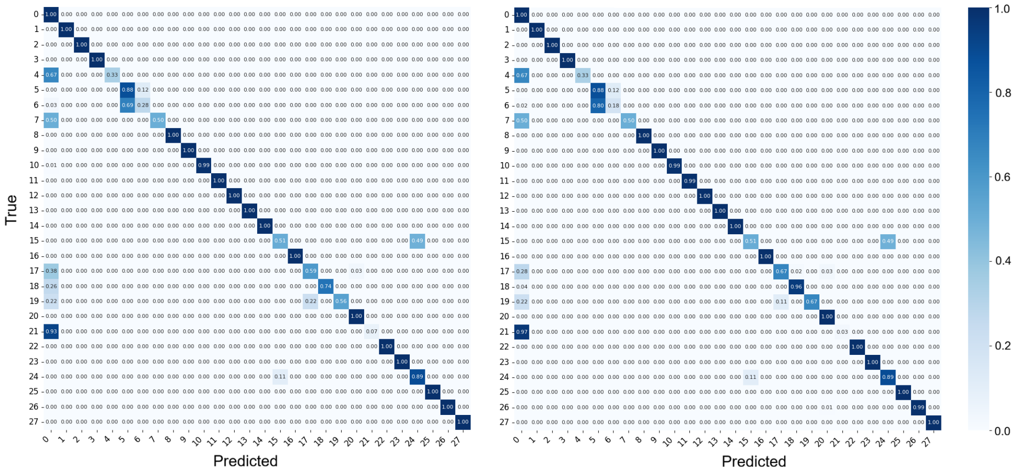


Figure 2. Confusion matrices for RF (**left**) and XGBoost (**right**), trained with CIC-IDS2017/2018 merged dataset.

Despite the near-perfect AUC scores, the classification performance metrics in Table 5 reveal notable complexity in the multi-class task. Both models exhibit excellent overall performance, with weighted accuracy scores above 0.998 and weighted F1 scores above 0.982 (RF: 0.98311; XGB: 0.98264). This indicates exceptional performance on the high-frequency classes.

Table 5. RF and XGBoost classification performance metrics with CIC-IDS2017/2018 merged dataset.

Model	Accuracy (Weighted)	Precision (Weighted)	Recall (Weighted)	F1 (Weighted)	Accuracy (Macro)	Precision (Macro)	Recall (Macro)	F1 (Macro)
RF	0.99875	0.98262	0.98640	0.98311	0.98640	0.92933	0.83321	0.86095
XGBoost	0.99954	0.98345	0.98668	0.98264	0.98668	0.92925	0.84249	0.86308

However, the macro-averaged metrics, which evaluate model stability across all 28 classes, show a slight reduction. The macro F1 scores are 0.86095 for RF and 0.86308 for XGBoost. This drop suggests difficulty maintaining high recall and precision on the rarer attack types.

XGBoost achieves marginally higher scores in both weighted accuracy (0.99954 vs. 0.99875) and macro F1 (0.86308 vs. 0.86095), largely driven by a higher macro recall (0.84249 vs. 0.83321). This suggests XGBoost is slightly more effective at identifying positive instances across all classes equally. The significant difference in the ROC curves for Class 21 (RF: AUC = 0.98; XGB: AUC = 0.89) is partially offset by XGBoost's overall better macro-performance, suggesting its lower AUC on that specific class does not critically impair its averaged results, maintaining strong classification performance across the entire, highly diverse dataset.

As both RF and XGBoost models were also trained with the UNSW-NB15 dataset, examination of the confusion matrices, provided in Figure 3, revealed that for the Dominant Class "Benign", XGBoost correctly predicted 344,117 samples, incurring 953 misclassifications (False Negatives), whereas Random Forest correctly predicted 343,912 samples with 1095 misclassifications in the same class.

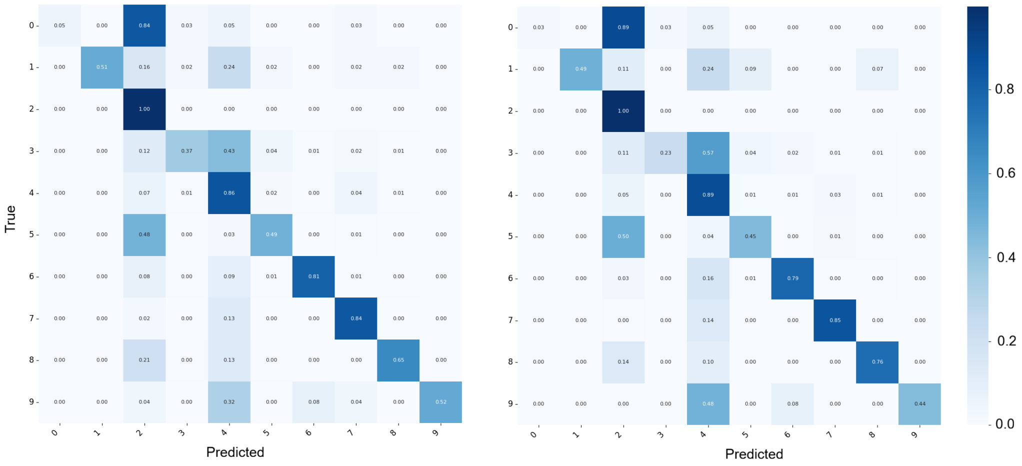


Figure 3. Confusion matrices for RF (**left**) and XGBoost (**right**), trained with UNSW-NB15 dataset.

