

UNIVERSITY OF WALES TRINITY SAINT DAVID

School of Computing

*An Evaluation of Explainable Machine Learning
Techniques for Network Intrusion Detection and
Digital Forensic Incident Reconstruction*

MASTER'S DISSERTATION

Submitted by

Keval Mahendra Motivaras

Student ID: 2515743

Supervisor

Dr Dipo Dunsin

25 September 2026

United Kingdom

Abstract

Explainable machine learning methods in network intrusion detection have come a long way, yet the increasing popularity of ensemble and deep learning models has led to the constant problem of interpretation. Security analysts need an understanding of how the alerts work, while the digital forensics analyst has the additional challenge of associating the alerts with preserved artifacts, timelines, provenance and the ability to construct a reproducible hypothesis about the incident. This systematic literature review examines the nexus between the explainable artificial intelligence (XAI), network intrusion detection and digital forensics incident reconstruction. Thirty relevant articles and preprints have been examined, twenty-one of which have been included in the review, while studies from the unrelated medical, materials science and point-cloud domains have been excluded from the analysis. The existing evidence shows that the explainable intrusion detection research is excellent in producing local and global feature attributions, yet it lacks faithfulness, stability, semantic validity and usefulness of the explanations. At the same time, the body of knowledge about digital forensics contains well-developed concepts of evidence preservation, timeline generation, provenance and reproducibility, though those are rarely applied in the context of explainable intrusion-detection research. Thus, the main research gap in explainable intrusion detection is not the lack of another accurate classifier or SHAP visualization, but rather the lack of controlled evaluation of explainability and the ability to produce a timeline, provenance and reproducibility of the reconstructed incident. The review calls for the adoption of an evidence-preserving methodology that will (i) compare the interpretable, ensemble and deep learning models; (ii) evaluate explanations in terms of both technical and human-oriented metrics and (iii) link the alerts explanations to their source artifacts via the auditable event representation. This approach will allow to shift the paradigm of XAI away from a simple presentation layer to a verifiable aspect of forensic process.

Keywords—Explainable artificial intelligence; explainable machine learning; network intrusion detection; network forensics; digital forensics; incident reconstruction; SHAP; LIME; provenance; timeline analysis.

Contents

1. Introduction	5
2. Literature Review	7
2.1 Introduction	7
2.2 Review scope and critical method	8
2.3 Explainable Machine Learning: Meaning, Methods and Evaluation	9
2.4 Machine Learning for Network Intrusion Detection	12
2.5 Data Validity, Reproducibility and Evidential Integrity	14
2.6 From Intrusion Detection to Explainable Intrusion Detection	17
2.7 Digital Forensic Incident Review	19
2.8 Critical Solutions and Methodological Agenda	21
2.9 Limitations of the Existing Evidence and of the Proposed Agenda	26
3. Methodology	26
3.1 Research Design	26
3.2 Dataset Selection	27
3.3 Data Pre-processing and Target Construction	28
3.4 Experimental Environment	28
3.5 Supervised Machine Learning Models	29
3.6 Model Evaluation	30
3.7 SHAP Explainability Analysis	30
3.8 LIME Explainability Analysis	31
3.9 SHAP-LIME Comparison	32
3.10 Explanation Stability Analysis	33
3.11 Prediction Error and Forensic-Oriented Analysis	34
3.12 Dashboard and Research Synthesis	34
3.13 Methodological Scope	34
4. Results	35
4.1 Overview	35
4.2 Supervised Intrusion-Detection Performance	35
4.3 SHAP Explanation Results	36
4.4 LIME Explanation Results	38
4.5 SHAP-LIME Consistency	39

4.6 Explanation Stability.....	40
4.7 Error-focused XAI analysis.....	41
4.8 Forensic-oriented Evidence Results	42
4.9 Summary of Results	43
5. Discussion and Research Contribution	44
5.1 Overview	44
5.2 Discussion of Intrusion-Detection Performance	44
5.3 Discussion of SHAP and LIME Features.....	45
5.4 Cross-XAI Agreement and Explanation Stability	46
5.5 Error Analysis and the Limits of Explanation	47
5.6 XAI and Forensic-Oriented Analysis	47
5.7 Research Contribution	48
5.8 Research Questions Revisited	49
6. Limitations and Future Work	50
6.1 Limitations.....	50
6.2 Future Work	52
6.3 Chapter Summary	53
7. Conclusion.....	53
REFERENCES	55
<i>APPEDICES</i>	58
Appendix A: Machine-Learning Model Evaluation	58
Appendix B: SHAP and LIME Explainability Outputs.....	59
Appendix C: Explanation Stability	61
Appendix D: Research Dashboard.....	62

1. Introduction

The continuing development of computing and networking technology increases the complexity of modern cyber environments and makes them more prone to adversarial activity. Cyber-attacks can lead to the compromise of confidentiality, integrity, and availability of information which may result in financial, operational and reputation losses. Thus, detecting malicious network activity has become one of the important goals of cybersecurity research. Machine-learning algorithms have grown in popularity due to their ability to learn patterns in data and identify suspicious activity which is hard to detect via conventional static rules [1]. However, the effectiveness of machine-learning based intrusion detection depends not only on the predictive power but also on the properties of the data which is used for the model training and testing [2].

A Network Intrusion Detection System (NIDS) detects network activity characteristic of attack behaviour. Machine learning formulates the intrusion detection task as a classification problem which relies on the ability of a model to classify a network flow as attack or benign using its characteristics. In the presence of labelled instances, supervised learning can be applied to train a classifier on known examples of both classes of flows. This work uses the CICIDS2017 dataset which was created with the goal of providing labelled data on benign traffic along with some modern attack types and a wide set of flow features [3]. The current study focuses on the subset related to DoS attacks and formulates the detection task as binary classification of benign and attack flows.

While high-classification performance is desirable, accuracy alone does not give much information about the rationale of model's predictions. This aspect becomes even more important in the field of cybersecurity because a model can perform well overall while still making some mistakes resulting in false positives or false negatives. A false positive will divert the attention of investigator to legitimate traffic while a false negative will let malicious activity go unnoticed. Therefore, understanding the reasons for a model's predictions is particularly valuable in case of machine-learning applications in cybersecurity domain.

Explainable Artificial Intelligence (XAI) is used for analysing factors which influence machine-learning predictions. LIME and SHAP are two popular post-hoc XAI techniques which create

local explanations of predictions. LIME creates a locally interpretable surrogate model around a particular prediction [4] while SHAP gives feature attribution values showing contribution of each feature in predictions of a model [5]. The use of XAI allows a security analyst to go beyond a binary prediction of Attack or Benign and understand the network flow characteristics which influenced the decision.

However, creating an explanation does not guarantee its validity and the correctness of a prediction. Different XAI approaches can highlight different influential features and produce different results depending on the choice of samples, their perturbation and even training conditions of a model. Therefore, the research on XAI for network intrusion detection has focused on such properties of XAI approaches as stability and robustness rather than explanation visualizations alone [6]. It results in an important distinction between detection performance, explanation validity and validity of the underlying security analysis.

This issue becomes even more crucial in the context of digital forensics. A digital forensic analysis involves the systematic examination of artefacts and evidence which is sufficient to draw reasonable conclusions about events [7]. An XAI result should not be taken as forensic evidence itself but rather serves as a tool which allows drawing attention to network flow characteristics and source artefacts which need further investigation. Additional context may include such information as time stamps, identities of source and destination nodes, identifiers of flows and packets and additional system logs.

Thus, the purpose of this research is the evaluation of explainable machine learning techniques for network intrusion detection with further evaluation of the ability of these techniques to produce results which can help with the digital-forensic analysis of incidents. Decision Tree, Random Forest and XGBoost models are evaluated in the consistent framework, and then the XGBoost model is selected for further explainability evaluation based on its performance in the experiments. XGBoost is a widely used and scalable approach of boosting of decision trees which is particularly suitable for structured classification problems [8]. LIME and SHAP are applied to the selected model and the produced explanations are compared in order to check whether they reveal similar influential network flow features. Stability of explanations under different experimental conditions is also analyzed.

The novelty of this research consists in moving beyond the evaluation of model performance and feature importance in the context of network intrusion detection. By analyzing the explanations of true positive, false positive and false negative predictions, one can understand not only the rationale of correct predictions but also get insights into apparently convincing yet incorrect explanations. The selected explanations are further linked to the source evidence in order to analyse their usefulness in the context of forensic analysis. Thus, this research does not present any novel combination of XGBoost, SHAP, LIME and CICIDS2017. It evaluates these components via a comprehensive research workflow

including model comparison, cross-XAI comparison, explanation stability evaluation, analysis of prediction errors and linking explanations to the source evidence.

2. Literature Review

2.1 Introduction

Network intrusion detection and digital forensic investigation are two different stages of the same problem. The first aims at recognition of some suspicious activity during its occurrence or shortly after, the second at determination of what had happened, how, by what systems and users and with what support of evidence. Machine learning has helped intrusion-detection systems (IDSs) identify more and more sophisticated patterns in the network traffic. At the same time the increased predictive power of machine learning has made one of its old challenges even more acute: the fact that some of the most powerful classifiers are non-interpretable models that do not share any details of their decision process. An IDS may confidently predict that a flow is malicious but give no idea of why and what evidence must be considered next.

The problem is commonly believed to have an explainable machine learning solution. XAI includes intrinsically interpretable models, local surrogate methods, additive feature attribution and gradient-based approaches. Its value in the field of cybersecurity consists in the hope that explanations could help an analyst discern between true alerts and false positives, recognize suspicious traffic attributes, debug a model or determine what logs are worth saving. At the same time the concept of explanation is often over-generalized. Belle and Papantonis distinguish transparent from opaque modeling, global from local understanding and model-specific from model-agnostic explanation. They highlight that the quality of explanation is always user-dependent and task-dependent [1]. In their study Yan et al. prove that cybersecurity explanations depend on robustness, trust, feature semantics and lack of common criteria for evaluation [2]. It is important because a technically accurate explanation of a classifier output does not become an explanation of a security event per se.

It becomes obvious when the specifics of digital forensic investigation are taken into consideration. Forensic analysis is built on preserved artefacts, documented chain of processing, temporal ordering, corroboration and possibility to reproduce or refute an inference. Traditionally, network forensics paid much attention to the preservation of traffic and communication reconstruction to derive investigative information [14]. Timeline-reconstruction research has emphasized the representation of an incident as events and relations, not a single label [15]. Meanwhile many studies of explainable intrusion-detection pay much attention only to the ranking of the influential features. The output shows that

destination port, flow duration or packet statistics has influence on the model, but gives no clue who initiated this activity, what process was involved, what caused it and is it possible to have a benign explanation of the same observation

This paper reviews recent research at the intersection of three areas: explainable machine learning, machine-learning-based network intrusion detection and digital forensic incident reconstruction. Its aim is not in the cataloguing of particular algorithms and reproduction of accuracy numbers. Rather it aims to assess the assumptions underlying their interconnections. There are four questions that organize the review. Firstly, what is explained in a machine learning-based IDS. Secondly, how trustworthy is the empirical evidence that supports the modern models. Thirdly, what extra requirements are introduced in case of using an alert for forensic reconstruction. Fourthly, what kind of research design is required for assessing the transition from a model prediction to a defensibly evidential account of an incident. The review shows that there is a certain separation of the structure of the research in each of the three domains. XAI research develops explanation methods and concepts of evaluation. NIDS research develops increasingly accurate classifiers, usually based on benchmarks. XAI-NIDS research applies various methods of explanation, such as SHAP to the alerts. Digital forensic research develops timelines, provenance models and forensic investigation processes. All three are valuable but the interconnection is still incomplete. The unanswered question is if an explanation, which is technically correct with respect to a classifier is stable, comprehensible and sufficiently connected to evidence for an incident reconstruction. Such a problem cannot be solved by simple addition of an explanation graphic to a high-performing model. What is required is the research design with data provenance, model validation, explanation quality, temporal reasoning and analyst performance assessment.

2.2 Review scope and critical method

The corpus of uploaded papers consisted of thirty PDF documents focused on XAI, NIDS and digital forensics. The critical narrative review has been chosen because the literature included conceptual reviews, systematic reviews, empirical classification studies and forensic frameworks that cannot be compared quantitatively through the meta-analysis. The screening has been done according to the relevance of title and abstract, publication status, methodological detail and contribution to the relationship between the three dissertation domains. Twenty-one papers have been selected as core evidence. Nine papers have been left out of the argumentation because they concerned medical risk prediction, materials science, **nanophotonic**, point-cloud classification and other areas whose empirical findings cannot be reasonably applied to cybersecurity and forensic investigation. Several papers with uncertain publication status or limited direct relevance have been excluded as well. This

prevents unnecessary inflation of the bibliography with the sources which do not relate to the research problem.

The selected literature can be divided into four analytical categories. The first category relates to XAI principles, cybersecurity explainability and comparison of explanation methods [1], [4]. The second category relates to machine learning and deep learning for intrusion and anomaly detection. This includes surveys, benchmarking studies and empirical proposals of models [5], [13]. The third category relates to the direct application of XAI in intrusion detection, specifically the SHAP-based framework of Wang et al. [4]. The fourth category includes network forensics, timeline reconstruction, alert reduction, AI in digital forensics and emerging explainable forensic frameworks [14], [21]. It is important to treat explainable intrusion detection as a distinct intersection category, otherwise direct evidence on XAI evaluation in the security will be overshadowed by surveys of either XAI or NIDS. Papers have been read comparatively with the unified set of critical questions. For the studies related to models the review analyzed the dataset, preprocessing, class distribution, validation design, performance metrics and the potential of generalization. For XAI studies the following questions were considered: what has been explained, whether it was local or global, which assumptions were used and was the quality of the explanation tested. For the forensic studies the review analyzed how evidence was preserved, how events were represented, whether provenance and uncertainty were maintained and whether the reconstruction could be reproduced by another investigator. Special attention has been paid to the claims that were beyond the provided evidence. For example, a paper may demonstrate excellent benchmark classification, but fail to show operational detection; it may produce intelligible plot of features but fail to show the benefit for an analyst; or it may automate artefact classification without incident reconstruction. This review does not pretend to be a complete systematic review of every paper indexed by Google Scholar or IEEE Xplore. The source set has been provided by the researcher and therefore it reflects the selection that was made prior to the review. It also contains more work on detection than on human factors and legal admissibility. These limitations are recognized rather than hidden. Nevertheless, the corpus is enough to reveal the patterns of methodology and the research gap. The argument is enhanced through the triangulation of different evidences: the surveys show the recurrent problems, the benchmarking demonstrates their empirical consequences and forensic frameworks show the insufficiency of the classification-level explanation for **reconstruction**.

2.3 Explainable Machine Learning: Meaning, Methods and Evaluation

Explainability is not a binary trait; rather, it depends on several aspects. For example, Belle and Papantonis define the concept of interpretability with reference to three elements – the model itself, the explanation process, and the target user [1]. While small decision trees lend themselves to inspection, deep networks require a post-hoc explanation. Similarly, a global

explanation concerns the whole model behaviour, whereas a local explanation applies to a specific instance. Some explanation methods depend on the model internals and, thus, are tailored to certain predictors, while others treat them as a black box. Such differences should inform the evaluation of the method. First, local explanations cannot verify consistent behaviour of the model globally, and second, a global feature ranking is insufficient justification for issuing alerts.

In cybersecurity, another dimension is added as features have operational meanings and dependencies between them. Network-flow variables like packet count, total number of bytes, duration, average rate, and inter-arrival time are not independent descriptive attributes. Manipulation of one feature without adjustment of the others will produce an impossible or out-of-distribution data point. The problem of instability, susceptibility to attacks, and difficulty to convert an explanation into understandable domain concepts are noted by Yan et al. in the survey of XAI in cybersecurity [2]. The importance of a feature depends on context: destination port becomes suspicious in combination with protocol, service, source role, and a particular timing, but not otherwise.

An example of a direct comparison of explanation methods is the uploaded paper of Nguyen et al. on LIME, SHAP, and class-activation map explanations on images [3]. While the general conclusion that some methods produce more detailed explanations at higher computational costs while others produce less detailed explanations faster is relevant, there is no particular relevance of the results to cybersecurity. Namely, explanation methods work in different domains with different types of perturbations, which makes direct comparisons irrelevant. Explanations of images rely on spatial proximity between pixels and visual regions, while explanations of network flows depend on protocols and constraints of tabular data.

Thus, the paper provides a general observation about the differences in the cost and detail of explanation methods but fails to prove that the same order will be applicable for NIDS data. LIME and SHAP are dominating much of the current applied XAI. LIME tries to approximate the behavior of the model in proximity to an example through a simpler surrogate and, therefore, offers both the flexibility of model-independence and concise explanations. On the other hand, LIME's downsides are the sensitivity to the notion of proximity and perturbations. In network data, naive perturbation of the data point disrupts semantic dependencies and makes the surrogate a poor model only locally. Repeated experiments can result in choosing different sets of features.

These are not reasons for not applying LIME but reasons why these factors need to be considered in the experiment. SHAP uses game-theoretic principles and provides additive feature attributions. In practice, SHAP allows producing both local explanations and aggregating them to produce a global summary of a model, including fast versions for tree-based models. That is why SHAP is appealing for tabular intrusion datasets and used in the framework of Wang et al. [4]. However, the seeming numerical precision of SHAP values is deceiving because the attribution reflects only the predictor and the reference distribution

but not the actual contribution to the cause of the incident. Correlated features can redistribute or share the contribution in ways depending on the conditional distribution assumed by the method. Thus, the highly ranked feature might not contribute to the real-world incident but to the dataset artifact.

