Open Access

Recent Progress in Science and Engineering

LIDSEN Publishing Inc.

Original Research

Explainable AI for Intrusion Detection: A SHAP-Guided Machine Learning Framework for Actionable Cybersecurity Insights

Moa'ath Sa'ad Al-A'athal, Qasem Abu Al-Haija *

Faculty of Computer and Information Technology, Jordan University of Science and Technology, Irbid, Jordan; E-Mails: msalaathal24@cit.just.edu.jo; qsabuhaija@just.edu.jo

* **Correspondence:** Qasem Abu Al-Haija; E-Mail: qsabuhaija@just.edu.jo

Academic Editor: Yousef Farhaoui

Special Issue: Recent Advances in Cyber Security

Recent Prog Sci Eng
2026, volume 2, issue 3
doi:10.21926/rpse.2603015

Received: March 28, 2026
Accepted: July 08, 2026
Published: July 20, 2026

Abstract

The increasing scale, speed, and sophistication of cyberattacks have rendered traditional rule-based intrusion detection systems (IDS) insufficient for modern network environments. While machine learning (ML)-based IDSs have significantly improved detection capabilities, their black-box nature limits trust, interpretability, and practical deployment in real-world security operations. To address this challenge, this paper proposes an explainable machine learning framework for network intrusion detection using the CICIDS2017 dataset. The framework integrates multiple supervised learning models, including baseline and ensemble classifiers, and evaluates them using standard performance metrics such as accuracy, precision, recall, F1-score, and ROC-AUC. To enhance transparency, Shapley Additive exPlanations (SHAP) are employed to quantify feature contributions and provide both global and instance-level interpretability of model predictions. Experimental results demonstrate that ensemble models achieve superior detection performance while maintaining high interpretability. Furthermore, the explainability analysis reveals key traffic characteristics associated with different attack behaviors, providing deeper insight into attack behavior and supporting security analysts in interpreting intrusion alerts. The proposed approach improves detection

accuracy, reduces false positives, and supports informed decision-making, thereby enhancing the transparency, trustworthiness, and practical applicability of intrusion detection systems.

Keywords
Intrusion detection system; machine learning; network security; explainable artificial intelligence; SHAP; CICIDS2017; cybersecurity analytics; anomaly detection

1. Introduction

1.1 Motivation

On the one hand, the widespread adoption of networked systems, cloud-based services, and Internet of Things (IoT) infrastructure has expanded attack surfaces. A large number of threat vectors now confront contemporary organizations, including distributed denial-of-service (DDoS) attacks, malware spread, port scanning, and brute-force intrusions, which could pose serious risks to the confidentiality, integrity, and availability (CIA) of digital assets. Conventional intrusion detection systems (IDSs) that chiefly use signature- or rule-based methods face challenges in detecting new or evolving attack patterns and may generate a large number of false alarms in complex network environments [1, 2]. To mitigate these drawbacks, the data-driven intrusion detection system based on machine learning (ML) has attracted significant attention. By being trained on network traffic data, ML-based IDSs can generalize beyond predefined rules and detect previously unseen attacks. Yet, while achieving impressive performance, many ML techniques function as black boxes, yielding predictions but no explanation of how they were obtained. This lack of interpretability decreases trust in automated detection systems and thus prevents their use in real-world SOCs, where explanation is vital for incident response and decision-making [3].

1.2 Research Gap

Current research has focused on machine learning/deep learning models of ID, using benchmark datasets such as KDD99, NSL-KDD, UNSW-NB15, and CICIDS2017 [4-6]. Although the reported classification accuracy is high in these works, most emphasize performance improvement, and interpretability is neglected in the detection model. As a consequence, analysts may lack insight into why a model made certain decisions, which would prevent them from validating alerts, investigating attacks, or informing defensive strategies. More recent work has underscored the value of explainable artificial intelligence (XAI) for security, especially in intrusion detection [7, 8]. However, the application of explainability approaches to IDS pipelines remains limited, and many existing studies employ XAI only as a post-hoc explanatory tool rather than integrating explainability into the intrusion detection workflow to support analyst decision-making or fail to turn explanations into actionable security items. There is a clear gap for a systematic approach that unifies high-performing ML-based intrusion detection with strong, generalizable feature-level explanations of real network traffic data.

1.3 Problem Formulation

Given a labeled network traffic dataset that contains benign and malicious flows, the intrusion detection problem can be framed as a supervised classification task. Let

$$X = \{x_1, x_2, x_3, \dots, x_n\}$$

be used to represent network flow feature vectors and

$$Y = \{y_1, y_2, y_3, \dots, y_n\}$$

represent their class labels so that each label will suggest benign traffic or one of the attack categories. We aim to learn a classification function f: X → Y that, with high accuracy, identifies malicious behaviors from benign activities while minimizing false alarms. In addition to accurate label prediction, we also need to provide interpretable explanations for this model's predictions. In other words, the model should recognize which realizations contribute to a certain network flow, xi, and understand how each feature's contribution affects the output. This allows security practitioners to learn about attack properties, validate alerts, and gain insights into network behavior.