For the mid-sized minority class "Fuzzers", XGBoost also demonstrated superior performance, recording 281 misclassifications (False Negatives) compared to 310 for Random Forest. Similarly, for class $\mathbf{C}_6$, XGBoost was slightly better with 2471 misassignments versus 2512 for Random Forest. A notable difference was observed in the misclassification of the Dominant Class "Benign" as the mid-sized class "Fuzzers", where XGBoost produced 589 instances of this error (a form of False Positive for "Fuzzers"), compared to 842 in Random Forest, indicating a markedly better specificity for XGBoost in avoiding this confusion. Conversely, for the smallest minority classes, such as "Analysis" and "Backdoor", Random Forest showed a slight edge, correctly classifying 2 and 23 instances, respectively, while XGBoost classified 1 and 22 instances. The overall accumulation of False Positives for the Dominant Class "Benign" was almost identical: 2735 for XGBoost and 2736 for Random Forest. The proportion of correctly classified samples in the dominant class consistently exceeded 99.7% for both models, but XGBoost maintained a modest advantage across most classes by reducing the accumulation of classification errors, particularly in the number of False Negatives for the larger minority classes ("Fuzzers" and "Generic") and in reducing a major off-diagonal error ("Benign" as "Fuzzers"). These findings indicate that while both ensemble approaches are highly effective, XGBoost offers slightly superior robustness and overall cleaner diagonal dominance in the confusion matrix.

On the other hand, quantitative performance metrics detailed in Table 6 show differences in classification of instances at the standard operating point.

Table 6. RF and XGBoost classification performance metrics with UNSW-NB15 dataset.

Model	Accuracy (Weighted)	Precision (Weighted)	Recall (Weighted)	F1 (Weighted)	Accuracy (Macro)	Precision (Macro)	Recall (Macro)	F1 (Macro)
RF	0.99660	0.98833	0.98899	0.98843	0.98899	0.80631	0.60933	0.66864
XGBoost	0.99719	0.98868	0.98942	0.98857	0.98942	0.77223	0.59244	0.64060

Both models achieve highly commendable global performance, with XGBoost showing a slight edge in Weighted Accuracy (0.99719 versus RF's 0.99660) and nearly identical Weighted F1 scores (XGB: 0.98857; RF: 0.98843). These weighted metrics, which are dominated by the larger classes, confirm their mastery over the frequent attack types. Nonetheless, differences emerge in the macro-averaged metrics, which evaluate performance equally across all ten classes. The Macro F1 scores are notably lower (RF: 0.66864; XGB: 0.64060) than the Weighted F1 scores, indicating that while they perform perfectly on the majority classes (as reflected in the ROC data), both models encounter difficulties accurately classifying the rarer attack types when using the default probability threshold. Specifically, RF demonstrates superior generalization capability in this context, achieving better results across all macro-level indicators: macro precision of 0.80631 and macro recall of 0.60933, compared to XGBoost's 0.77223 and 0.59244. This pattern gently suggests that while XGBoost is slightly more accurate overall, RF is more balanced in its predictive ability across the full spectrum of diverse network traffic classes.

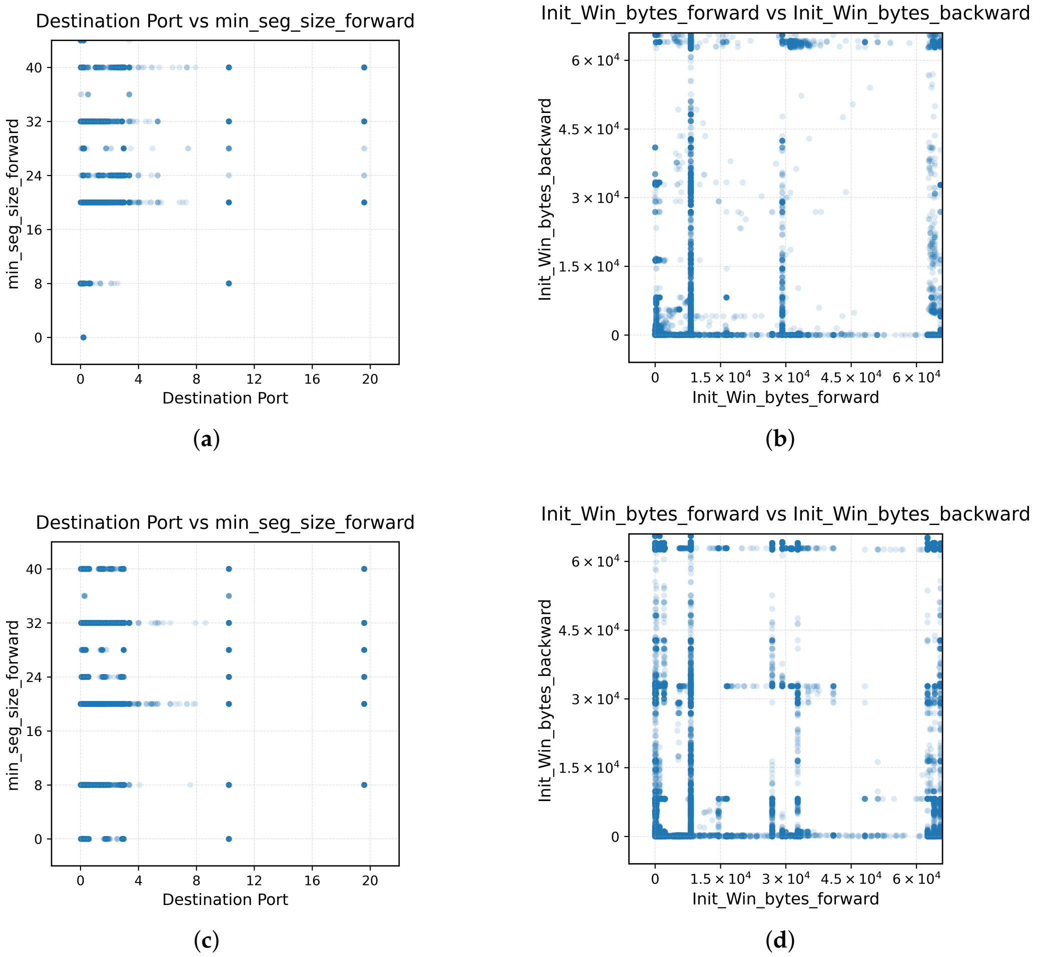
5.2. Fidelity and Stability of Explanations Using coLIME