One of the clearest demonstrations of the XAI in intrusion detection is provided by Wang et al. [4]. The work employs SHAP to provide local and global explanations of the features and connections of model predictions and interpretable traffic characteristics. This work is important as it is an attempt to move beyond the reporting of the classification performance and incorporate the interpretation as part of the IDS design. Furthermore, this work demonstrates that explanations can help in debugging of models. However, one needs to be careful with the evidential claim made by the work because it uses NSL-KDD – a benchmark dataset that came from the older era of intrusion-detection research. Namely, the work produces feature attributions and does not investigate whether the explanation helps analysts in making investigative decisions, whether explanations are stable when the model is retrained on a new dataset or the data distribution is changed, and whether the attributed features can be traced back to a particular event sequence.

One of the common problems in applied XAI is confusing plausibility with faithfulness. Namely, the explanation seems plausible when the highly-ranked features correspond to what a domain expert would expect. Plausible explanations might help in communicating with domain experts, but this is not enough to prove that the explanation is a faithful representation of the model's behavior. The method could generate a believable story while the model depends on an entirely different decision boundary. In the contrary, the faithful explanation might show that the model works in an unintuitive way, but this information is nevertheless true and should be investigated.

That is why in the evaluation of the explanation method, one needs to consider at least four aspects: whether the prediction is correct, whether the explanation represents the model, whether the user understands the explanation, and whether the explanation allows making valid real-world inference. Technical evaluation could cover some of these questions. Faithfulness can be assessed by removing highly ranked features and investigating how the model output changes as expected. The explanation's stability could be checked in different runs and for nearby examples. Robustness could evaluate how disproportionately the attribution changes in response to a change of the input. Sparsity could be evaluated as ability to express the explanation with few factors. Consistency could compare the rankings of features attributed by different methods. The computational latency could be evaluated for operational applications, particularly, for high-volume alerts.

However, each of these evaluation aspects is not sufficient alone, and each is dependent on the definition of a valid perturbation. Human-centered evaluation is equally important because readable explanation is not necessarily helpful. Users might rely on explanations too much, especially, if there is a high-confidence score accompanying it. Namely, the goal is

to achieve calibration of reliance – that the users trust the explanations and recommendations provided by them if they are supported by the evidence but challenge the explanation that is weak or contradicts some facts. Appropriate experiments could compare alert-only, alert-plus-XAI, and alert-plus-XAI-plus-evidence conditions using the tasks such as attack identification, evidence selection, and next-step determination. Accuracy of decision, time, calibration of confidence, and error detection would be more important than liking of a particular graph in the experiment.

Thus, XAI research provides necessary tools but exposes the dangers of category errors. Namely, the model explanation is the statement about the computational decision made by a model under a particular representation. The security interpretation is the human-made judgment that links this statement with domain knowledge. A forensic conclusion is the inference about the past events based on preserved artifacts and reproducing the reasoning. Therefore, a rigorous evaluation should keep these stages apart.

2.4 Machine Learning for Network Intrusion Detection

Machine learning is applied to detect intrusions in computer networks because fixed signatures have difficulties dealing with novel, modified or changing attacks. Anomaly-based approaches try to learn patterns of normal or malicious behaviour and classify deviations. Supervised classifiers, clustering, one-class classifiers, ensembles, convolutional networks, recurrent networks and generative models are mentioned in the literature. The paper by Bou Nassif et al. gives a comprehensive systematic review of the research on anomaly detection, which is a problem covering a wide range of data types and learning paradigms [5]. Their synthesis is helpful for understanding the place of NIDS in the larger anomaly-detection context, but it shows clearly why any one algorithm cannot be considered superior. The effectiveness of the algorithms is highly dependent on the type of anomaly, representation, labelling and operational costs.

The popularity of random forests was driven by their capability to model nonlinear relationships and handle mixed feature importance and variance reduction in ensembles. Resende and Drummond give an overview of the research applying random forests to Intrusion Detection Systems and note the strong performance of such models and consistent issues of datasets choice, feature selection and inconsistency of evaluations [6]. One of the reasons behind the popularity.

The deployment gap is developed further by Pacheco et al. in their survey of machine learning for network traffic classification [8]. They find that academic studies frequently rely on static, labelled datasets and offline experiments, while real networks involve encrypted traffic, changing applications, scarce ground truth, concept drift and resource constraints. These findings are directly relevant to NIDS. A model trained on one institutional or laboratory distribution may deteriorate when protocols, services, user behaviour or

adversary tactics change. The same problem applies to explanations: a stable feature attribution in a benchmark may not remain meaningful when the feature distribution shifts.

Early machine-learning NIDS studies demonstrate both the opportunity and the risk of benchmark-driven claims. Lee et al. present a machine-learning-based intrusion-detection approach and report strong results in an experimental setting [9]. Such work helped establish the viability of automated classification, but comparisons with contemporary systems are difficult because datasets, attack definitions and validation practices have evolved. Bhati et al. report very high performance for an XGBoost ensemble on KDD Cup 99 [10]. XGBoost is a strong candidate for tabular NIDS because it models nonlinear feature interactions efficiently. However, KDD Cup 99 contains known redundancy and reflects traffic and attacks from a much earlier period. Near-perfect accuracy on that dataset should therefore be interpreted as evidence of fit to the benchmark, not proof of current operational detection.

Research provide a more revealing benchmark by evaluating thirty-one machine-learning algorithms on CICIDS2017 [11]. Their study demonstrates that aggregate performance can conceal substantial variation among attack classes. No single algorithm reliably dominates every category, especially under severe class imbalance. This is a crucial corrective to papers that report only overall accuracy. In practice, the cost of missing a rare infiltration or web attack may be much greater than misclassifying a common benign flow. Macro-averaged precision, recall and F1-score, class-specific confusion matrices, false-positive and false-negative rates, and precision-recall curves are therefore necessary.

Class imbalance has encouraged the use of generative methods. Park et al. propose an enhanced AI-based NIDS using generative adversarial networks to create additional minority-class examples [12]. Their work addresses a genuine limitation: classifiers trained on skewed data may learn the majority classes while treating rare attacks as noise. Synthetic augmentation can improve representation, but it introduces a new validity question. A generative model may reproduce the statistical surface of the training set without generating semantically realistic network behaviour. If synthetic records violate protocol relationships or amplify dataset artefacts, the detector and its explanations may become more confident without becoming more generalisable. Evaluation should therefore assess not only class balance and classifier scores but the realism and diversity of generated traffic.

Research combine convolutional and long short-term memory networks in a hybrid CNN-LSTM NIDS [13]. The architecture is designed to capture local feature patterns and sequential dependencies, and the paper reports strong performance across benchmark datasets. This type of model is representative of the current accuracy-interpretability challenge. It can model relationships that simpler classifiers miss, but the resulting internal representation is difficult to inspect. Post-hoc methods can identify influential inputs, yet they may not explain how temporal dependencies were encoded. Moreover, high

benchmark scores do not directly show robustness to drift, adversarial manipulation or different network environments.

Across these studies, dataset selection is the dominant threat to external validity. Older datasets facilitate comparison but poorly represent current applications, encryption and attack tactics. Newer datasets improve traffic diversity but remain laboratory products with fixed labels and bounded scenarios. Random splitting can allow closely related records from the same capture period to enter both training and test sets, inflating apparent generalisation. Temporal splitting, attack-scenario holdout and cross-dataset evaluation provide stronger evidence, although differences in feature schemas complicate comparison. A controlled local incident scenario can complement public datasets by providing known ground truth and traceable logs, but it should not be presented as a substitute for operational diversity.

Preprocessing is another source of hidden variation. Feature cleaning, encoding, normalisation, class merging, sampling and feature selection can change the task substantially. A model may achieve high accuracy because identifiers or capture-specific fields leak class information. Feature selection can improve efficiency and reduce overfitting, but it can also remove variables needed for forensic interpretation. Research should record each transformation and retain a mapping from derived features to raw evidence. Without that mapping, the model may be explainable at the feature level but disconnected from the artefacts an investigator can verify.

Operational evaluation must include cost and timing. A NIDS that produces too many false positives creates alert fatigue and delays investigation. A highly accurate model that requires expensive preprocessing or slow explanation may be unsuitable for real-time use. Calibration also matters: a probability score should reflect empirical uncertainty, not merely a softmax output. Detection latency, throughput, memory use and explanation time should be reported alongside predictive metrics. These considerations lead to a broader conclusion: the best model for forensic support may not be the model with the highest isolated F1-score. It may be the model whose errors are understood, whose explanations are stable, and whose features can be traced back to preserved evidence.

2.5 Data Validity, Reproducibility and Evidential Integrity

Quality of an explainable system is limited by quality of its data. Although this statement can be mentioned from time to time in literature on model development, it is very seldom applied all the way to explanation and forensic stages. Intrusion dataset is not a neutral representation of the reality. On the contrary, it arises from the capture environment, attack scripts, network roles, collection method, labeling rules and feature extraction choices. These factors shape what model learns and, as a consequence, what explanation attributes next. An explanation may be technically faithful to the model but operationally meaningless if the model learned something in dependence on the way dataset is built.

An example of such a problem is dataset leakage. Identifiers, timestamps, duplicated flows or capture-related features can disclose the class label of a flow. Random train/test split can put almost identical flows from the same session to both parts of the split. In this case, the model is validated on familiar patterns and not on truly independent activity. It can achieve a good score using a number of algorithms and give an illusion of a robust security signal. An explainer can reinforce this illusion, attributing importance to the feature which leaks class label. The proper reaction here is not to delete explanations found by investigator but to design validation procedure in the way which minimises leakage.

Temporal validation is especially relevant for the purpose. Network behavior changes with changes in users, applications and infrastructure, and also with adaptation of adversary to new circumstances. Chronological validation tests ability of the model trained on the previous traffic to predict the following one. Attack scenario holdout tests generalization to the related but unseen scenario of attack. Cross-dataset testing is even more powerful but harder to do due to feature schema mismatches. Pacheco et al. state that research for deployment must take into consideration the problems of concept drift, lack of labels and changes in traffic conditions [8]. This means that explanation generalization must be evaluated together with prediction generalization. The model can have a good F1-score, but it can rely on the different features. Also, it can have the similar attributions but deteriorated calibration.

Class imbalance is another connection between prediction and explanation. Maseer et al. show large performance differences for the different types of attacks in CICIDS2017 [11], while Park et al. use generative augmentation to increase representation of minority attacks [12]. Thus, the evaluation of explanations just on the basis of abundant classes will make them seem reliable while they fail on rare and crucial attacks. It means that the sample of explanations must be stratified by attack type and error category. Explanations for minority classes must be evaluated by stability and evidentiary value. Synthetic examples must be separated from the truly captured ones in order to let an analyst know where the pattern of explanation comes from - from the observed traffic or from the generator.

Ground truth is more complicated than the simple class label. Public NIDS datasets usually indicate whether a record is benign or belongs to one of the attack categories, but forensic reconstruction needs the event level ground truth: when the action started, what host initiated it, which artifacts were created and in which sequence. Classification label cannot answer these questions. Controlled scenarios must keep an independent scenario log with all the planned actions of the attacker, execution times of the attack tools, involved assets and expected artifacts. Scenario log must not be used as a feature of the model; it is an external ground truth for evaluation of the reconstruction.

Reproducibility of research requires more than just publishing code. The exact version of the dataset, its hashes, filtering rules, definition of features, splitting indices, software versions, random seeds and trained models must be kept. Explanation configuration must also be recorded. For SHAP, it is the type of the explainer and background/reference set. For LIME, it is the sampling size, kernel width, distance metric, discretization and feature selection settings. For gradient-based methods, it is the baseline and integration parameters. Two investigators who use the same model with the different explainer backgrounds can provide different explanations for the same alert. In forensic context, such difference must be visible and reviewable.

Integrity of evidence requires additional restrictions which do not exist in typical machine learning benchmarks. Raw packet captures and logs must be kept in read-only format and be hashed at the moment of acquisition. Feature extraction and analysis must be done on the verified copies of this evidence. Each derived artifact must contain information about its source and transformation. At the moment when alert is generated, the system must save the version of the model, input vector, timestamp, predicted probabilities and other configuration parameters. At the moment when explanation is generated, it must be associated not with the retrained or updated model, but with the exact prediction. It will prevent an investigator from explanation generation using the model different from the one used to generate the prediction.

Timestamps require special treatment. Network, server and endpoint clocks may be different; events may be logged at different stages of processing; and there may be time zone differences or daylight saving changes. A timeline sorted down to milliseconds creates an illusion of precision. The reconstruction must contain information about the source of the timestamp, known offset and uncertainty interval. The events can be sorted to the categories of definitely before, definitely after and temporal overlapping when evidence does not allow to establish the sequence. Such uncertainty is not considered in explainable NIDS research because flow record is treated as a fixed row, but forensic interpretation cannot ignore it.

Data minimization and privacy protection must be balanced with forensic readiness. Long term retention of traffic can reveal credentials, communications and personal information, the problem which is already discussed in early works on network forensics [14]. Research environment must collect only the data needed for experiment, limit the access, protect anonymity in reporting and document retention periods. However, over-anonymization can deprive the system of entity relationships needed for reconstruction. The defensible solution must be the use of the pseudonymous identifiers which allow to identify entity in different sources while protecting its personal identity. Mapping must be stored separately and have controlled access.

There is an analytical aspect of reproducibility as well. Another examiner must be able to look at the same evidence and to distinguish observations from automated or human

inferences. Deterministic pipeline helps, but does not solve the problem. Event representation must allow annotations, confidence and alternative hypotheses. When analysts disagree, the system must keep both interpretations instead of silent overwrite. It follows the forensic principle that conclusions can always be questioned.

These data and integrity requirements change the research question. It is not a problem of choosing SHAP or LIME to get a better explanation in the notebook. It is the problem of generating a reproducible explanation for the exact historical prediction and tracing it to the unmodified evidence without overstating temporal and causality. By taking data governance as an integral part of explainability, the proposed research escapes from the weakness typical of many projects which use impressive interface with an undocumented evidence pipeline.

2.6 From Intrusion Detection to Explainable Intrusion Detection

Moving from classic Network Intrusion Detection Systems (NIDS) to explainable NIDS is commonly described as adding an explanation layer to an existing classifier. Though this description is technologically useful, it is conceptually inadequate. Explaining requires data acquisition, feature selection, modelling, and an explainer assumption about what a model means. If any of the steps is unreliable, then the explanation may lead to drawing a wrong lesson with unnecessary confidence. Hence, explainable intrusion detection should be considered a system characteristic rather than a visualization tool.

SHAP framework, developed by Wang et al., represents an important landmark example [4]. It proves that local explanations help to understand which features drove the given flow towards an attack class, while global summary explains model's patterns. This facilitates debugging and helps security analysts to evaluate if the model uses reasonable traffic characteristics. The contribution of this work is building an interpretable bridge between classifier and user. Its shortcoming is that the interpretation stays close to the model, and the framework does not ensure that the explanation brings an investigator to the right sequence of packets, to the right log or to the right attack stage.

Yan et al.'s survey about cybersecurity is the source of an important insight regarding this limitation [2]. It shows that explainability can assist in finding vulnerabilities, analyzing malware, detecting intrusions and other security tasks, but the community lacks the unified metrics and user evaluation. The explanation also exposes the detector to attacks since it provides details about influential features, and attackers can use them to fool the detector. Thus, explainability is a dual-use instrument that allows for enhancing accountability but increasing the attack surface of the decision process.

Another issue is the semantic validity. Explainers perturbate the features to measure their influence on the decision. There are constraints between protocols, ports, byte sums, packet numbers, duration, and flags. If the explainer treats the features as independent, it can generate unrealistic samples. Then the surrogate can look like a good approximation around

the improbable point. The robust XAI-NIDS technique should either apply constraints-aware perturbations or generate alternatives using valid transformation. In the case of categorical variables, the transition options should respect the protocol. The continuous features should have boundaries and relationships that can be derived from the normal traffic or specified by network.

The stability of the explanation should also be measured in different ways. The computational stability refers to the consistency of the attribution results under the repeated executions of the algorithm. The local stability refers to similarity of semantically similar flows' explanations. The model stability refers to the changes of the explanation after the model retraining with another seed or sample. The temporal stability refers to persistence of the explanations in time. The method can be deterministically stable for the fixed model but operationally unstable due to changing the feature base after the model retraining. Security analysts need to understand if the change in the explanation comes from real changes in the attacks, model's drift, or from the explainer instability.

The relation of the explanation to the false positive alerts needs to be examined separately. In the anomaly-based IDS the unusual activity can cause alert even if it is not malicious. The explanation can show what were the unusual features that is useful information, but it cannot prove the intent of maliciousness. The analyst needs to correlate the flow with asset role, maintenance windows, user's actions and other events. Thus, the explanation should be offered as a tool for generating hypotheses. The interface should distinguish feature contribution, model's confidence, supportive artefacts and analyst's conclusions instead of merging them in the one authoritative story.

Explainable NIDS should support a diversity of users. The developer needs diagnostics that includes feature dependence, residual error and global behavior. The security operations analyst needs a concise explanation for the prioritizing and next evidence to investigate. The forensic examiner needs a traceable way to reconstruct the path from alert to the raw artefact and the history of the model and explainer's parameters. The manager can require the aggregated information on the risk rather than low-level packet features. One explanation format cannot serve all those purposes equally well. The evaluation of the method must define the user before considering it understandable.

This important methodological consequence is that the quality of explanation should be assessed both for the correct and incorrect predictions. Many studies demonstrate the explanations for the correctly detected attacks, that can create confirmation bias. The false positives show whether the explanations can help to reject alerts; the false negatives reveal what the model did not detect; confidently wrong classifications test whether users can challenge the system. Including those cases is crucial for assessing proper reliance. The system that raises the trust in every output can worsen the security even if users perceive it as easy to use.