To illustrate the role of dependency modeling within the CoLIME framework, we first examine how strongly correlated feature pairs behave in the original datasets and how well their joint distributions are reproduced through copula-based sampling. For the demonstration purposes, features with high monotonic dependence (Spearman's $\rho \approx 0.8$) were identified in each dataset, enabling the construction of bivariate copulas tailored to these relationships. Figure 4 presents a comparison between the observed joint distributions and their copula-generated counterparts for two representative feature pairs.

In Figure 4a,c, the dependency analysis of "min_seg_size_forward" and "Destination Port" resulted in the low Maximum Mean Discrepancy (MMD = 0.0018), which indicates a high fidelity in reproducing the global distribution shape. This observation is further supported by a low Energy Distance ($d_{\text{energy}} = 0.152$), which confirms the statistical similarity between the original and generated multivariate distributions. However, the analysis of tail dependence reveals a reduction in extreme value correlation, with the upper tail coefficient decreasing from $\alpha_{upper}^{orig} = 0.64$ to $\alpha_{upper}^{syn} = 0.37$, suggesting that while the central tendency is captured effectively, the generative model slightly underestimates the dependence of rare events in this feature pair. The scatterplot in Figure 4b,d demonstrates the method's capability to model complex dependencies between "Init_Win_bytes_forward" and "Init_Win_bytes_backward". While the energy distance is elevated ($d_{\text{energy}} = 73.52$), this is largely attributable to the high magnitude of the raw byte values rather than struc-

https://doi.org/10.3390/app16010119

tural divergence. Crucially, the MMD remains low (MMD $=$ 0.0053), confirming that the synthetic neighborhood successfully mimics the geometric structure of the original data. Furthermore, the synthetic data exhibits a slightly conservative estimation of tail dependence ($\alpha_{\text{upper}}^{\text{syn}} = 0.136$ vs. $\alpha_{\text{upper}}^{\text{orig}} = 0.052$).



(**a**) (**b**)

(**c**) (**d**)

Figure 4. CIC-IDS2017/2018 dataset: Scatter plots of network flow feature pairs: (**a**) "Destination Port" vs "min_seg_size_forward", (**c**) synthetic copula-generated perturbations for the same pair, (**b**) "Init_Win_bytes_forward" vs "Init_Win_bytes_backward", and (**d**) corresponding copula-generated perturbations.

Comparatively, for demonstration purposes, the most correlated features were selected in the CIC UNSW-NB15 dataset (see Figure 5).

Figure 5a,c demonstrate how the complex joint distribution between "Bwd IAT Mean" and "Packet Length Max" was reproduced by the fitted copula. This alignment is substantiated by a low maximum mean discrepancy of 0.0011 and an energy distance of 18.06. Crucially, the method captures the existence of extremal dependencies in the upper tail ($\alpha_{\text{upper}}^{\text{syn}} = 0.195$), reflecting the strong tail dependence in the original distribution ($\alpha_{\text{upper}}^{\text{orig}} = 0.485$). Comparatively, Figure 5b,d demonstrates that the copula-based perturbations successfully maintained the underlying dependence structure between "Bwd Packets/s" vs. "FWD Init Win Bytes", with an energy distance of 127.33 and a maximum mean discrepancy of 0.0075. Moreover, the method demonstrates a capability in modeling extreme relationships, with a synthetic upper tail dependence coefficient ($\alpha_{\text{upper}}^{\text{syn}} = 0.138$) that reflects the significant tail dependence observed in the original distribution ($\alpha_{\text{upper}}^{\text{orig}} = 0.251$).