Thus, the XAI-NIDS literature gives a reason for cautious optimism. The feature attribution can reduce the opaqueness of the model, find spurious dependencies and improve the communication. But the evidence does not allow to consider explanations as incident stories. The next step of research should connect explanation outputs to evidence retrieval and reconstruction tasks while maintaining uncertainty and distinguishing model reasoning from historical facts.

2.7 Digital Forensic Incident Review

Inference in digital forensic incident reconstruction is the task of inferring past events based on potentially partial, heterogeneous, and sometimes questionable artefacts. Unlike classification, where inferences are pointwise in nature, reconstruction is relational and temporal; it studies relationships between observations, identifies possible temporal sequences, judges their evidential support, and measures the residual uncertainty. While network traffic data can show that two machines communicated with each other, a proper explanation usually needs host logons, authentications, process actions, and context to explain the reason behind the communication and actions following it.

An early study by Corey et al. on the challenges of network forensics [14] highlights the importance of archiving network traffic and reducing the data into information for investigation. The technical context may have changed since 2002, but the basic requirement remains relevant. Real-time monitoring cannot always reconstruct the past, and an alert of a detector does not preserve all the data needed for further analysis. Therefore, network-forensic preparedness depends on capturing, storing, indexing, and protecting the data from alterations. In turn, these requirements create a practical tension with privacy, storage costs, and encryption.

Chabot et al. offer an ontology-based approach to reconstructing and analyzing incident timelines in the digital domain [15]. The authors formalize the entities, events, and relationships in order to allow integration and query of heterogeneous evidence. The novelty in this approach lies in treating reconstruction as knowledge representation rather than mere ordering of the timestamps. Provenance and semantic relationships help the investigator trace the reconstructed event back to its sources. Moreover, the approach allows better communication and repeatability due to the common structure of events. The problems include building an appropriate ontology, integration of sources, and the computational cost caused by increasing volume of the evidence.

The distinction between low-level records and high-level events is crucial. Flow record, DNS query, and file timestamp are examples of observations that were created by the collection process. Reconnaissance, credential attacks, or command-and-control actions are examples of events – interpretations built out of multiple observations. A good reconstruction will preserve the transformation from one level to another. For instance, if a machine learning

model classifies the traffic flow as command-and-control, the label should be preserved as an inference related to the original record, not as a fact put in the timeline.

Akremit et al. offer an approach to IDS alert reduction for forensic preparedness [16]. The alert volume is indeed an issue, as duplicated, low-value, and poorly contextualized alerts may hide those that relate to the key events of interest. Thus, reducing the number of alerts can improve efficiency; however, doing so aggressively may cause discarding evidence about a multistep attack. Therefore, the process of forensic-ready alert reduction should be reversible and auditable, preserving original alerts while recording the correlation or suppression rules and allowing the examiner to recover the basis of the reduced view.

The Particle Deep framework proposed by Koroniotis and Moustafa combines particle swarm optimization with deep learning for network forensics in IoT environments [17]. This research is valuable insofar as it frames the problem of detection in the context of network forensics and recognizes the heterogeneity and volume of IoT traffic. Moreover, the work illustrates the transition to automated feature selection and deep classification.

Nevertheless, its main evaluation is still predictive. Optimizing the detector for attack classification does not by itself validate the event reconstruction, evidence provenance, or the analyst reasoning. This framework serves as an example of a bridge between NIDS and network forensics, but not as a complete evidential process.

Fähndrich et al. conduct a structured literature review on strong artificial intelligence (AI) and digital forensics [18]. They state that AI can assist with the analysis of large amounts of evidence and complicated analytical tasks, yet they recognize several issues, including explainability, bias, accuracy, robustness, and legal usability. The value of this paper lies in its awareness that technical performance of an AI method cannot be evaluated only by its predictive utility if it can influence forensic or legal decisions. Moreover, the integration of advanced AI with mature forensic requirements is still a developing area.

Alam and Demir apply the explainable AI to cyber-forensic file fragments classification [19]. The paper uses textual weighting and local explanations in order to make file type prediction more interpretable. Thus, this work illustrates that XAI can explain which patterns of bytes help to classify an artefact and consequently help the examiner understand the automated evidence triage process. Nevertheless, the scope of the problem remains limited as it illustrates the difference between artefact explanation and incident reconstruction. For instance, correctly recognizing a file fragment is helpful, but it is still not enough to determine its creation time, how it appeared in the system, which user has accessed it, and how it is related to the network attack.

GeoForensic-AI is one of the most sophisticated recent attempts to integrate explainable forensic AI in the modern digital ecosystem [20]. The approach combines geospatial reasoning, temporal graph modeling, attribution, and distributed learning, achieving strong benchmark results. The value of this framework is in its intention to move from the isolated

classification to multi-source forensic analysis. However, the claims should be carefully qualified as follows: the evaluation is based on public benchmark data, not the independent operational cases; the IP geolocation may be inaccurate due to proxies or virtual private networks; the temporal graph part is computationally expensive; and the federated design presumes some communication conditions that do not always hold. Importantly, this framework has not been independently replicated and validated by the examiners yet, thus, it should be considered as an emerging research direction.

The literature review on cybersecurity incidents in the water sector conducted by Hassanzadeh et al. adds the incident-level perspective to the discussion [21]. The work demonstrates that real incidents have the organizational, physical, and operational consequences that cannot be determined from network-flow labels. Public incident reports are incomplete and sometimes lack technical details, but they show the importance of context, system architecture, and cascade effects. For the critical infrastructure, reconstruction must establish connections between cyber activities, process changes, and human decision-making, and this once again demonstrates the need to include attribution in the explanation.

Some persistent forensic requirements appear in the reviewed literature. First, provenance: every derived feature, model output, and inferred event must be traceable to the source artefacts and processing steps. Second, temporal consistency: the timestamps should be normalized, the uncertainties should be recorded, and the causal sequence should be established, not merely chronological order. Third, reproducibility: the version of the model, preprocessing steps, parameters, and explainer settings should be preserved in order to be able to repeat the analysis. Fourth, corroboration: the results of automatic classification should be cross-checked with independent network, host, or application evidence. Fifth, consideration of alternative hypotheses: the reconstruction must allow multiple narratives and not impose a single one. Sixth, chain of custody: the evidence integrity must be preserved from acquisition to reporting.

These requirements are more strict than usual XAI evaluation. A correct attribution can still be weakly forensically due to nontraceable features. Stable explanation can be misleading due to errors in timestamps. Understandable explanation can result in incorrect incident hypothesis due to missing host context. The basic question is not the transparency of the classifier in the abstract terms but improvement of the well-documented and challengeable process of investigation.

2.8 Critical Solutions and Methodological Agenda

The above-identified gap calls for a methodology that preserves evidence for future comparative purposes. The point here should not be about automating the investigative process, but about determining if explainable machine learning enhances the efficiency,

reproducibility and quality of reasoning about network attacks. The following six connected layers constitute the proposed solution: evidence acquisition, provenance-aware feature construction, comparative intrusion detection, multi-method explanation, evidence-linked reconstruction and analyst evaluation.

The first layer is evidence acquisition. A controlled incident environment should collect packet captures or flow records, firewall, DNS, authentication and selected host events. Timestamps, hashing and documentation of each source are required. Differences in clock offsets and gaps in recording should be noted. Public datasets such as CICIDS2017 are helpful for comparing with existing research on NIDS based on published benchmarks. In addition, another dataset or another incident scenario should allow testing whether the method relies too heavily on a single benchmark. A controlled scenario may include reconnaissance, multiple authentication attempts, initial compromise, command-and-control communication and data exfiltration. The purpose of creating such scenario is not reproducing each real network, but having a scenario with known events and artifacts in order to check its reconstruction.

The second layer is provenance-aware feature construction. Any model feature should have a mapping to the record or aggregation window from which it was constructed. The mapping should include code, parameters and feature version of the transformation. Any feature that leaks identity or capture conditions should be either deleted or explicitly tested. Pre-processing should be done only on training data. If datasets are different, the common set of features will allow conducting cross-dataset comparison, while still evaluating each dataset-specific model within the methodology. Preserving provenance helps solve a problem ignored by most ML pipelines, which is allowing tracing the evidence back from any attributed feature.

The third layer is comparative intrusion detection. A constrained decision tree is an intrinsic interpretability baseline. A random forest or XGBoost model will represent high-performing tree ensemble models due to a very strong record demonstrated in IDS literature [6], [10]. A CNN-LSTM or another neural model will be a representative of an opaque nonlinear architecture [13]. The comparison will be limited deliberately to representatives of model families, instead of searching for dozens of algorithms. Hyperparameters should be optimized using the same validation methodology and computation budget whenever possible.

Evaluation of the model detection will report accuracy as a supplementary metric. Main metrics will include per-class precision, recall and F1-score; macro and weighted average scores; false positives and negatives; area under the precision-recall curve; confusion matrix and calibration. Accuracy scores will be reported on a repeated seed basis. Splits into training and test sets are better done based on time or scenario, instead of the random split. Computation costs for training and inference will be measured, since the value for operations depends on throughput and latency. The study will keep a structured set of true

positives, false positives, false negatives and confidently wrong predictions for explanation and analyst testing.

The fourth layer is multi-method explanation. SHAP and LIME methods are good choices as comparators because they are commonly used and make different assumptions. For the neural model, a gradient-based method will be included, if it can be implemented and justified. The interpretable tree model will be used as a baseline for post-hoc explainers. Explanations will be produced at both local and global levels. Local explanations will be needed for individual alerts; global analysis will help to see dataset artefacts and model dependencies.

Explanation evaluation will include several complementary measures. Faithfulness can be assessed by deleting or changing the feature and measuring the change in the prediction score. But modification should respect feature dependencies. A constraint-aware perturbation can be done using valid ranges, protocol rules and joint feature transformations. Stability can be checked by measuring rank correlation and attribution vector similarity through repeated explainer runs, multiple seeds and near-by valid flows. Robustness tests will include small changes, which should not change the attack semantics. Sparsity can be measured by checking how many features explain a certain proportion of attribution mass. Explanation latency should be estimated on realistic alert volumes. Disagreement between explainers should be reported, not averaged.

The fifth layer is an evidence-linked event representation. Alerts will be represented as inference nodes connected to the source flow, features, model version and explanations. Then, related network and host artifacts will be correlated into candidate events. The representation will distinguish observed facts, measurements, model predictions and hypotheses. Temporal relations will include uncertainty, if any is possible in the clock synchronization or event boundaries. All provenance links will allow tracing claims back to evidence and transformation. A provenance graph or a structured timeline will meet these requirements without implying causality.

Forensic evaluation will use the criteria missing from most XAI-NIDS papers. Event coverage will be measured as the percentage of the known ground-truth incident scenario. Temporal-order accuracy will evaluate the correctness of the reconstructed sequence of events and their dependencies. Evidence precision will check if the selected artifacts are actually relevant. Provenance completeness will check if there is traceable support for each inference. Reproducibility will verify if another analyst or another run of the tool can reconstruct the same output given a recording of the configuration. Uncertainty quality will assess how missing evidence and alternative hypotheses are accounted for.

The sixth layer is analyst-centered evaluation. With an approved ethical research protocol and accessible participants, cybersecurity students, practitioners or forensic examiners can complete short investigation tasks under counterbalanced conditions. In one condition, only

the alert will be presented to them; in the other one, an alert with the model explanation will be shown; in the third, the alert with provenance-linked timeline will be presented. These tasks will involve identifying attack stage, selecting relevant artifacts, rejecting a false positive and choosing the next investigative step. Outcomes will include decision accuracy, evidence precision, completion time, calibration of confidence and its appropriate use. Clarity and workload of analysts are useful secondary criteria, but task performance should not be replaced by them.

Security and adversarial evaluation should be included. Explanation methods can help to reveal weak points of the detector and even manipulate them. The study should test if small evasion-oriented feature changes affect the attribution before the prediction change, or the attacker can modify his classification without altering explanation. Sensitive explanation details should be managed properly during deployment. The goal of this stage is providing the accountability and transparency for authorized analysts, not disclosing the decision boundary of the detector to anyone.

There are several safeguards, which improve validity of the study. Versions of model and explainers, random seeds and background datasets should be recorded. Raw evidence should be immutable; analysis should be performed on the working copy of evidence. Any missing values and failed captures should be documented. Instructions of analysts should avoid leading language. The same cases should be used in different explanation conditions whenever possible. Tests should be chosen according to distribution and study design, and effect size should go with any statistical significance. Negative results should be kept, including attacks for which explanations are unstable or reconstruction is incomplete.

The above-mentioned methodology provides a more meaningful form of comparison than ranking of accuracy. It allows seeing where each model-explainer combination succeeds and fails. A transparent model can be preferable for some attack classes even if the deep-learning model has a slightly better overall accuracy. A technically faithful but slow explanation method may be unusable for triage, but useful for post-incident analysis. An explanation can help with the selection of evidence without improving the classification of attack. These conditional results can become a foundation for recommendations.

The critical solution to the research problem is therefore both architectural and evaluative. Explainability should be integrated into the chain of evidence, not appended after the prediction. Forensic principles should guide data collection and feature construction from the beginning. Analyst evaluation should focus on decision quality and its calibration, not only on trust. The output should not be an automatically produced narrative hiding the uncertainty; it should be an auditable set of observations, inferences and hypotheses for an investigator to examine, reproduce and test.

The review provides the four linked research questions. The first one is the comparison of the interpretability, ensemble and deep learning models on different attack classes based on

the class-sensitive and time-robust measures. This question prevents the methodology from assuming that the most complex model is always the best. The second question is the comparison of LIME and SHAP, and optionally the gradient-based method in faithfulness, stability, robustness, sparsity and latency under the semantically valid network perturbation. The third question is about the effectiveness of explanations for identifying the relevant evidence, rejecting misleading alerts and choosing the next investigative action. The fourth question is if the explanation outputs can be linked to network and host artifacts in provenance-aware representation, improving the event coverage, ordering and reproducibility.

These research questions should be answered simultaneously, instead of being separate mini-projects. A model with good performance, but unstable explanations is not usable for investigative work. A technically faithful explanation, but incomprehensible to an analyst, can help developers, but not analysts. A provenance graph with accurate artifacts, but built on the wrong hypothesis can efficiently store the wrong evidence. Therefore, the study should produce a multi-dimensional profile for each model-explainer combination, not merging all criteria into one arbitrary score. A Pareto reasoning will be more appropriate here. The study will reveal combinations offering defensible trade-offs between real-time triage, retrospective analysis and forensic reporting.

The expected contributions are threefold. Empirically, it will provide comparative information about explanation behavior on different attacks, datasets and errors of prediction. Methodologically, it will define the evaluation process combining detection, explanation and reconstruction criteria and preserving provenance of evidence. Practically, it will provide the recommendations on when explanation is necessary for alert triage, when more evidence is required and when the output should be not produced at all due to lack of stability or evidence quality. These contributions are more durable than just a claim about the best-performing algorithm on a benchmark.

The last design principle of the study is selective explanation. Not every routine benign prediction requires a costly local explanation, and not every alert needs to be fully reconstructed. The system can use calibrated risk, novelty, uncertainty and criticality of the assets to determine the level of explanation and evidence collection. Routine high-confidence events will get a concise trace; uncertain, novel or important alerts will trigger deeper explanation and provenance collection. This approach manages computational cost, but treats explanation not as an optional fancy feature. Selection criterion itself should be documented and evaluated for important cases not to be systematically omitted.

2.9 Limitations of the Existing Evidence and of the Proposed Agenda

The existing corpus suffers from certain limitations that must be accounted for in any conclusion drawn therefrom. To start with, NIDS research remains largely benchmark-oriented. Public datasets allow comparative analysis across different studies; yet, they do not reflect all kinds of encrypted traffic, organizations' diversity, current cloud infrastructure, and the adversary's adaptability. Second, performance metrics are inconsistent. Different definitions of classes, preprocessing, training, and testing procedures make it impossible to compare numbers without proper context. Third, XAI is not thoroughly assessed. Many papers provide explanations without any faithfulness, stability, and human factors evaluation. Fourth, many digital-forensics studies involve only prototypes or case studies while access to real-life investigations is restricted by privacy and legal concerns.

These limitations will not be addressed completely by the proposed methodology. Although a controlled incident provides the ground truth, it is less complicated than an operational incident. Evaluation across different datasets may be hindered by the incompatibility of feature schemas. Human subjects studies may use small sample sizes and college students as opposed to experienced investigators. Faithfulness tests require a definition of the intervention that may be imperfect itself. An evidence graph preserves provenance but does not resurrect the data that was not collected in the first place. Thus, the conclusions should take into account these limitations. Namely, they should be limited to the studied models, datasets, explainers, and scenarios.

3. Methodology

3.1 Research Design

In this research, a quantitative experimental research methodology is used to evaluate and compare supervised machine learning and explainable artificial intelligence methods for network intrusion detection and forensic-oriented analysis purposes. The methodology was structured in the form of a chain of interrelated experimental phases and not as one single classification experiment. These phases include data preprocessing, supervised intrusion-detection modelling, comparative models' evaluation, explainability analysis, cross XAI comparison, explanation stability analysis, prediction error analysis, linkage with forensic evidence and finally presentation through analyst-friendly dashboard.

The general methodological workflow is illustrated in Figure 1. The proposed design follows the sequence Detect → Explain → Compare → Test Stability → Analyze Errors → Link to Evidence → Reconstruct Where Evidentially Supported. This order of phases was selected since high prediction performance itself does not prove that the decision made using machine learning algorithms is interpretable, reproducible and evidentially sufficient for the forensic analysis.