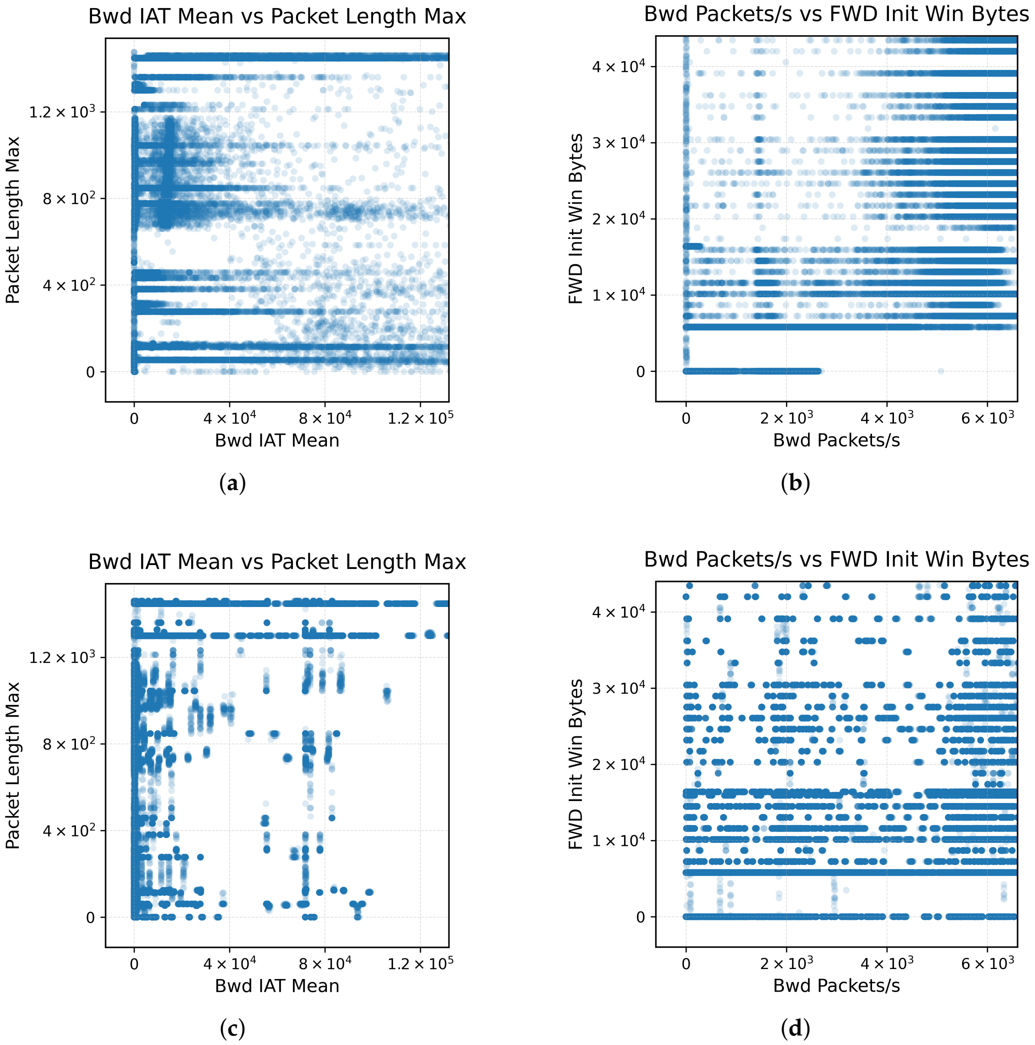


Figure 5. CIC UNSW-NB15 dataset: Scatter plots of network flow feature pairs: (**a**) "Bwd IAT Mean" vs. "Packet Length Max", (**c**) synthetic copula-generated perturbations for the same pair, (**b**) "Bwd Packets/s vs. "FWD Init Win Bytes", and (**d**) corresponding copula-generated perturbations.

After analyzing the capacity of copula-based perturbations to reproduce the joint feature distributions, we now examine how these perturbations affect the quality of the resulting explanations. Specifically, we quantify explanation quality along two complementary axes: stability, which measures the consistency of feature attributions under repeated perturbations, and fidelity, which captures how well the surrogate model locally approximates the black-box classifier.

With the merged CIC-IDS2017/2018 dataset, CoLIME consistently outperforms Original LIME in both stability and fidelity for RF and XGBoost classifiers as it is shown in Table 7. Average stability is higher under CoLIME (0.8069 vs. 0.7652 for RF; 0.7576 vs. 0.7300 for XGBoost), and the gains in fidelity are even more pronounced (0.1676 vs. 0.1115, and 0.0491 vs. 0.0403, respectively), indicating that copula-based perturbations yield explanations that are simultaneously more consistent and more faithful to the underlying model. The confidence interval bounds for CoLIME are systematically shifted upwards for both stability and fidelity, showing that these improvements are not only observed on average but are also supported across repeated runs. In particular, the fidelity intervals for CoLIME do not substantially overlap with those of Original LIME, suggesting a robust enhancement rather than a marginal effect. Variance estimates further support this conclusion: stability variance is comparable between methods, while fidelity variance is notably lower for CoLIME, especially for XGBoost. Overall, the copula-based neighborhoods lead to more reliable and informative local surrogate explanations.

Table 7. Stability and Fidelity averages, their confidence interval averages and variance averages for different perturbation methods and classifiers for merged CIC-IDS2017/2018 dataset.

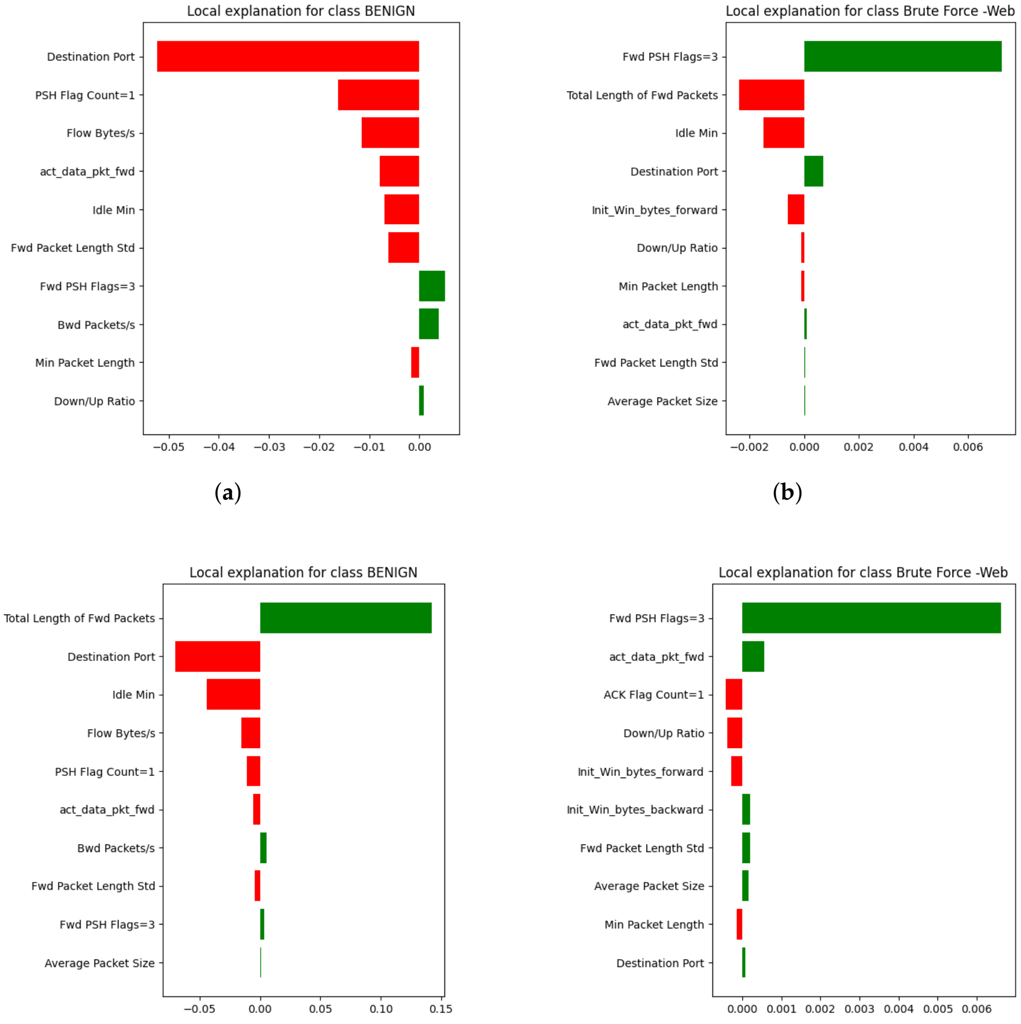
Metric	RF		XGBoost	
	Original LIME	CoLIME	Original LIME	CoLIME
Avg. of Stability	0.7652	**0.8069**	0.7300	**0.7576**
Avg. of Fidelity	0.1115	**0.1676**	0.0403	**0.0491**
Avg. of Stability CI low	0.8127	**0.8532**	0.7759	**0.8062**
Avg. of Fidelity CI low	0.1504	**0.2299**	0.0641	**0.0847**
Avg. of Stability CI high	0.7177	**0.7606**	0.6841	**0.7091**
Avg. of Fidelity CI high	0.0726	**0.1054**	0.0134	**0.0165**
Avg. of Stability variance	0.0150	**0.0143**	**0.0140**	0.0157
Avg. of Fidelity variance	0.0261	**0.0105**	0.0110	**0.0048**

With the UNSW-NB15 dataset, CoLIME again improves both stability and fidelity relative to Original LIME for both classifiers, as shown in Table 8. For RF, average stability increases from 0.5804 to 0.5927 and fidelity from 0.1004 to 0.1298. For XGBoost, the gains are even more pronounced in fidelity (0.1596 vs. 0.0996), with a parallel improvement in stability (0.5924 vs. 0.5811). The confidence intervals are again consistently shifted upwards for CoLIME. Both lower and upper bounds for stability and fidelity are higher under CoLIME for RF and XGBoost, indicating that the enhancements are not limited to a few favourable runs but hold across repeated sampling. In particular, the fidelity CIs show limited overlap between methods, especially for XGBoost, suggesting that the fidelity gains are statistically meaningful. Variance behaviour reflects a modest trade-off: CoLIME slightly reduces stability variance for RF but increases it marginally for XGBoost, and fidelity variance is higher under CoLIME, particularly for XGBoost. Overall, CoLIME yields more accurate explanations, with a small increase in dispersion that accompanies the higher fidelity regime. The practical improvements of CoLIME over the original LIME implementation are visible in Figure 6.

Table 8. Stability and fidelity averages, their confidence interval averages and variance averages for different perturbation methods and classifiers for the CIC-UNSW-NB15 dataset.