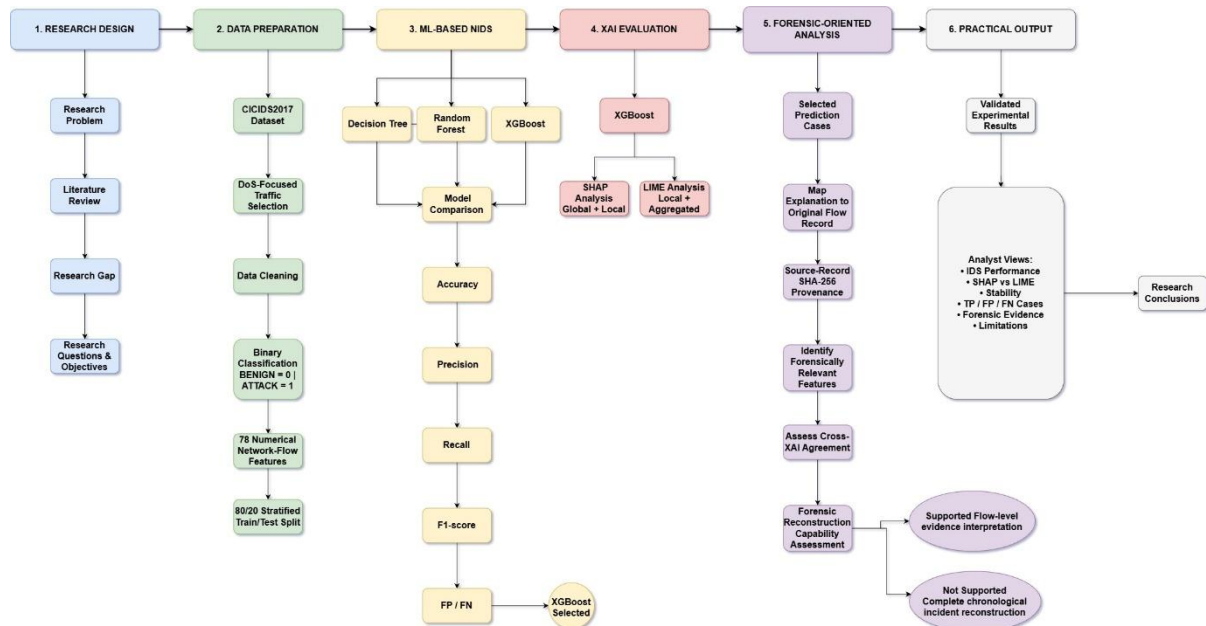


Figure 1. Methodology of research on explainable machine learning-based network intrusion detection and forensic analysis.

According to Figure 1, at first, in the present study, the research problem was defined, and the chosen network dataset was prepared. Then, three supervised classifiers were evaluated under standard experimental conditions. The best performing classifier in this experiment was further analyzed with the help of SHAP and LIME explanation approaches. Quantitative comparison of these approaches' results was followed by stability tests and comprehensive analysis of True Positive (TP), False Positive (FP), and False Negative (FN) predictions. Finally, the results of explanations' analysis were linked with source network flows' records to evaluate their forensic potential.

3.2 Dataset Selection

The experiments were performed using the CICIDS2017 dataset built by Sharafaldin et al. [3]. CICIDS2017 is a collection of both benign and malicious traffic with flow-level features suitable for supervised intrusion detection investigation. The CICIDS2017 dataset includes different attacks and has been broadly used as a benchmark to test the performance of intrusion detection techniques.

For this research, the last experiment was intentionally limited to a DoS-oriented subset of attacks. The chosen examples included benign traffic and attacks associated with DoS Hulk, DoS GoldenEye, DoS slowloris, DoS Slowhttptest, and Heartbleed. Such limitation allowed to achieve more explainable and stable results oriented on forensic analysis without declaring it as a universal representation of all attacks available in the CICIDS2017.

The source dataset included 692,703 records. After cleaning the dataset, the size of the experimental dataset became 691,406 records and 80 features. 78 numerical network flow features were used for building predictive models, while the original categorical label and the new binary one was kept as separate features.

3.3 Data Pre-processing and Target Construction

The pre-processing process was done in Python by means of Pandas and NumPy. Invalid values within the feature set (positive and negative infinities) and records containing unusable missing values were filtered out. The labels in the original CICIDS2017 dataset used for the forensics analysis were kept, while one more binary label column was added:

```
BinaryLabel= {  
0, BENIGN  
1, ATTACK  
}
```

Thus, the problem was reformulated as a supervised binary classification task, which consists in differentiation between benign and malicious network flows.

During exploratory data analysis, it was discovered that there are duplicate records in the selected records. Roughly speaking, 11.7% of all the records turned out to be duplicates. These records were not filtered automatically because identical or very similar network flows can correspond to the real network events that were simply duplicated and not mistakes. So, their filtering without additional information can potentially change the distribution of traffic. It is a limitation in the methodology and will be taken into account when interpreting the model performance results.

Textual Label and new Binary_Label columns were removed from the predictors set, and we had 78 numerical flow features left. The data were split into two sets: 80% for training and 20% for testing. Stratification technique was used during the splitting process in order to preserve class distribution in both parts of the data set. Random_state parameter equal to 42 was used.

3.4 Experimental Environment

All the experiments were conducted in Python via the Google Collab environment. The primary software libraries used involved Pandas and NumPy for data processing, Scikit-learn

for models' construction and evaluation, XGBoost for gradient-boosting classification, SHAP and LIME for explainability analysis, SciPy for statistical comparison, and Matplotlib for visualizations.

Experiments were organized in a modular way with individual notebooks representing specific methodology steps. There were different notebooks for data preprocessing, Decision Tree modeling, Random Forest modeling, XGBoost modeling, models comparison, SHAP analysis, LIME analysis, cross-XAI analysis, stability analysis, forensic analysis, and the final synthesis. Such modular structure defined the scope of experimental activities while providing opportunities to reuse the results of previous steps.

3.5 Supervised Machine Learning Models

Three tree-based supervised classifiers were chosen: Decision Tree, Random Forest, and XGBoost. While the goal was not to state that all existing intrusion detection techniques are represented by these models, it is intended to provide the progression of distinct tree-based learning methods within the framework of controlled experiments.

A Decision Tree was chosen as a relatively simple tree-based baseline. Decision trees learn the hierarchy of decisions made on the basis of labelled observations [9]. A Random Forest was chosen as an ensemble method where predictions are made based on the results provided by several trees, thus minimizing dependency on the behaviour of one particular tree [10]. XGBoost was chosen as a gradient-boosting technique since it provides an effective implementation of the tree boosting for structured machine learning tasks [8].

The following was the configuration of experiments:

Model	Experimental configuration
Decision Tree	max_depth=8, min_samples_split=10, min_samples_leaf=5, random_state=42
Random Forest	n_estimators=100, max_depth=12, min_samples_split=10, min_samples_leaf=5, random_state=42, n_jobs=-1
XGBoost	n_estimators=100, max_depth=6, learning_rate=0.1, subsample=0.8, colsample_bytree=0.8, random_state=42, eval_metric="logloss"

All the models were tested in the same binary classification task. Model selection for further XAI analysis was made on the basis of performance in the current experiment rather than the presumption about inherent superiority of certain algorithm.

3.6 Model Evaluation

The measures of interest in assessing model performance were accuracy, precision, recall, and F1-score, whereas False Positive (FP) and False Negative (FN) were tracked individually due to their direct practical meaning in intrusion detection tasks.

Accuracy determined the ratio of correct classifications overall, whereas precision determined the ratio of correctly predicted attacks in the total number of predictions. Recall determined the ratio of correctly detected attacks in the total number of attacks. F1-score was the harmonic mean of precision and recall:

$$\text{Precision} = \text{TP} / (\text{TP} + \text{FP})$$

$$\text{Recall} = \text{TP} / (\text{TP} + \text{FN})$$

$$\text{F1} = 2 \times (\text{Precision} \times \text{Recall}) / (\text{Precision} + \text{Recall})$$

The confusion matrix was also considered for finding true positives (TP), true negatives (TN), FP, and FN due to the possibility of overlooking small numbers of errors that matter in the field of application. After comparison of models, XGBoost had the best overall performance in experiments and was therefore picked up for further explainability analysis.

3.7 SHAP Explainability Analysis

SHAP was used in order to explore global and local impacts on XGBoost predictions. SHAP provides attributions for each prediction in a way that is based on Shapley values [5].

TreeExplainer was built for the trained XGBoost classifier. In order to perform global analysis, 2,000 observations were randomly drawn from the test set with `random_state = 42`. Global feature importance was found by computing mean absolute SHAP values, and bar and beeswarm summary visualizations were plotted.

Local explanations were generated for representative TP, FP, and FN observations. Waterfall plots were used in order to see which features contributed to pushing the model output toward attack prediction or away from it. SHAP values should be understood as contributions to the model output rather than probabilities.

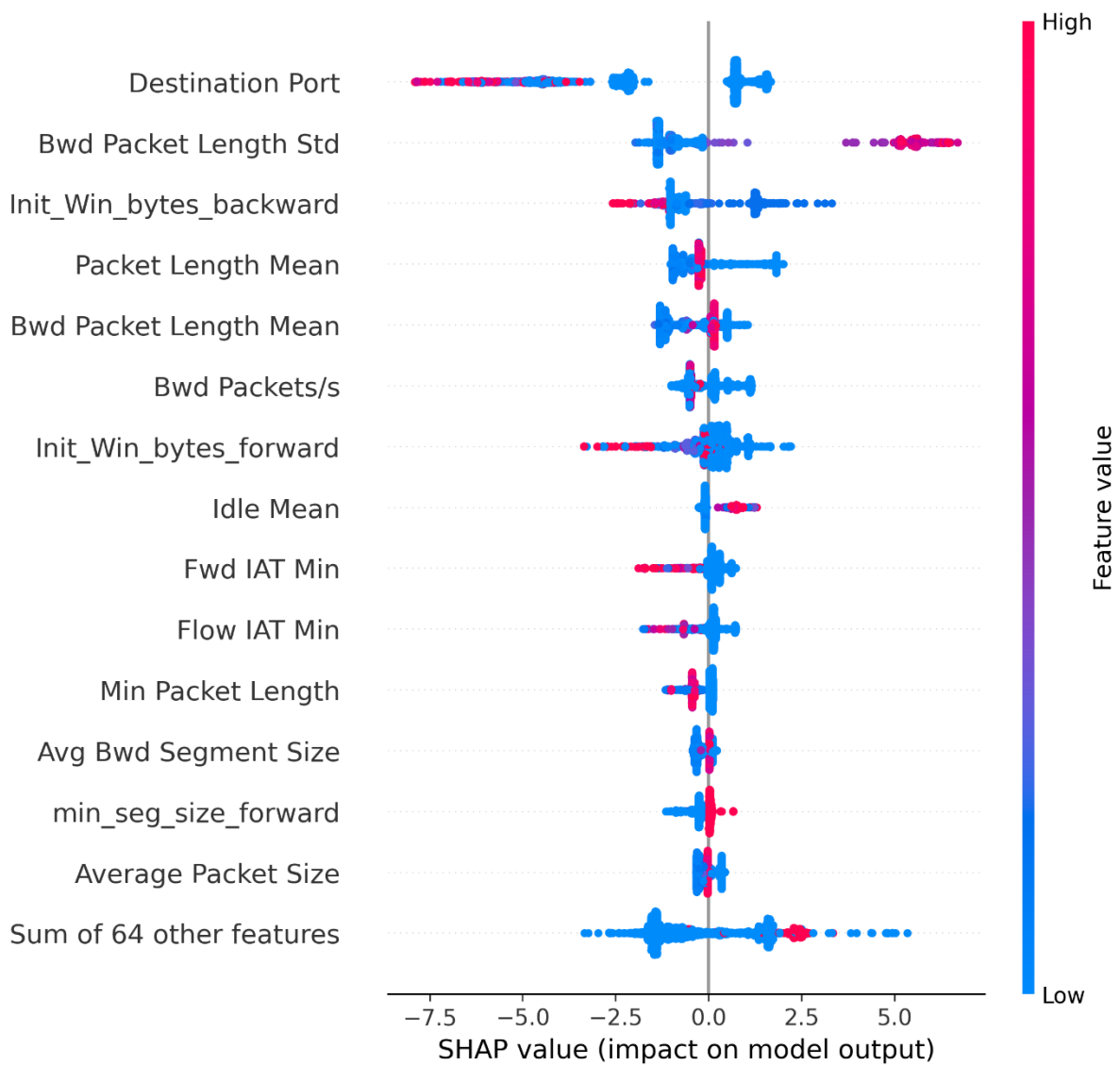


Figure 3.1: SHAP implementation

3.8 LIME Explainability Analysis

LIME was also used for explanation purposes in order to use another method for XAI to avoid bias towards only one technique. LIME approximates the behaviour of a complex classifier around individual observations with an interpretable surrogate model [4].

The configuration for Lime_Tabular_Explainer included 10,000 randomly chosen training observations (random_state = 42) that were defined as a part of the dataset containing Benign and Attack classes, allowing for continuous feature discretization.

The same TP, FP, and FN observations that were chosen for SHAP analysis were also explained by means of LIME. Local explanations were provided with the use of 15 features and 5,000 perturbation samples for each explanation.

An additional aggregated LIME analysis was performed for 50 test observations. Local explanations for all test observations were obtained using 15 features and 1,500 perturbation samples. Absolute local feature weights were aggregated and averaged in order to obtain an aggregated indicator of recurring feature importance that should be understood as aggregation of local LIME explanations rather than global ones.

3.9 SHAP-LIME Comparison

As SHAP and LIME produce different magnitudes of attribution, the raw values were not compared numerically. However, the comparison of feature rankings and their overlap was performed.

Top-k feature overlap, Jaccard similarity, and Spearman rank correlation were calculated. For two feature sets A and B, the definition of Jaccard similarity was the following:

$$J(A, B) = |A \cap B| / |A \cup B|$$

Comparison was performed for various ranking depths, such as Top-10 and Top-15 features. Thus, it was possible to find out whether independently generated explanations were identifying the same important features without assuming that SHAP values are equal to LIME coefficients.

The emphasis on evaluating the explanation method rather than presenting it corresponds to recent research in the field of NIDS that examines the stability, robustness, sparsity, and descriptive accuracy of SHAP and LIME [6].

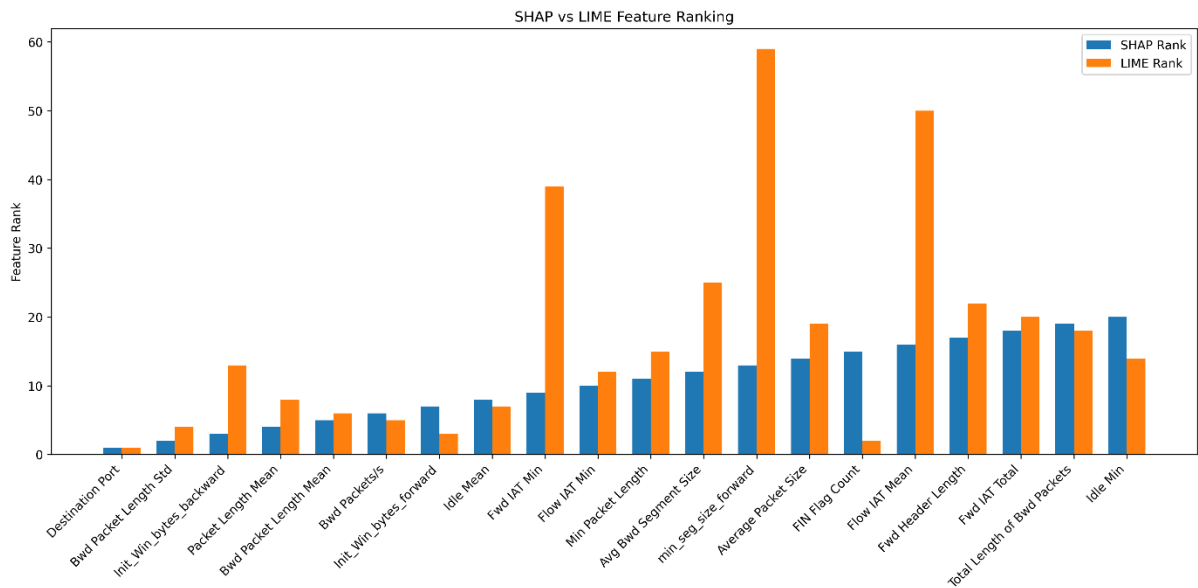


Figure 3.9 SHAP–LIME Comparison

3.10 Explanation Stability Analysis

A special stability phase assessed the reproducibility of explanation patterns in case of changing experimental settings. Stability was assessed separately from predictive accuracy and explanation faithfulness.

SHAP global stability was assessed by using different test data samples. Comparisons were performed using Top-10 Jaccard similarity, Spearman correlation, Kendall tau correlation, and cosine similarity. Additionally, bootstrap confidence intervals were calculated for prominent SHAP features.

LIME local stability was assessed separately for chosen true positives (TP), false positives (FP), and false negatives (FN) by generating new explanations under changed random settings. Analysis included feature-set overlap, rank correlation, signed cosine similarity, and direction consistency. Also, aggregated LIME stability was assessed for different test observation sets.

Finally, model stability was analyzed by retraining XGBoost with different random seeds (42, 7, 21, 84, 123). Comparison of predictive performance and SHAP explanations was performed in order to assess the necessity of having stable explanations when having stable performance of the model.