Metric	RF		XGBoost	
	Original LIME	CoLIME	Original LIME	CoLIME
Avg. of Stability	0.5804	**0.5927**	0.5811	**0.5924**
Avg. of Fidelity	0.1004	**0.1298**	0.0996	**0.1596**
Avg. of Stability CI low	0.4714	**0.4891**	0.4861	**0.4912**
Avg. of Fidelity CI low	0.0412	**0.0586**	0.0480	**0.0551**
Avg. of Stability CI high	0.6894	**0.6963**	0.6761	**0.6936**
Avg. of Fidelity CI high	0.1597	**0.2010**	0.1512	**0.2641**
Avg. of Stability variance	0.0232	**0.0210**	**0.0176**	0.0201
Avg. of Fidelity variance	**0.0070**	0.0102	**0.0052**	0.0214

Based on the comparative analysis of the feature attribution generated by CoLIME and LIME on the merged CIC-IDS2017 and CSE-CIC-IDS2018 datasets, the CoLIME methodology demonstrates superior explanatory fidelity and robustness within the cyber-security context. The original LIME algorithm shows a tendency toward local instability, which is confirmed by its disproportionately high confidence on the "Total Length of Fwd Packets" (weight: 0.142) for the Benign class (Class 0), as shown in Figure 6c. Meanwhile, CoLIME provides a more balanced and semantically meaningful assignment, as clearly shown in Figure 6a. In network traffic analysis, flow volume features like packet length are often susceptible to high variance. LIME's heavy weighting of this metric suggests potential overfitting to local noise. Conversely, CoLIME generates a more distributed feature ranking for benign traffic, correctly reducing the priority of volume indicators and giving preference to structural protocol indicators, such as "Destination Port" and "PSH Flag Count". Furthermore, regarding the "Brute Force-Web" attack (Class 17), CoLIME assigns a higher relevance to the critical "Fwd PSH Flags" feature (0.0072) compared to LIME (0.0066) (Figures 6b and 6d, respectively). By focusing on the "PSH flag", which indicates that a technical signature is often used in brute force attempts to bypass the buffer, CoLIME accurately captures the dynamics of attack behavior. Consequently, CoLIME offers a more precise, domain-aligned interpretation of the model's decision boundaries, mitigating the risk of spurious correlations inherent in standard local explanations.



(**a**) (**b**)

(**c**) (**d**)

Figure 6. Single instance explanation comparison between CoLIME and LIME. (**a**) CoLIME explanation for class "Benign". (**b**) CoLIME explanation for class "Brute Force-Web". (**c**) LIME explanation for class "Benign". (**d**) CoLIME explanation for class "Brute Force-Web".Green bars indicate positive feature contributions, while red bars indicate negative contributions to the predicted class.

6. Discussion

While previous approaches, including DLIME and ALIME, achieve realistic sampling by constraining perturbations within training clusters or autoencoder manifolds, the joint distribution of selected features is explicitly captured through copula modeling in this study. This has the advantage of preserving complex dependency structures (including tail correlations) in a parametric way. Also, while ALIME and CVAE-LIME require training a generative model (which can be time-consuming and dataset-specific), copulas offer a more straightforward statistical fitting to data slices, which can be efficient for low-dimensional dependencies. In terms of stability, the proposed strategy of perturbation localization (radius or fixed-feature approaches) is conceptually simpler than S-LIME's statistical guarantees but complements the copula sampling by ensuring we do not wander too far from the instance. The proposed method is shown to achieve a favorable balance, attaining substantially higher fidelity than baseline LIME (on par with CVAE-based sampling [22]) while preserving acceptable stability slightly lower than that of DLIME's deterministic approach [16], yet considerably superior to the original LIME's random perturbations. As a limitation, it is worth mentioning that the accurate copula fitting requires sufficient coverage in the joint feature space; in highly sparse regions, dependence estimation may be less stable. CoLIME may also provide limited improvements when feature dependencies are weak or absent, as the benefit of copula-based sampling is directly tied to the strength and stability of underlying relationships. Notably, the magnitude of the observed improvements is strongly influenced by dependency strength: feature pairs with pronounced monotonic relationships benefit most from copula modeling, whereas pairs with weak or unstable correlations show only marginal gains.

The computational complexity of CoLIME is modest for the copula-based component itself, but the overall cost depends on the surrogate model used. While RF, DT, and other surrogates were computationally efficient, our experiments indicated that K-Nearest Neighbors (KNN) yielded the highest fidelity and stability (KNN stability with CoLIME ≥ 0.822, which is $\sim$13% higher compared to the other models; KNN fidelity ≥ 0.584, which is between 2 and 10 times higher than the values obtained with the other four models). However, KNN required, on average, more than 30 times longer training time than the other surrogates (RF, DT, and others), primarily due to its sensitivity to local density, lack of model compression, and the computational cost of repeated distance evaluations on large perturbation sets. This overhead makes KNN impractical for large-scale intrusion detection datasets in the current framework. Nevertheless, these promising results indicate that KNN has substantial potential as a surrogate for dependency-aware perturbation methods, and we outline it as an important direction for future research. Specifically, we plan to explore approximate nearest-neighbor search structures (e.g., KD-trees, ball trees, FAISS indexing), dimensionality-reduction-assisted KNN, and locality-weighted KNN variants, which may significantly reduce computational time while preserving the explanation quality observed in this study.

7. Conclusions

This paper introduced CoLIME, a copula-based extension of LIME for local explanations on highly imbalanced network intrusion data. Key contributions are: a dependency-aware perturbation framework that uses fitted marginal distributions and 2D copulas to generate realistic synthetic neighborhoods; localization strategies that constrain perturbations within a radius or along selected feature dimensions; and an empirical evaluation of fidelity–stability trade-offs on a large-scale intrusion detection benchmark with RF and XGBoost classifiers. Compared to baseline original LIME implementation, CoLIME consistently achieved higher local fidelity of the surrogate model while also maintaining higher stability throughout multiple explanation generations. For the most dependent feature pairs, copula-based perturbations improved the mean fidelity by 21.84–50.31% with the merged CIC-IDS2017/2018 dataset and 29.28–60.24% with the UNSW-NB15 dataset, while increasing mean Jaccard Similarity by 3.78–5.45% and 1.95–2.12% with merged CIC-IDS2017/2018 and UNSW-NB15 datasets, respectively. These results indicate that preserving inter-feature dependence in the perturbation mechanism is crucial when explaining models trained on structured, imbalanced network traffic. Practically, we recommend applying CoLIME in settings where strong feature dependencies can be identified (e.g., via rank correlations) and where fidelity to the black-box behavior is as important as run-to-run stability.

Author Contributions: Conceptualization, K.S. and A.P.-T.; methodology, M.B. and L.M.; software, M.B. and L.M.; validation, A.P.-T. and K.S.; formal analysis, K.S. and A.P.-T.; investigation, K.S. and L.M.; resources, M.B.; data curation, M.B.; writing—original draft preparation, M.B., K.S., L.M. and A.P.-T.; writing-review and editing, M.B., K.S. and A.P.-T.; visualization, L.M. and A.P.-T.; supervision, K.S. and A.P.-T. All authors have read and agreed to the published version of the manuscript.

Funding: This research was co-funded by the European Union under Horizon Europe program grant agreement No. 101059903, the European Union funds for the period 2021–2027, and the state budget of the Republic of Lithuania financial agreement No. 10-042-P-0001.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data used in this study are available from the first author upon reasonable request.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Pawlicki, M.; Choraś, M.; Kozik, R.; Hołubowicz, W. On the Impact of Network Data Balancing in Cybersecurity Applications. In Proceedings of the Computational Science—ICCS 2020, Amsterdam, The Netherlands, 3–5 June 2020; Krzhizhanovskaya, V.V., Závodszky, G., Lees, M.H., Dongarra, J.J., Sloot, P.M.A., Brissos, S., Teixeira, J., Eds.; Springer: Cham, Switzerland, 2020; pp. 196–210.
2. Nebaba, A.N.; Savvas, I.K.; Butakova, M.A.; Chernov, A.V.; Shevchuk, P.S. *Improving Multiclass Classification of Cybersecurity Breaches in Railway Infrastructure Using Imbalanced Learning*; Association for Computing Machinery: New York, NY, USA, 2022. [CrossRef]
3. Bacevicius, M.; Paulauskaite-Taraseviciene, A. Machine Learning Algorithms for Raw and Unbalanced Intrusion Detection Data in a Multi-Class Classification Problem. *Appl. Sci.* **2023**, *13*, 7328. [CrossRef]
4. Ayantayo, A.; Kaur, A.; Kour, A.; Schmoor, X.; Shah, F.; Vickers, I.; Kearney, P.; Abdelsamea, M. Network intrusion detection using feature fusion with deep learning. *J. Big Data* **2023**, *10*, 167. [CrossRef]
5. Pavithra, S.; Venkata Vikas, K. Detecting Unbalanced Network Traffic Intrusions With Deep Learning. *IEEE Access* **2024**, *12*, 74096–74107. [CrossRef]
6. Mirsadeghi, S.M.H.; Bahsi, H.; Vaarandi, R.; Inoubli, W. Learning From Few Cyber-Attacks: Addressing the Class Imbalance Problem in Machine Learning-Based Intrusion Detection in Software-Defined Networking. *IEEE Access* **2023**, *11*, 140428–140442. [CrossRef]
7. Ribeiro, M.T.; Singh, S.; Guestrin, C. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. *arXiv* **2016**, arXiv:1602.04938. [CrossRef]
8. Miró-Nicolau, M.; Jaume-i-Capó, A.; Moyà-Alcover, G. A comprehensive study on fidelity metrics for XAI. *Inf. Process. Manag.* **2025**, *62*, 103900. [CrossRef]
9. Tan, Z.; Tian, Y.; Li, J. GLIME: General, Stable and Local LIME Explanation. In Proceedings of the Advances in Neural Information Processing Systems, New Orleans, LA USA, 10–16 December 2023; Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., Levine, S., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2023; Volume 36, pp. 36250–36277.
10. Knab, P.; Marton, S.; Schlegel, U.; Bartelt, C. Which LIME Should I Trust? Concepts, Challenges, and Solutions. In Proceedings of the Explainable Artificial Intelligence, Istanbul, Turkey, 9–11 July 2025; Guidotti, R., Schmid, U., Longo, L., Eds.; Springer: Cham, Switzerland, 2025; pp. 28–52.
11. Zhou, Z.; Hooker, G.; Wang, F. S-LIME: Stabilized-LIME for Model Explanation. In Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, 14–18 August 2021; ACM: New York, NY, USA, 2021; pp. 2429–2438. [CrossRef]