3.11 Prediction Error and Forensic-Oriented Analysis

A forensic-oriented phase conducted an in-depth analysis of one TP, one FP, and one FN prediction. Such design ensured that explainability analysis would consider not only successful predictions.

For each case, the model prediction, attack probability, SHAP explanation, and LIME explanation were associated with the corresponding retained network-flow record. The level of explanation agreement at feature level was analyzed, and the related network characteristics were classified by their forensic meaning.

Also, a provenance mechanism was developed. The cleaned analytical dataset was hashed with SHA-256, and each source record received a unique case identifier and its own SHA-256 hash. Hashing was used in order to improve analytical traceability and reproducibility without creating a formal chain of custody.

Such distinction is appropriate in light of the digital-forensics research that promotes scientific and structured investigation process and the necessity of tethering conclusions to recoverable digital evidence [7].

The accessible dataset did not contain Source IP, Destination IP, Timestamp, and Flow ID information and did not contain raw packet payloads. Therefore, flow-level evidence interpretation was possible but the chronological reconstruction of an incident was not supported. Missing host identities and timestamps were not reconstructed.

3.12 Dashboard and Research Synthesis

The experimental outputs were integrated in a Python Streamlit dashboard. The dashboard does not retrain machine-learning models; instead, it loads the preserved CSV files and figures from the experimental notebooks.

The interface presents model performance, SHAP and LIME feature importance, cross-XAI comparison, stability analysis, TP/FP/FN case exploration, forensic evidence, and identified research limitations. In other words, the dashboard serves as an analyst-facing presentation layer over validated experimental results rather than an experimental stage.

3.13 Methodological Scope

The methodology was designed to evaluate the relationship between detection performance, explanation consistency, and forensic usefulness, rather than deploying production IDS. Therefore, results are limited to the DoS-focused CICIDS2017 dataset, binary classification setup, used models, and explanation techniques.

The methodology also explicitly distinguished explanation stability and explanation correctness. A reproducible explanation does not prove the correctness of prediction, and model output explanation does not provide forensic evidence itself. Such distinctions are the basis for the upcoming Results and Discussion sections.

4. Results

4.1 Overview

This chapter describes the experimental results produced by the machine learning, explainability, stability, and forensic-based phases described in Chapter 3. The results have been arranged in accordance with the process of the research methodology, namely, supervised models' comparison, SHAP and LIME explainability, cross-method agreement, explanation stability, prediction error, and forensic evidence linkages.

4.2 Supervised Intrusion-Detection Performance

Comparative evaluation of classifiers Decision Tree, Random Forest, and XGBoost has been carried out in case of a binary task with emphasis on DoS. The following evaluation criteria were used: accuracy, precision, recall, F1-score, false positives, and false negatives. According to the experiment, XGBoost was the best classifier, having the maximum scores for all four major criteria and minimum numbers of False Positives and False Negatives.

The Decision Tree classifier has been taken as the simplest baseline one and provided the following results: accuracy - 0.994041, precision - 0.991504, recall - 0.992134, and F1-score - 0.991819. Moreover, this classifier made 428 False Positives and 396 False Negatives. Although these results show good classifying capabilities, the remaining error numbers turned out to be higher than the corresponding ones for XGBoost.

XGBoost provided the following scores: accuracy - 0.999747, precision - 0.999444, recall - 0.999861, and F1-score - 0.999652. It made 28 False Positives and 7 False Negatives. The training took about 24.46 seconds, while predictions took about 0.34 seconds.

Model	Accuracy	Precision	Recall	F1-score	False Positives	False Negatives
Decision Tree	0.994041	0.991504	0.992134	0.991819	428	396

Model	Accuracy	Precision	Recall	F1-score	False Positives	False Negatives
XGBoost	0.999747	0.999444	0.999861	0.999652	28	7

Table 1. The main performance results for baselines

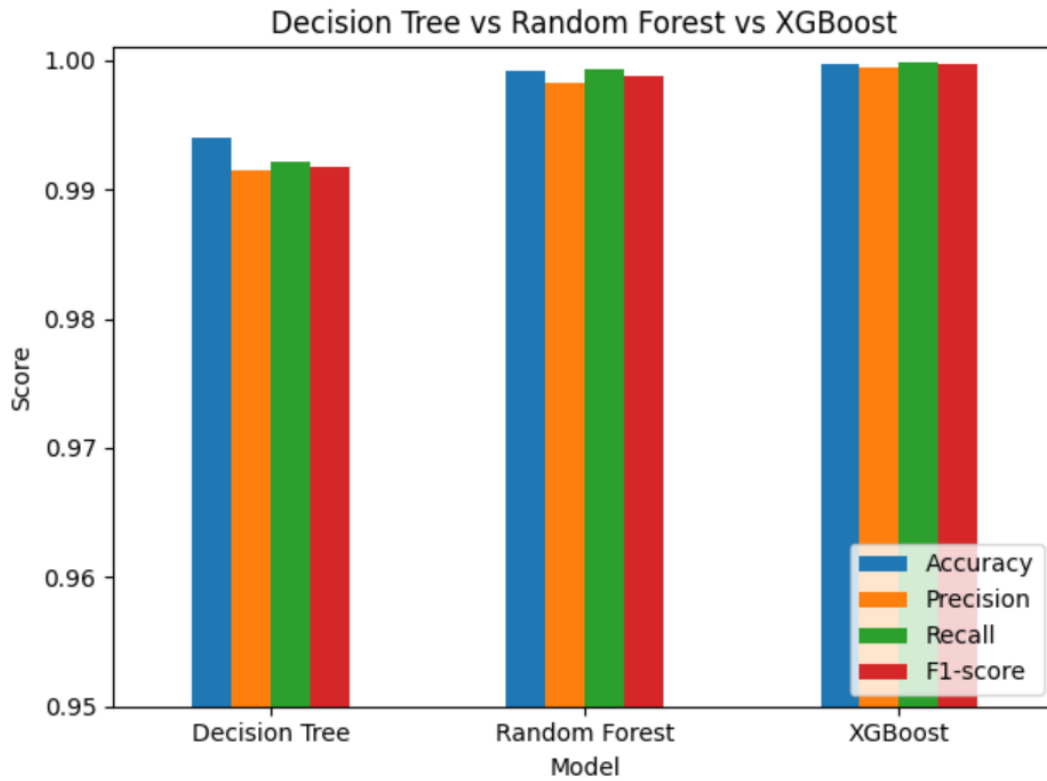


Figure 2: Decision Tree vs Random Forest vs XGBoost comparison

This overall analysis thus supported the choice of XGBoost for the following XAI experiments. It is worth noting that this decision should be considered applicable to the current dataset and experimental setting only, and cannot be considered proof of the superiority of XGBoost over other intrusion detection algorithms.

4.3 SHAP Explanation Results

Global analysis of the SHAP values was performed using a test sample containing 2,000 observations. The analysis showed that the most important attribute was Destination Port with the mean absolute SHAP value of approximately 2.419. Other features included Bwd Packet Length Std with 2.111 and Init_Win_bytes_backward with 1.036.

Also, the most important attributes were Packet Length Mean, Bwd Packet Length Mean, Bwd Packets/s, Init_Win_bytes_forward, Idle Mean, Fwd IAT Min, and Flow IAT Min. Overall, these results show that the classification made using the XGBoost algorithm was based on the usage of information about ports, packet lengths, characteristics of backward flows, window values, and time-related features.

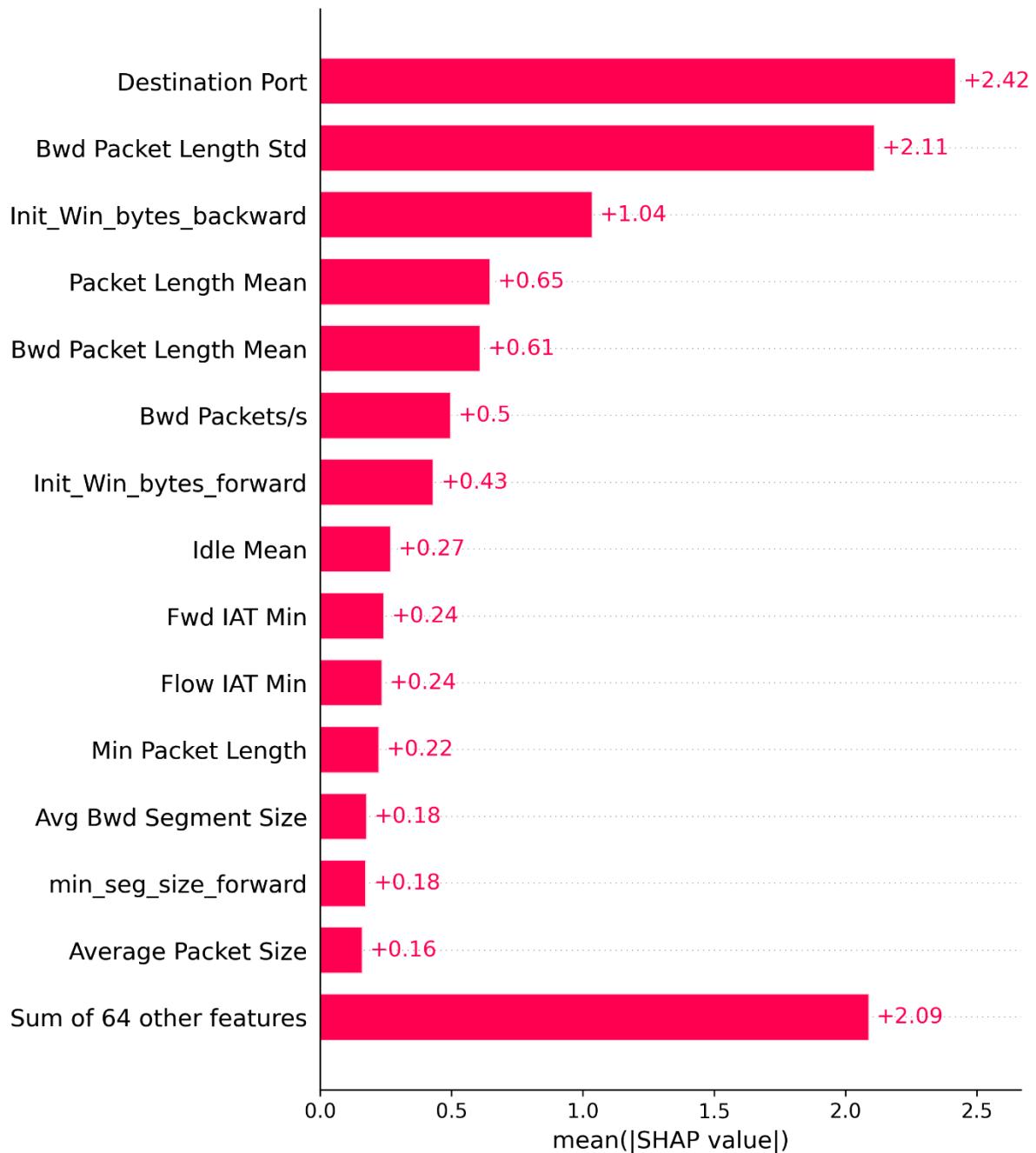


Figure 3: SHAP Global Feature Importance

In addition, the SHAP beeswarm plot shows that the feature contribution was dependent not only on its magnitude but also on the individual characteristic of each observation.

Positive contributions of SHAP increased the value of the model output and pushed it towards the Attack class, while negative contributions decreased it and pushed it towards the Benign class.

Local SHAP analysis was performed for the selected TP, FP, and FN observations. The selected True Positive case was correctly classified as an attack with the attack probability of approximately 99.9965%. The selected False Positive case is an observation with benign traffic misclassified as an attack with the attack probability of approximately 99.7936%. The selected False Negative case is an observation with the attack misclassified as benign with the attack probability of approximately 39.56%, which was below the threshold for binary classification.

Thus, the selected local cases became the basis for the following analysis focused on errors and forensics.

4.4 LIME Explanation Results

As was mentioned, LIME analysis was performed for the selected TP, FP, and FN observations, which were used for the SHAP experiment in order to exclude possible differences between the explanation methods caused by model prediction.

Aggregated analysis of 50 local LIME explanations showed that the most important feature was Destination Port, similarly to the results of the global SHAP analysis. The other top features were FIN Flag Count, Init_Win_bytes_forward, Bwd Packet Length Std, Bwd Packets/s, Bwd Packet Length Mean, Idle Mean, and Packet Length Mean.

Thus, the results of LIME analysis had some common features with SHAP, namely concerning such features as destination port, backward packets behaviour, packets sizes, and network flow timing. In contrast to SHAP, LIME assigned much more importance to such features as FIN Flag Count, Fwd PSH Flags, and SYN Flag Count.

Since LIME explanation is represented by coefficients of the linear model, while SHAP values have another meaning, the numerical values of these two methods cannot be compared.

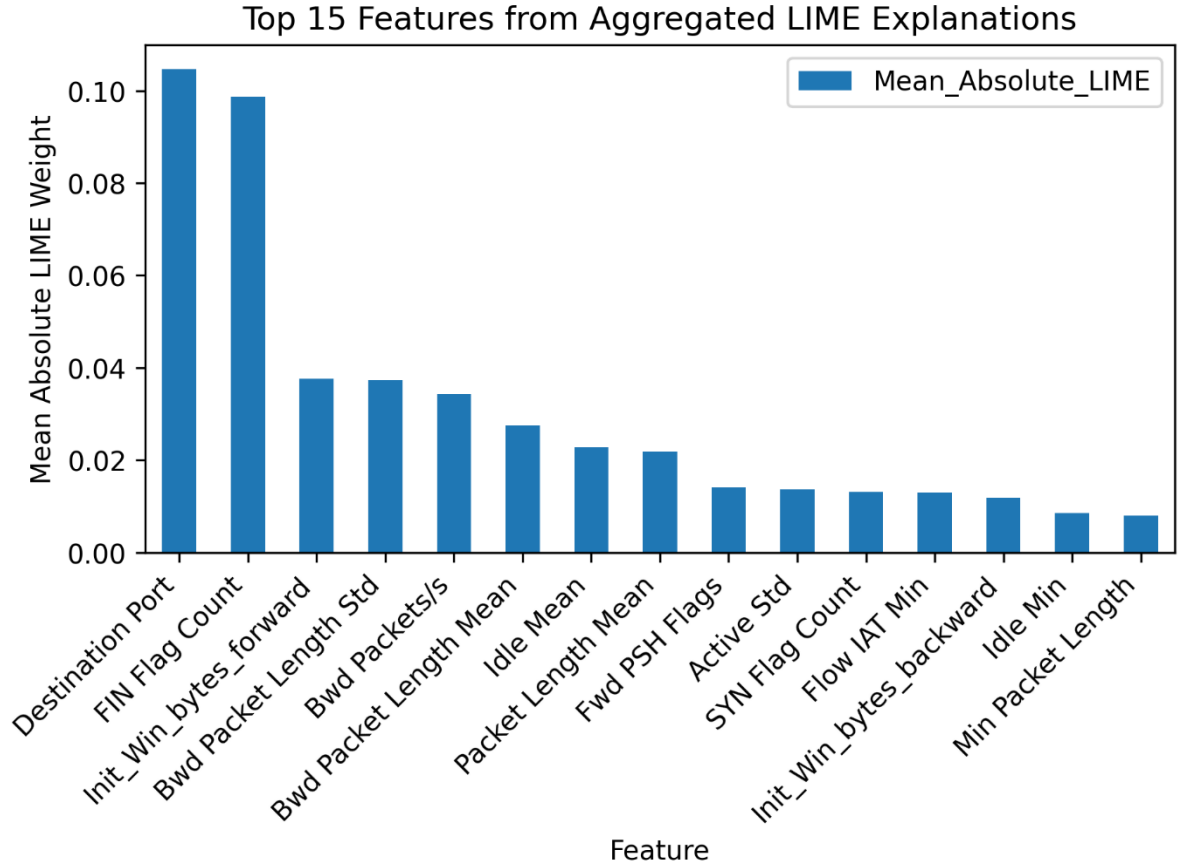


Figure 4: Aggregated LIME Feature Importance

4.5 SHAP-LIME Consistency

A numerical comparison was performed to check whether SHAP and LIME provide consistent feature rankings.

The two methods share seven of their Top-10 features, resulting in a Top-10 Jaccard similarity of about 0.5385. In the case of Top-15 features, there are eleven common features, which corresponds to a Jaccard similarity of about 0.5789. The Spearman rank correlation is about 0.6458, the reported p-value being about 1.72×10^{-10} .

Measure	Result
Common Top-10 features	7
Top-10 Jaccard similarity	0.5385
Common Top-15 features	11
Top-15 Jaccard similarity	0.5789
Spearman rank correlation	0.6458
Spearman p-value	1.72×10^{-10}

Table 2. SHAP-LIME consistency

Destination Port turned out to be the most consistent feature, ranking first according to both approaches. Features with high consistence are: Bwd Packet Length Mean, Bwd Packets/s, Idle Mean, and Flow IAT Min.

There are several features whose rankings differ considerably. For example, FIN Flag Count was ranked as second feature by LIME and as fifteenth by SHAP. Also, Fwd IAT Min is ranked as ninth by SHAP and is much lower in ranking by LIME. Thus, the two explainers identify an overlapping set of influential features but do not give the same explanation.

4.6 Explanation Stability

The stability study was aimed at checking whether the explanations are consistent with respect to variations in sampling tests, LIME randomness, and XGBoost retraining.

In particular, SHAP shows a high global stability under various test sampling conditions. Its mean Top-10 Jaccard similarity is 1.0000, the mean Spearman correlation is 0.9986, the mean Kendall correlation is 0.9814, and the mean cosine similarity is 0.9996. The first seven globally ranked features always stay the same in any sample analyzed.

Measure	Mean result
Top-10 Jaccard	1.0000
Spearman correlation	0.9986
Kendall correlation	0.9814

Measure	Mean result
Cosine similarity	0.9996

Table 3. SHAP sampling stability

The LIME method is more variable in particular cases. For the True Positive case, its mean Top-10 Jaccard similarity is 0.5824, as compared to 0.7906 and 0.8404 in the False Positive and False Negative cases correspondingly. The signed cosine similarities are 0.9236, 0.9496, and 0.8748 correspondingly.

The aggregate LIME analysis was more stable than some local explanations, providing a mean Top-10 Jaccard similarity of 0.7303, a mean Spearman correlation of 0.7829, a mean Kendall correlation of 0.6273, and a mean cosine similarity of 0.9925.

In the case of XGBoost retraining, five random seeds were used. As a result, the mean predictive accuracy is still about 0.999771 with a standard deviation of only 0.000017. However, the explanations generated by SHAP for the retrained models have a lower mean Top-10 Jaccard similarity of 0.7727, although the cosine similarity is 0.9935.

It means that stable predictive accuracy does not necessarily correlate with consistent explanations.

4.7 Error-focused XAI analysis

Analysis of true positives (TP), false positives (FP), and false negatives (FN) allows checking explanation behaviour outside the aggregate classification performance.

The TP case corresponds to a DoS Hulk flow that is correctly classified as an attack with an attack probability of 0.9999653. The FP case corresponds to a BENIGN flow that is incorrectly classified as an attack with a probability of 0.9979361. The FN case corresponds to a DoS GoldenEye flow that is incorrectly classified as benign with an attack probability of 0.3956447.

Case	Original class	Prediction	Attack probability
True Positive	DoS Hulk	Attack	0.9999653
False Positive	BENIGN	Attack	0.9979361
False Negative	DoS GoldenEye Benign		0.3956447

Table 4. Selected prediction cases

The FP case is especially interesting since the classifier shows high confidence for an incorrect attack classification. Yet, both SHAP and LIME generate consistent explanations for this decision. This empirically proves that an explanation can point out the features which cause a model decision even when this decision contradicts the original class.

Local cross-method comparison showed different degrees of consistence among cases. The TP explanation includes seven common Top-10 features of SHAP and LIME, which provides a Jaccard similarity of 0.5385. The FP and FN explanations include five common Top-10 features correspondingly, resulting in Jaccard similarities of 0.3333.

4.8 Forensic-oriented Evidence Results

Finally, the source network-flow records were used to connect each selected prediction to its evidence. Individual records and the cleaned analytical dataset were SHA-256 hashed for analysis provenance and reproducibility.

The SHA-256 hash of the retained dataset is:

3d3d9611d2cf7c3da661d052a88aea2f6e50eb45d6677e94e0215184041ccaf1

Thus, the selected TP, FP, and FN records can be traced from the model prediction and XAI explanation to a specific retained source flow.

The available evidence allows examining such features as Destination Port, packet lengths, packet rates, idle-time characteristics, flow timing, and TCP flags. However, the final machine-learning CSV does not include Source IP, Destination IP, Timestamp, and Flow ID fields, as well as raw packet payloads.

Thus, the experiment allows performing flow-level evidence interpretation and analysis traceability but does not allow building a complete chronological timeline of an incident.

	Forensic Requirement	Available
0	Original analysed flow record retained	Yes
1	Source record hash generated	Yes
2	Original attack label available	Yes
3	Model prediction reproducible	Yes
4	Prediction probability available	Yes
5	SHAP explanation available	Yes
6	LIME explanation available	Yes
7	Cross-method explanation comparison available	Yes
8	Source IP available	No
9	Destination IP available	No
10	Timestamp available	No
11	Flow ID available	No
12	Chronological incident timeline possible	No
13	Raw packet payload available	No - not present in ML CSV

Figure 5: Forensic capability assessment from Notebook 11.

This separation represents an important border in the experiment findings: Explainable AI (XAI) can identify and prioritize influential characteristics in the source network-flow record, but the available evidence is not enough to reconstruct the complete sequence, identity, and context of an intrusion incident.

4.9 Summary of Results

Through the entire experimental pipeline, XGBoost demonstrated the best detection performance under the current benchmark conditions, and therefore, it was chosen for explainability analysis. Both SHAP and LIME identified several common influential network-flow features with Destination Port ranked first by both methods, but not identical rankings.

The stability tests showed high consistency of SHAP global explanations under the variation in test sampling. LIME is more variable, especially in the case of individual local explanations. Also, retraining tests showed that high stable predictive accuracy and variations in explanation rankings can coexist.

The analysis of true positives (TP), false positives (FP), and false negatives (FN) extended the evaluation outside the correctly classified traffic. In particular, high-confidence false positives prove that a detailed explanation can provide the internal reasoning behind an incorrect prediction. Finally, the connection of explanation outputs to retained source records allows using XAI to perform evidence-oriented investigation, but at the same time, it reveals the limitations caused by missing temporal, host identity, and packet-level information.

5. Discussion and Research Contribution

5.1 Overview

This research studied the possibility of enhancing high-performance machine learning-based network intrusion detection through explainability, explanation evaluation, and forensic-oriented evidence analysis. The findings demonstrate that these goals should not be considered a single problem. A model may have excellent predictive performance and still produce mistakes; an explanation method may produce a reasonable explanation of a prediction and differ from another explanation method; and an interpretable prediction is not necessarily forensic evidence.

Therefore, the key conclusion of this research is that detection accuracy, explanation reliability, and forensic evidential value are connected but not identical concepts. As a consequence, the research experimental design went beyond the usual classification evaluation by comparing models, analysing SHAP and LIME explanations, measuring cross-method agreement, testing explanation stability, studying individual prediction mistakes, and evaluating connection with retained source-flow records.

5.2 Discussion of Intrusion-Detection Performance

The first research question focused on how effectively Decision Tree, Random Forest, and XGBoost discriminated the benign from attack traffic in the current experimental conditions. All three algorithms performed well, but XGBoost had the best results of all. XGBoost reached an accuracy of 0.999747, precision of 0.999444, recall of 0.999861, F1-score of 0.999652, with 28 false positives and 7 false negatives.

The above results provided the basis for the choice of XGBoost as a target algorithm for explanation experiments. Nevertheless, the high scores should be treated cautiously. The experiment used a DoS-oriented subset of CICIDS2017 and 80/20 stratified random split and therefore demonstrated good performance of models in the specific benchmark configuration and not the equivalency of performance on a real-life network.

It is especially important in light of Engelen, Rimmer, and Joosen findings about problems in CICIDS2017 associated with traffic generation, flow generation, feature extraction, and labelling. These researchers found that the properties of the benchmark can greatly affect the evaluation of machine learning algorithms. Thus, the high scores obtained in this research should be considered benchmark-specific results and not a proof of near-perfect intrusion detection on a real-life network.

Another important methodological consideration is the use of random flow-level partitioning. There is likely some related traffic in both train and test sets, meaning that the classification task is easier than if it would be performed against different temporal windows or networks. As a result, the importance of model comparison consists in providing the controlled basis for selecting a classifier for the XAI experiments and not the proof of universal superiority of XGBoost.

5.3 Discussion of SHAP and LIME Features

The second research question focused on network-flow features that affected XGBoost predictions. Both SHAP and the combined LIME analysis produced Destination Port as the highest-ranked feature. The additional features found by both methods include Bwd Packet Length Std, Bwd Packet Length Mean, Bwd Packets/s, packet-length features, window features, and timing features.

The consistent appearance of the same set of features among two independently generated explanation methods is valuable since the feature ranking can be treated as the additional source of information besides the usual one-way feature importance ranking. In addition, Gaspar, Silva, and Silva similarly used SHAP and LIME for intrusion detection and stated that explanation methods can provide cybersecurity analysts with additional information about the model behaviour [11]. At the same time, the work of Gaspar, Silva, and Silva shows that comparing explanation methods is also valuable since there is no need to assume the presence of one explanation as sufficient for interpretability.

At the same time, the feature importance should not be considered as the causal effect. For example, the prominent ranking of the Destination Port is only an indicator that the trained XGBoost classifier relies heavily on this feature in the selected dataset. It does not show that the particular destination port alone proves the existence of the attack.

Such a consideration is important for forensic use since the explanation of feature importance provides information about how the classifier made the prediction. It does not prove the existence of any security event based on this explanation.

5.4 Cross-XAI Agreement and Explanation Stability

The third research question tested whether SHAP and LIME explanations are consistent and stable. The two methods have 7 top-10 features in common, showing top-10 Jaccard similarity of 0.5385. There are 11 common features among the top-15 lists, showing top-15 Jaccard similarity of 0.5789. Spearman rank correlation of 0.6458 indicates the positive relationship between the features' ranking by the two methods.

The result shows that the two methods share the common set of important features and still provide method-specific ranking. Such a finding is important since the agreement of the two independent methods cannot be expected a priori. SHAP and LIME methods are based on the different mathematics and have different meanings of their raw feature attributions. Therefore, in this study, the comparison of the features' ranking and feature overlap, and not the feature attributions themselves, was performed.

The inconsistencies of SHAP and LIME also demonstrate the reason to use more than one explanation method since limiting the analysis to a single method can produce some analytical biases. For example, FIN Flag Count feature got much higher ranking from LIME than from SHAP, whereas certain timing features were ranked higher by SHAP.

The experiments on explanation stability added the secondary layer of analysis. The global SHAP explanation method demonstrated high consistency in different test cases with the average top-10 Jaccard similarity of 1.0000, Spearman correlation of 0.9986, Kendall correlation of 0.9814, and cosine similarity of 0.9996. On the contrary, LIME showed larger inconsistency at the level of individual predictions. For example, the TP case has top-10 Jaccard similarity of 0.5824, whereas the FP and FN cases got 0.7906 and 0.8404 similarities, respectively.

The result aligns with the existing problems associated with the evaluation of explanation methods themselves. Ghorbani, Abid, and Zou showed that the explanations of machine learning models can significantly change even if the prediction remains the same [13]. At the same time, research described the differences in the stability characteristics of LIME and SHAP and discussed the importance of LIME perturbation for explaining the same model twice [15].

However, the appropriate conclusion to make from the findings of the present study is not the generalization that "SHAP is always more stable than LIME." The correct conclusion is more modest: under the conditions of this study, the global SHAP explanations are more stable than LIME explanations.

Of special interest is the result of retraining XGBoost with different random seeds. Predictive performance remained very stable with mean around 0.999771 and standard deviation of 0.000017. However, the SHAP top-10 Jaccard similarity fell to about 0.7727. This result

shows that the stability of the predictive performance does not automatically lead to the explanation stability; the model can perform very similarly, and the relative importance of the individual features can vary.

5.5 Error Analysis and the Limits of Explanation

The analysis of TP, FP, and FN cases produced one of the key findings of this research. Conventional machine learning evaluation usually provides some aggregate metrics summarizing performance. At the same time, in the context of cybersecurity, individual mistakes are important since the false negative case describes the traffic that has been classified as benign and therefore can be left without necessary response and the false positive case means the waste of resources and possible misleading of analysts.

The selected false positive case demonstrates the above problem very well. The original label was BENIGN, but the XGBoost predicted it as Attack with attack probability of about 0.997936 (approximately 99.79%). Both SHAP and LIME provided the structured explanations of why the XGBoost classified this record as the Attack.

The result highlights the important difference: the explanation can describe correctly why the classifier made the prediction, but this does not automatically mean that the prediction is correct. The correct explanation does not constitute ground truth.

The above consideration is confirmed by the research findings indicating that the post hoc explanations should not be immediately considered as reliable representation of the model's correctness. Slack et al. found out that LIME and SHAP can be manipulated under adversarial conditions to produce seemingly acceptable explanations of the classifier behaviour [14]. Of course, the findings pertain to a different field, but they show that the explanation, even if looks convincing, should be carefully evaluated rather than accepted without hesitation.

In the context of intrusion detection, the above result strengthens the argument for keeping the role of the analyst in the decision-making process. Even though SHAP or LIME can point to the features worth investigating, the analyst should verify the alert against the originating network evidence and other contextual information.

5.6 XAI and Forensic-Oriented Analysis

The fourth research question tested whether XAI could help in obtaining forensic knowledge and reconstruction of incidents. The results of the study produce the partially positive answer: XAI assisted in evidence-oriented, flow-level investigation, but the dataset used was insufficient for full chronological reconstruction of the incident.

For every case selected, the prediction and explanation produced by the model could be connected to the retained source network-flow record. The SHA-256 hash was calculated for both the analytical dataset and selected records, allowing showing that the record being analyzed was kept during the entire analysis procedure. Such an approach improves the reproducibility and traceability of the analysis.

At the same time, hashing alone is not sufficient for the creation of a full forensic chain of custody. Moreover, the final machine-learning CSV did not contain the Source IP, Destination IP, Timestamp, and Flow ID attributes, as well as the content of packets. Such a lack of data makes impossible the identification of which endpoints communicated, the time of events, and how the individual flows should be arranged chronologically.

The above findings are consistent with the emerging work in the domain of XAI and digital forensics. Research highlighted the potential of explainability techniques to assist in interpreting AI-assisted forensic analysis, recognizing the need for trust and human intervention [17]. More recently, Khalid, Iqbal, and Fung offered an XAI-oriented digital-forensics framework where the AI-based predictions should remain interpretable and be incorporated into the forensic process [16].

Thus, the present study positioned the XAI as an investigative tool linking model prediction to source evidence. It can be used to direct the analyst's attention to influential flow attributes and to choose records for analysis. It is unable to prove that the attack happened or to reconstruct missing information about the events.

5.7 Research Contribution

The algorithms used in this study (Decision Trees, Random Forest, XGBoost, SHAP, LIME, CICIDS2017) are well-established tools and are not presented here as novel approaches. Instead, the contribution of this research is in the combination and evaluation of those technologies across multiple levels of trust and forensic utility.

First of all, the research offers a controlled comparison of three supervised tree-based classifiers prior to the explanation experiment. This helps to apply the explainable AI (XAI) approach to the classifier that is the best performer according to empirical results.

Secondly, the research goes beyond producing SHAP and LIME plots by comparing them quantitatively through feature overlap, Jaccard similarity, and rank correlation. This allows checking whether two different explanation methods identify the common influential network features.

Third, the explanation stability is tested explicitly. This is important since the mere fact of the explanation existence does not guarantee that it is reproducible under different sampling

conditions, perturbations, and model training. The results also separate the model stability from the explanation stability.

Fourth, the XAI is applied not only to the successful predictions, but also to the wrong ones. This means that the research considers true positives (TP), false positives (FP), and false negatives (FN) and not just the successful alerts.

Fifth, the selected explanations are connected to the retained source records and provenance information. The approach adds the evidence-oriented component in which the model attributions are used as a guide to source evidence.

Finally, the research results are represented in an analyst-friendly Streamlit dashboard. While this part of the work does not add any experimental evidence, but provides the practical layer allowing to analyse the predictive performance, SHAP/LIME explanations, their stability, prediction mistakes, and forensic potential within the unified workflow.

The above-mentioned contribution can be described as:

Detect → Explain → Compare → Test Stability → Analyze Errors → Link to Evidence → Reconstruct Where Evidentially Supported

This sequence reflects the transition from the simple question "Did the model classify the network flow correctly?" to the broader one: "Why did it make that decision, do different explainers agree, is the explanation stable, what is the behavior in case of mistake, and what source evidence supports the interpretation?"

5.8 Research Questions Revisited

The above findings give the following answers to the research questions:

Research Question	Overall Finding
RQ1: How effectively do DT, RF, and XGBoost classify the traffic?	All approaches performed well; XGBoost performed the best under the current experimental conditions.

Research Question	Overall Finding
RQ2: Which features influence XGBoost predictions?	SHAP and LIME agreed on Destination Port and several packet-related, timing, and backward-flow features.
RQ3: Are SHAP and LIME explanations consistent and stable?	They agreed to a meaningful but limited extent. SHAP showed high stability under the global sampling analysis, LIME showed greater local instability.
RQ4: Can XAI assist in forensic incident reconstruction?	To a certain extent. XAI helped to interpret and connect to the source flow records, but the lack of timestamps, endpoint information, Flow IDs, and packet content prevented chronological reconstruction.

6. Limitations and Future Work

6.1 Limitations

Despite the apparent success and promise of applying explainable machine learning to network intrusion detection and forensic-oriented analyses as evidenced by the experimental results, several limitations are worth taking into account while analyzing those results.

First, the limitation of the dataset scope should be noted. A DoS-oriented subset of CICIDS2017 was selected for conducting the research; therefore, the final classification task is to distinguish between Benign and Attack network traffic. Thus, one should not interpret the findings as a proof of similar performance of other types of attacks, e.g., brute force attack, botnet, infiltration or web attack. Therefore, the extremely high accuracy of the XGBoost classifier is dataset-specific.

Second, the train-test splitting method poses a limitation. Stratified random splitting was used in order to keep the class balance and provide for consistent evaluation of the three algorithms. However, similar network flows may appear in both train and test partitions, which makes the classification task easier than evaluation with temporally independent network traffic and/or with another dataset. The performance should thus be understood as a within-dataset benchmark.

Furthermore, the preprocessing included detecting duplicate records, which accounted for about 11.7% of the selected records. The duplicates were kept due to the possibility that

repeated features of a network flow can be an actual repetition of traffic and not a mistake. It is possible that the existence of the duplicates affected the learned distribution and caused similarities between train and test samples. A separate sensitivity analysis with retained and removed duplicates is out of the scope of the current research.