12. Gawantka, F.; Just, F.; Savelyeva, M.; Wappler, M.; Lässig, J. A Novel Metric for Evaluating the Stability of XAI Explanations. *Adv. Sci. Technol. Eng. Syst. J.* **2024**, *9*, 133–142. [CrossRef]
13. Saarela, M.; Geogieva, L. Robustness, Stability, and Fidelity of Explanations for a Deep Skin Cancer Classification Model. *Appl. Sci.* **2022**, *12*, 9545. [CrossRef]
14. Canadian Institute for Cybersecurity (CIC), University of New Brunswick. Datasets-Research. 2025. Available online: https://www.unb.ca/cic/datasets/index.html (accessed on 9 November 2025).
15. Tan, Z.; Tian, Y.; Li, J. GLIME: General, Stable and Local LIME Explanation. *arXiv* **2023**, arXiv:2311.15722. [CrossRef]
16. Zafar, M.R.; Khan, N. Deterministic Local Interpretable Model-Agnostic Explanations for stable explainability. *Mach. Learn. Knowl. Extr.* **2021**, *3*, 525–541. [CrossRef]
17. Meyer, D.; Nagler, T.; Hogan, R.J. Copula-based synthetic data augmentation for machine-learning emulators. *Geosci. Model Dev.* **2021**, *14*, 5205–5215. [CrossRef]
18. Haugh, M. An Introduction to Copulas. In *IEOR E4602: Quantitative Risk Management*; Lecture Notes; Columbia University: New York, NY, USA, 2016.
19. Ansari, J.; Rockel, M. Dependence properties of bivariate copula families. *Depend. Model.* **2024**, *12*, 20240002. [CrossRef]
20. Shankaranarayana, S.M.; Runje, D. ALIME: Autoencoder Based Approach for Local Interpretability. *arXiv* **2019**, arXiv:1909.02437. [CrossRef]
21. Saadatfar, H.; Kiani-Zadegan, Z.; Ghahremani-Nezhad, B. US-LIME: Increasing fidelity in LIME using uncertainty sampling on tabular data. *Neurocomputing* **2024**, *597*, 127969. [CrossRef]
22. Yasui, D.; Sato, H. Improving local fidelity and interpretability of LIME by replacing only the sampling process with CVAE. *IEEE Access* **2025**, *13*, 53084–53099. [CrossRef]
23. Sharafaldin, I.; Habibi Lashkari, A.; Ghorbani, A.A. Toward generating a new intrusion detection dataset and intrusion traffic characterization. In Proceedings of the 4th International Conference on Information Systems Security and Privacy, Funchal, Spain, 22–24 January 2018; SCITEPRESS-Science and Technology Publications: Setúbal, Portugal, 2018.
24. Mohammadian, H.; Habibi Lashkari, A.; Ghorbani, A.A. Poisoning and Evasion: Deep Learning-Based NIDS under Adversarial Attacks. In Proceedings of the 2024 21st Annual International Conference on Privacy, Security and Trust (PST), Sydney, Australia, 28–30 August 2024; IEEE: Piscataway, NJ, USA, 2024; pp. 1–9. [CrossRef]
25. Breiman, L. Random Forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
26. Liaw, A.; Wiener, M. Classification and regression by randomForest. *R News* **2002**, *2*, 18–22.
27. Chen, C.; Liaw, A.; Breiman, L. *Using Random Forest to Learn Imbalanced Data*; University of California: Berkeley, CA, USA, 2004; Volume 110, p. 24.
28. Belgiu, M.; Drăguţ, L. Random forest in remote sensing: A review of applications and future directions. *ISPRS J. Photogramm. Remote Sens.* **2016**, *114*, 24–31. [CrossRef]
29. Chen, J.; Li, K.; Tang, Z.; Bilal, K.; Yu, S.; Weng, C.; Li, K. A parallel random forest algorithm for big data in a spark cloud computing environment. *IEEE Trans. Parallel Distrib. Syst.* **2016**, *28*, 919–933. [CrossRef]
30. Verma, V. Exploring Key XGBoost Hyperparameters: A Study on Optimal Search Spaces and Practical Recommendations for Regression and Classification. *Int. J. All Res. Educ. Sci. Methods (IJARESM)* **2024**, *12*, 3259–3266. [CrossRef]
31. Imani, M.; Beikmohammadi, A.; Arabnia, H.R. Comprehensive analysis of random forest and XGBoost performance with SMOTE, ADASYN, and GNUS under varying imbalance levels. *Technologies* **2025**, *13*, 88. [CrossRef]
32. Kapoor, S.; Perrone, V. A simple and fast baseline for tuning large XGBoost models. *arXiv* **2021**, arXiv:2111.06924. [CrossRef]
33. Khan, A.A.; Chaudhari, O.; Chandra, R. A review of ensemble learning and data augmentation models for class imbalanced problems: Combination, implementation and evaluation. *Expert Syst. Appl.* **2024**, *244*, 122778. [CrossRef]
34. Bergstra, J.; Bengio, Y. Random search for hyper-parameter optimization. *J. Mach. Learn. Res.* **2012**, *13*, 281–305.
35. Aas, K.; Czado, C.; Frigessi, A.; Bakken, H. Pair-copula constructions of multiple dependence. *Insur. Math. Econ.* **2009**, *44*, 182–198. [CrossRef]
36. Bedford, T.; Cooke, R.M. Probability Density Decomposition for Conditionally Dependent Random Variables Modeled by Vines. *Ann. Math. Artif. Intell.* **2001**, *32*, 245–268. [CrossRef]
37. Bedford, T.; Cooke, R.M. Vines-A new graphical model for dependent random variables. *Ann. Stat.* **2002**, *30*, 1031–1068. [CrossRef]
38. Kurowicka, D.; Cooke, R.M., Distribution-Free Continuous Bayesian Belief Nets. In *Modern Statistical and Mathematical Methods in Reliability*; World Scientific: Singapore, 2005; pp. 309–322. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

https://doi.org/10.3390/app16010119