The explainability analysis also has certain limitations. Global feature importance in terms of SHAP was computed with 2,000 test sample observations; at the same time, the LIME-based aggregated feature importance was constructed based on 50 LIME local explanations. The two types of explanations are thus derived from different samples and have different analytical scope. Even though it is possible to make use of ranks and feature importances overlap to compare the two, one should not interpret the finding as a comparison of the global explanation power of the two techniques.

LIME itself is additionally characterized by stochastic nature of the perturbation process and the construction of local surrogates. The experiments performed with LIME revealed that local explanations can be different in repeated runs, similarly to SHAP explanation, which can change depending on random seed values. One should thus not assume that explanation outputs are invariant properties of the dataset and predictions made on it.

One more limitation of the research is the lack of focus on explanation consistency and faithfulness in terms of causality. Concordance of SHAP and LIME explanations implies agreement on attributions but does not confirm faithfulness of the explanations in terms of causation. Additionally, constancy of the explanations does not prove the correctness of the prediction. As an example, a False Positive with high confidence score can serve.

The limitation of the forensic part of the research is the availability of information in the pre-processed dataset CICIDS2017. Namely, there was no Source IP, Destination IP, Timestamp and Flow ID information, as well as raw packet payloads, in the final analytical dataset. The lack of the aforementioned information complicates the chronological reconstruction of the events and activity correlation across particular endpoints. The research thus provides a basis for flow-level interpretation and does not allow for a comprehensive digital-forensics investigation.

Finally, the detailed forensic analysis included only one True Positive, False Positive and False Negative cases. The selected examples are thus not a comprehensive list of prediction outcomes. The Streamlit dashboard is therefore an offline analytical interface for the saved experiment results and not a production environment for network intrusions detection.

6.2 Future Work

Firstly, the study should aim to extend its scope beyond the small set of DoS attacks in CICIDS2017. New categories of CICIDS2017 attacks could be considered, and the binary classification problem could be expanded to multi-class problem. Such approach would make it possible to explore whether the observed importance of features, explanation agreement, and stability are preserved for other types of attacks.

Secondly, more rigorous generalization testing should be conducted. Instead of using only a randomly split flow-level dataset, future research could consider temporal, scenario-based, or even attack-group based splits. Testing on other intrusion detection datasets such as UNSW-NB15 and CSE-CIC-IDS2018 would give insights whether the study results depend on CICIDS2017.

More preprocessing experiments could be done to evaluate whether duplication of records and high correlation between flows affect the results of the study. Comparison of the performance and explanations of a model before and after intentional deletion of duplicates would provide evidence regarding whether repetitive traffic affects the results.

Future XAI evaluation could be expanded beyond agreement and stability. Systematic evaluation of explanation faithfulness, robustness, sparsity, and computational complexity would be beneficial. Incorporation of new methods of explanation where theoretically justified would give opportunity to examine whether various techniques of attribution give similar results. The research would still have to separate explanation plausibility and prediction correctness.

The forensic component of the study could benefit from more rich source evidence. In future studies and controlled network experiments, timestamp, source/destination IPs, ports, Flow IDs, and where feasible and legal – packet captures – should be recorded. The network flows data could be correlated with firewall logs, authentication logs, DNS logs, and host telemetry information. In this way, the XAI selected alerts could be traced back to the actual sequence of events instead of individual flows.

Additionally, more true positives, true negatives, false positives, and false negatives should be investigated to find out whether the forensic and explanation results observed in the selected samples are valid for the overall model behavior. Special attention should be paid to high-confidence incorrect predictions because they are important operationally and show where the explanations lead to misplaced confidence.

Lastly, the current offline dashboard could be developed into a near-real-time prototype. Such prototype could include network flow acquisition, automatic preprocessing, prediction, explanation, and evidence retrieval. The system would require further considerations of

computational latency, cybersecurity requirements, evidentiary preservation, and user experience prior to the operational use.

6.3 Chapter Summary

This chapter provides evidence for potential benefits of XAI in enhancing the transparency of intrusion-detection decisions and facilitating flow-level forensic analysis. Still, the results are limited by the selected dataset, experiment design, and evidence. In particular, predictive accuracy, stability of explanations, and forensic reliability should not be considered equivalent notions.

Future research could benefit from more extensive attack coverage, more rigorous validation, richer forensic evidence, comprehensive evaluation of XAI techniques, and detailed analysis of errors. Such efforts would facilitate the determination whether the results of explainable intrusion-detection systems are reliable enough for practical cybersecurity and forensic investigation.

7. Conclusion

This research examines the potential of explainable machine learning in the context of network intrusion detection and determines how well the model explanations may assist forensic incident analysis. The experimental design encompasses supervised intrusion detection, explainability, comparison of explanations, stability tests, error analysis, and linking evidence into a coherent workflow.

The first research question compares Decision Tree, Random Forest, and XGBoost classifiers in the same experimental setting. All models demonstrate good classification performance on the CICIDS2017 data set, which focuses on DoS attack scenarios. In particular, XGBoost is the best performer in this experiment, achieving an accuracy of about 99.97% and producing 28 False Positives and 7 False Negatives. The latter justify the choice of the algorithm for the subsequent analysis, despite the data- and design-specificity of the result.

The second research question aims to identify network flow features impacting XGBoost classifier's predictions. Both SHAP and LIME reveal a number of overlapping important features, including Destination Port and attributes associated with packet size, backward flow activity, and network time. The findings imply that the classifier's predictions could be analyzed beyond the binary Benign/Attack classification.

The third research question is concerned with the consistency and stability of explanations. SHAP and LIME share seven features among their Top-10 lists, meaning there is substantial yet imperfect agreement between them. The stability tests show that the SHAP global explanations are very consistent in the samples used, while LIME is more flexible locally. Moreover, the re-training experiment shows that stable prediction performance does not lead to identical explanations.

The fourth research question is aimed at the forensic value of explainable artificial intelligence (XAI). Explanations of selected True Positives, False Positives, and False Negatives are successfully connected to preserved network flow records and provenance data. However, the lack of Source IP, Destination IP, Timestamp, Flow ID, and packet-level data makes it impossible to conduct full chronological analysis of incidents. As a result, XAI allows for forensic interpretation but not reconstruction of network flows.

The main contribution of this research is not in the application of XGBoost, SHAP, and LIME independently but in their comprehensive evaluation concerning detection performance, comparison of explanations, stability of explanations, model errors, and connection to evidence. Notably, there is a high confidence False Positive in this research showing that a plausible explanation may exist alongside the erroneous prediction. Therefore, explanations cannot be taken as proof of the model correctness or as independent forensic evidence.

Overall, this research proves that accurate detection of attacks does not suffice for the forensic-oriented intrusion detection. More accountable analytical process requires prediction explanation, evaluation of the explanation, error investigation, and consistent drawing of conclusions based on existing evidence.

REFERENCES

- [1] K. Shaukat et al., "A survey on machine learning techniques for cyber security in the last decade," *IEEE Access*, vol. 8, pp. 222310–222354, 2020, doi: 10.1109/ACCESS.2020.3041951.
- [2] H. Hindy et al., "A taxonomy of network threats and the effect of current datasets on intrusion detection systems," *IEEE Access*, vol. 8, pp. 104650–104675, 2020, doi: 10.1109/ACCESS.2020.3000179.
- [3] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *Proc. 4th Int. Conf. Information Systems Security and Privacy (ICISSP)*, 2018, pp. 108–116, doi: 10.5220/0006639801080116.
- [4] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why should I trust you?: Explaining the predictions of any classifier," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, 2016, pp. 1135–1144, doi: 10.1145/2939672.2939778.
- [5] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems*, vol. 30, 2017, pp. 4765–4774.
- [6] O. Arreche, T. R. Guntur, J. W. Roberts, and M. Abdallah, "E-XAI: Evaluating black-box explainable AI frameworks for network intrusion detection," *IEEE Access*, vol. 12, pp. 23954–23988, 2024, doi: 10.1109/ACCESS.2024.3365140.
- [7] N. L. Beebe and J. G. Clark, "A hierarchical, objectives-based framework for the digital investigations process," *Digital Investigation*, vol. 2, no. 2, pp. 147–167, 2005, doi: 10.1016/j.diin.2005.04.002.
- [8] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, 2016, pp. 785–794, doi: 10.1145/2939672.2939785.
- [9] V. Belle and I. Papantonis, "Principles and practice of explainable machine learning," *Frontiers in Big Data*, vol. 4, Art. no. 688969, 2021, doi: 10.3389/fdata.2021.688969.
- [10] F. Yan, S. Wen, S. Nepal, C. Paris, and Y. Xiang, "Explainable machine learning in cybersecurity: A survey," *International Journal of Intelligent Systems*, vol. 37, no. 12, pp. 12305–12334, 2022, doi: 10.1002/int.23088.
- [11] V. Corey, C. Peterman, S. Shearin, M. S. Greenberg, and J. Van Bokkelen, "Network forensics analysis," *IEEE Internet Computing*, vol. 6, no. 6, pp. 60–66, Nov./Dec. 2002.
- [12] Y. Chabot, A. Bertaux, C. Nicolle, and T. Kechadi, "An ontology-based approach for the reconstruction and analysis of digital incidents timelines," *Digital Investigation*, vol. 15, pp. 83–100, 2015, doi: 10.1016/j.diin.2015.07.005.
- [13] M. Wang, K. Zheng, Y. Yang, and X. Wang, "An explainable machine learning framework for intrusion detection systems," *IEEE Access*, vol. 8, pp. 73127–73141, 2020, doi: 10.1109/ACCESS.2020.2988359.
- [14] A. Bou Nassif, M. Abu Talib, Q. Nasir, and F. M. Dakalbab, "Machine learning for anomaly detection: A systematic review," *IEEE Access*, vol. 9, pp. 78658–78700, 2021, doi: 10.1109/ACCESS.2021.3083060.
- [15] J. Lansky et al., "Deep learning-based intrusion detection systems: A systematic review," *IEEE Access*, vol. 9, pp. 101574–101599, 2021, doi: 10.1109/ACCESS.2021.3097247.

- [16] A. Halbouni et al., "CNN-LSTM: Hybrid deep neural network for network intrusion detection system," *IEEE Access*, vol. 10, pp. 99837–99849, 2022, doi: 10.1109/ACCESS.2022.3206425.
- [17] A. Hassanzadeh, A. Rasekh, S. Galelli, M. Aghashahi, R. Taormina, A. Ostfeld, and M. K. Banks, "A review of cybersecurity incidents in the water sector," *Journal of Environmental Engineering*, vol. 146, no. 5, Art. no. 03120003, 2020, doi: 10.1061/(ASCE)EE.1943-7870.0001686.
- [18] H. T. T. Nguyen, H. Q. Cao, K. V. T. Nguyen, and N. D. K. Pham, "Evaluation of explainable artificial intelligence: SHAP, LIME, and CAM," in *Proc. FPT AI Conference (FAIC)*, 2021, pp. 1–6.
- [19] P. A. A. Resende and A. C. Drummond, "A survey of random forest based methods for intrusion detection systems," *ACM Computing Surveys*, vol. 51, no. 3, Art. no. 48, 2018, doi: 10.1145/3178582.
- [20] F. Pacheco, E. Exposito, M. Gineste, C. Baudoin, and J. Aguilar, "Towards the deployment of machine learning solutions in network traffic classification: A systematic survey," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 2, pp. 1988–2014, 2019, doi: 10.1109/COMST.2018.2883147.
- [21] C.-H. Lee, Y.-Y. Su, Y.-C. Lin, and S.-J. Lee, "Machine learning based network intrusion detection," in *Proc. 2nd IEEE International Conference on Computational Intelligence and Applications*, 2017, pp. 79–83, doi: 10.1109/CIAPP.2017.8167184.
- [22] B. S. Bhati, G. Chugh, F. Al-Turjman, and N. S. Bhati, "An improved ensemble based intrusion detection technique using XGBoost," *Transactions on Emerging Telecommunications Technologies*, vol. 32, no. 6, Art. no. e4076, 2021, doi: 10.1002/ett.4076.
- [23] Z. K. Maseer, R. Yusof, N. Bahaman, S. A. Mostafa, and C. F. M. Foozy, "Benchmarking of machine learning for anomaly based intrusion detection systems in the CICIDS2017 dataset," *IEEE Access*, vol. 9, pp. 22351–22370, 2021, doi: 10.1109/ACCESS.2021.3056614.
- [24] C. Park, J. Lee, Y. Kim, J.-G. Park, H. Kim, and D. Hong, "An enhanced AI-based network intrusion detection system using generative adversarial networks," *IEEE Internet of Things Journal*, vol. 10, no. 3, pp. 2330–2345, 2023, doi: 10.1109/JIOT.2022.3211346.
- [25] A. Akremi, H. Sallay, and M. Rouached, "Intrusion detection systems alerts reduction: New approach for forensics readiness," in *Security and Privacy Management, Techniques, and Protocols*. Hershey, PA, USA: IGI Global, 2018, pp. 255–275, doi: 10.4018/978-1-5225-5583-4.ch010.
- [26] N. Koroniotis and N. Moustafa, "Enhancing network forensics with particle swarm and deep learning: The Particle Deep framework," in *Proc. 7th International Conference on Artificial Intelligence and Applications*, 2020, doi: 10.5121/csit.2020.100304.
- [27] J. Fährndrich, W. Honekamp, R. Povalej, H. Rittelmeier, S. Berner, and D. Labudde, "Digital forensics and strong AI: A structured literature review," *Forensic Science International: Digital Investigation*, vol. 46, Art. no. 301617, 2023, doi: 10.1016/j.fsidi.2023.301617.
- [28] S. Alam and A. K. Demir, "SIFT: Sifting file types—application of explainable artificial intelligence in cyber forensics," *Cybersecurity*, vol. 7, Art. no. 52, 2024, doi: 10.1186/s42400-024-00241-9.
- [29] A. Manivannan and A. Amalanathan, "GeoForensic-AI: A lightweight and explainable forensic AI framework for modern digital ecosystems," *IEEE Access*, vol. 14, pp. 96450–96467, 2026, doi: 10.1109/ACCESS.2026.3686828.
- [30] J. R. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, no. 1, pp. 81–106, 1986, doi: 10.1007/BF00116251.

- [31] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001, doi: 10.1023/A:1010933404324.
- [32] G. Engelen, V. Rimmer, and W. Joosen, "Troubleshooting an intrusion detection dataset: The CICIDS2017 case study," in *Proc. 2021 IEEE Security and Privacy Workshops (SPW)*, 2021, pp. 7–12, doi: 10.1109/SPW53761.2021.00009.
- [33] D. Gaspar, P. Silva, and C. Silva, "Explainable AI for intrusion detection systems: LIME and SHAP applicability on multi-layer perceptron," *IEEE Access*, vol. 12, pp. 30164–30175, 2024, doi: 10.1109/ACCESS.2024.3368377.
- [34] A. Ghorbani, A. Abid, and J. Zou, "Interpretation of neural networks is fragile," in *Proc. AAAI Conf. Artificial Intelligence*, vol. 33, no. 1, 2019, pp. 3681–3688, doi: 10.1609/aaai.v33i01.33013681.
- [35] T. A. Roshinta and G. Szűcs, "A comparative study of LIME and SHAP for enhancing trustworthiness and efficiency in explainable AI systems," in *Proc. 2024 IEEE Int. Conf. Computing (ICOCO)*, 2024, pp. 134–139, doi: 10.1109/ICOCO62848.2024.10928183.
- [36] D. Slack, S. Hilgard, E. Jia, S. Singh, and H. Lakkaraju, "Fooling LIME and SHAP: Adversarial attacks on post hoc explanation methods," in *Proc. AAAI/ACM Conf. AI, Ethics, and Society (AIES)*, 2020, pp. 180–186, doi: 10.1145/3375627.3375830.
- [37] S. W. Hall, A. Sakzad, and S. Minagar, "A proof of concept implementation of explainable artificial intelligence (XAI) in digital forensics," in *Network and System Security, Lecture Notes in Computer Science*, vol. 13787, Cham, Switzerland: Springer, 2022, pp. 66–85, doi: 10.1007/978-3-031-23020-2_4.
- [38] Z. Khalid, F. Iqbal, and B. C. M. Fung, "Towards a unified XAI-based framework for digital forensic investigations," *Forensic Science International: Digital Investigation*, vol. 50, Art. no. 301806, 2024, doi: 10.1016/j.fsidi.2024.301806.

APPEDICES

The following appendices provide additional visual support for the experimental methodology, results, and discussion as discussed in the dissertation. All experimental outputs were generated via the research notebooks that were developed and used, and the screenshots of the dashboard show how the results are displayed.

Appendix A: Machine-Learning Model Evaluation

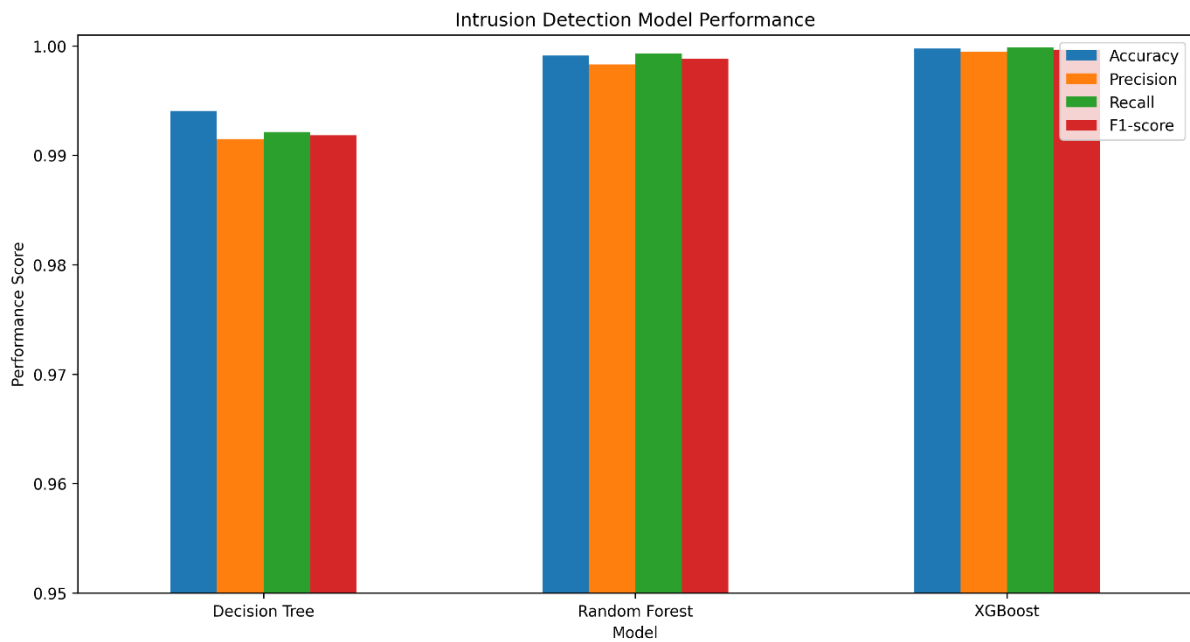


Figure A.1: Performance comparison of Decision Tree, Random Forest and XGBoost using accuracy, precision, recall and F1-score.

Appendix B: SHAP and LIME Explainability Outputs

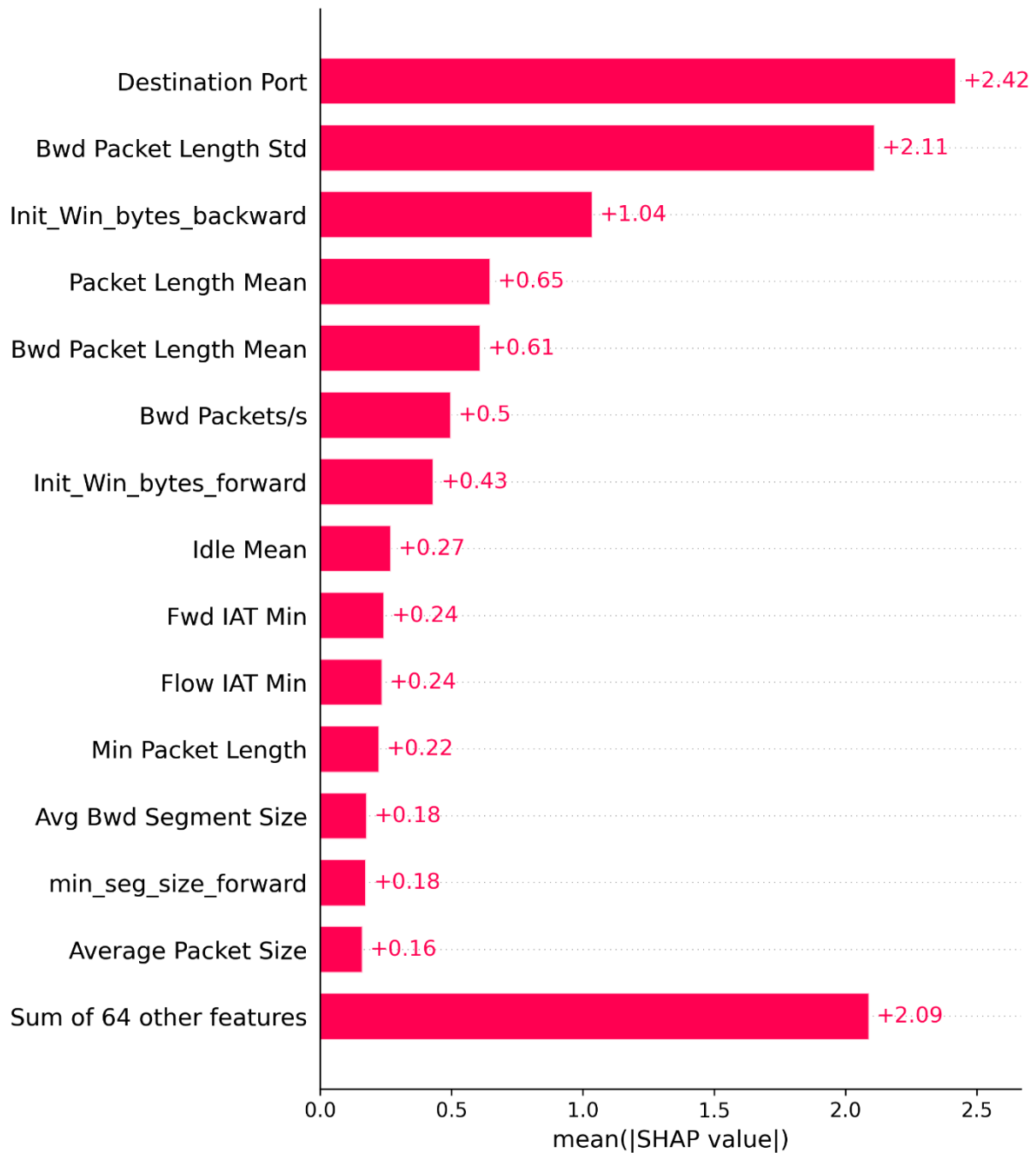


Figure B.1: Global SHAP summary plot

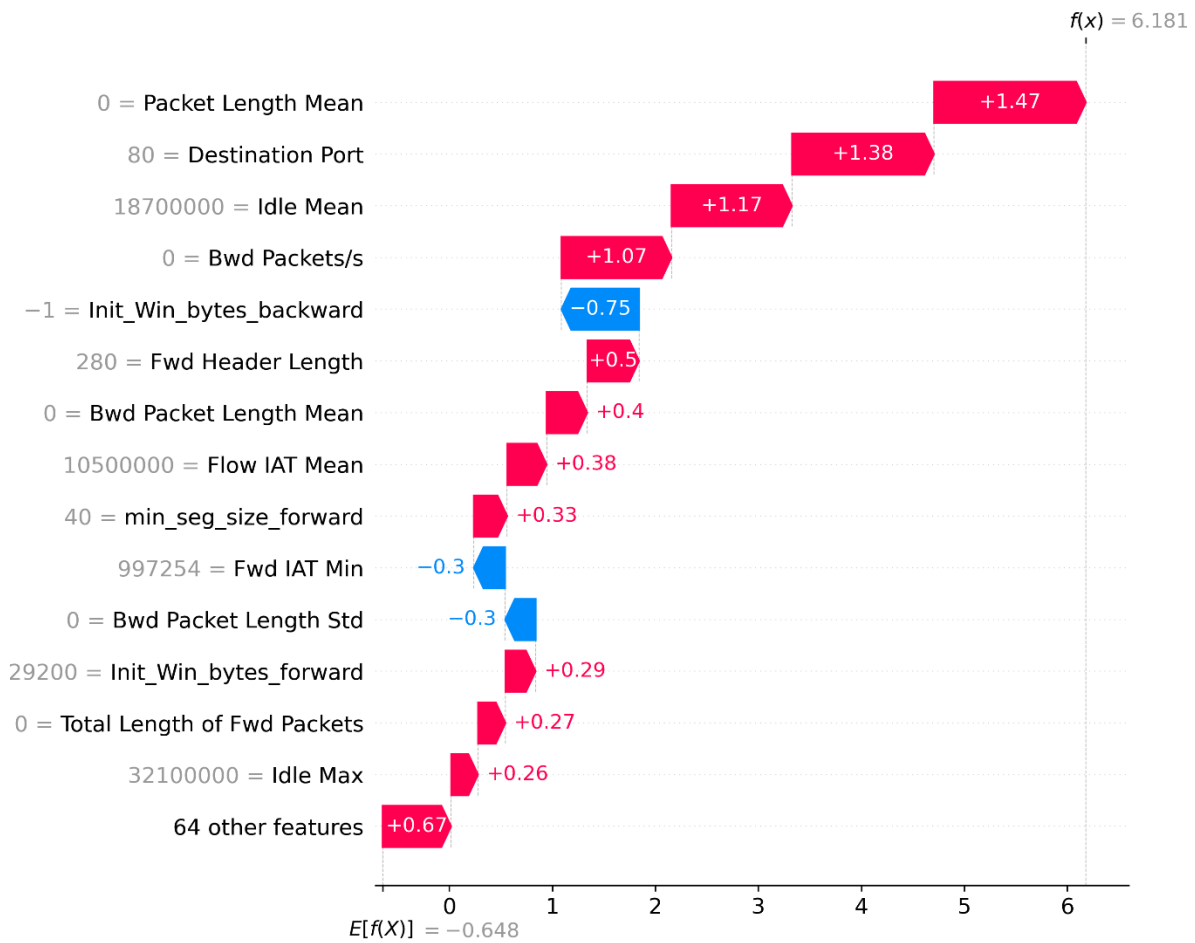


Figure B.2: SHAP waterfall explanation for the selected False Positive

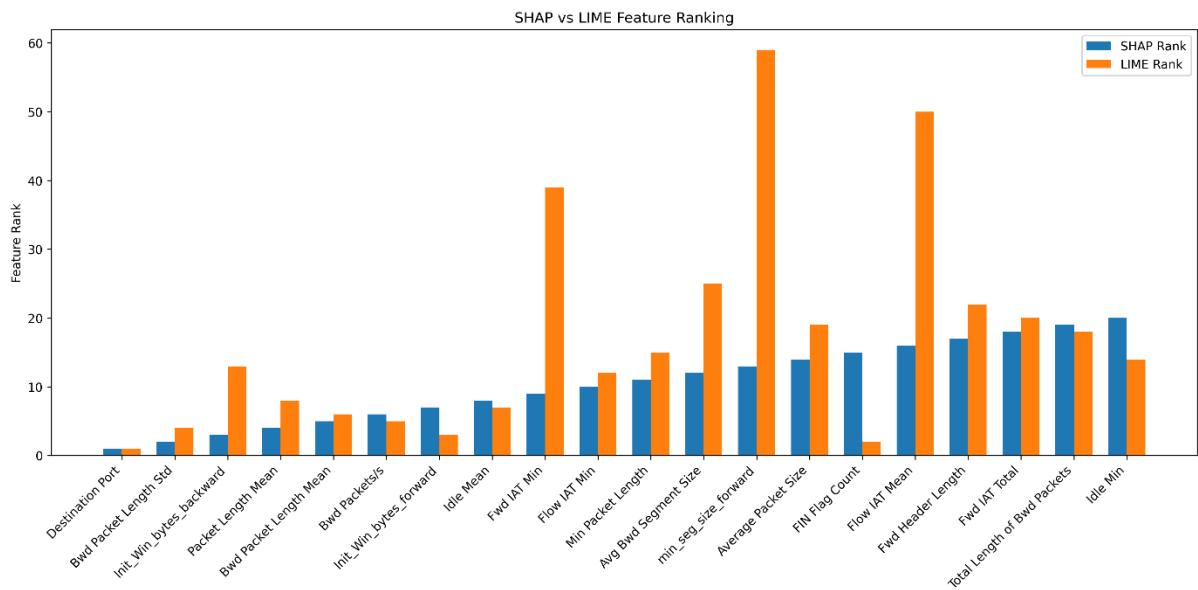


Figure B.3: Comparison of SHAP and LIME feature rankings

Appendix C: Explanation Stability

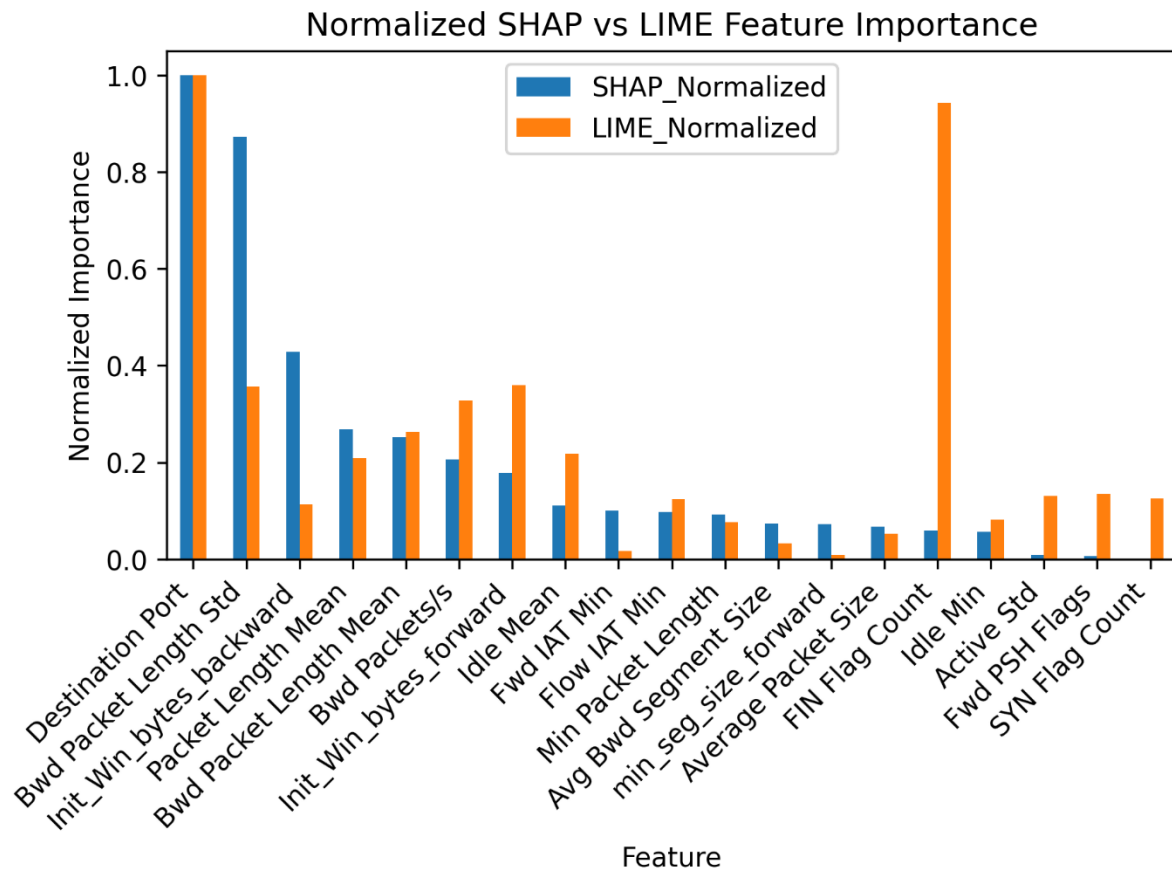


Figure C.1: Selected results from the explanation stability experiments

Appendix D: Research Dashboard

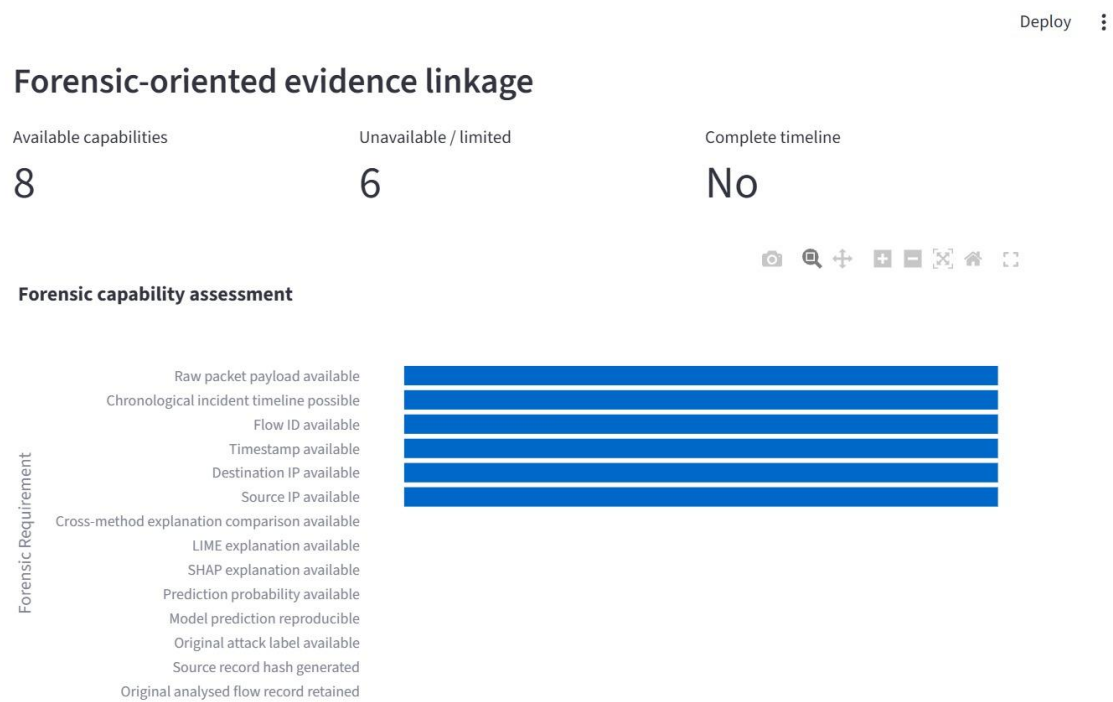


Figure D.1: Implemented Streamlit dashboard