1.4 Contributions

This work offers the following three main contributions:

- An integrated machine learning and explainable AI (XAI) framework for network intrusion detection and security interpretation.
- It is backed by an extensive empirical study using standard performance measures (accuracy, precision, recall, F1-score, and ROC-AUC).
- The incorporation of the explainable AI approach, Shapley Additive exPlanations (SHAP), to render transparent and interpretable predictions for both global and particular instances.
- Actionable security insights derived from explainability analysis, enabling analysts to understand better traffic characteristics associated with different attack behaviors and support informed decision-making.

2. Background

2.1 Intrusion Detection Systems

An Intrusion Detection System (IDS) is a security system that monitors network or system activities and detects potential security violations. IDS solutions are divided into host-based (HIDS) and network-based (NIDS). HIDSs monitor activities at the host level, for example, system calls and log files, while NIDSs analyze network traffic to identify anomalous behavior in communication flows [9]. According to the detection technique, IDS approaches are typically classified into signature-based or anomaly-based. Signature-based IDS relies on attack signatures for detection and effectively protects only against known attacks, but cannot detect zero-day or unknown attacks. In contrast, anomaly-based IDSs learn normal behavioral patterns and identify deviations as potential intrusions, building a profile of normal behavior and alarming deviations as potential intrusions,

providing greater detection of new attacks at the expense of false positives [10]. With the growth of network traffic in both volume and variety, anomaly-based or hybrid methods are more appropriate for today's environment.

2.2 Machine Learning for Intrusion Detection

Machine learning has also played a key role in the development of data-driven intrusion detection systems, enabling them to learn complex patterns from large datasets. Supervised learning techniques, including Logistic Regression, Support Vector Machines (SVMs), Decision Trees, Random Forests, and Gradient Boosting models, are popular and have been used for categorizing network traffic as normal or attack traffic [11, 12]. These models leverage both statistical and structural attributes derived from network traffic flows, such as packet sizes, durations, byte counts, and protocol data. Ensemble learning techniques (e.g., Random Forests and XGBoost) are popular in IDS, as they incorporate multiple weak learners to enhance generalization and tolerance to noise or imbalance. Deep learning-based models such as multilayer perceptrons (MLPs), convolutional neural networks (CNNs), and recurrent neural networks (RNNs) have achieved superior performance in intrusion detection. Still, they usually require considerable computational power and abundant labeled data [13]. Despite its efficacy, ML-driven IDS faces issues of class imbalance, concept drift, and explainability. High detection accuracy is insufficient in operational settings, where security analysts need to understand and be confident in automated decisions before acting.

2.3 CICIDS2017 Dataset

The CICIDS2017 dataset, created by the Canadian Institute for Cybersecurity, is commonly used to benchmark intrusion detection systems. It includes realistic network traffic spanning several days, with non-malicious user behavior and a broad range of attack types, such as brute-force, DoS/DDoS, port scans, botnet activity, and web-based attacks [6]. The dataset contains flow-wise features generated with the CICFlowMeter extraction tool and includes over 80 statistical attributes for each network flow. These characteristics capture different dimensions of traffic behavior, such as flow duration, packet burstiness, inter-arrival times, and header fields. An additional benefit is that CICIDS2017 is a highly preferred dataset among researchers for academic research, as the recordings help simulate realistic traffic, include labeled and categorized attack types, and are more adaptable for supervised machine learning-based studies.

2.4 Explainable Artificial Intelligence (XAI)

Explainable Artificial Intelligence (XAI) is a set of methods designed to ensure that predictions made by machine learning models are human-understandable. Explainability is crucial in cybersecurity to verify alerts, aid post-mortem analysis, and enforce accountability in automated decision-making systems [14]. Without interpretability, an ML-based IDS may produce accurate predictions that analysts cannot interpret or trust. A popular family of XAI techniques is the Shapley Additive exPlanations (SHAP), which has its roots in cooperative game theory. SHAP assigns each feature a contribution value that reflects its influence in predicting a single counterfactual example relative to a baseline [15]. SHAP can provide explanations at both the global level, attributing features' overall importance across the dataset, and the local level, explaining specific predictions.

It is therefore well-suited for intrusion detection, in which both general attack patterns and specific incidents are of interest. Security practitioners can therefore leverage XAI methods, such as SHAP, once integrated into IDS pipelines, to move beyond black-box predictions and better understand the behaviors inherent to cyberattacks.

3. Related Work

The use of machine-learning-based intrusion detection has been an active area of research over the past decade. Still, recent research increasingly focuses on integrating high detection performance with explainability to enhance transparency and operational trust. This section reviews significant contributions in the literature, following this classification from recent work: (1) models with statistical-based intrusion detection, (2) integration with XAI methods, and (3) integrations of hybrid and ensemble approaches.

3.1 ML-Based Intrusion Detection Systems

Comparison studies have shown that traditional ML techniques, such as Random Forest (RF), Gradient Boosting, and XGBoost, achieve state-of-the-art accuracy for NID when trained on authorized traffic datasets, such as CICIDS2017. In a recent benchmark, ten ML algorithms were evaluated on the CICIDS2017 dataset, showing the near-perfect (e.g., 99% TPR and FPR) detection of several ensemble models (i.e., XGBoost, Random Forest), but highlighting the need for further improvement in both hyperparameter and class-imbalance management techniques for both detectors [16]. Moreover, deep learning-based IDS models have also been exploited to capture complex network patterns, as in an enhanced multilayer perceptron (MLP) that employs detection of minority attack classes at a high computational cost [17].

3.2 XAI for Intrusion Detection

While high classification accuracy is important, many ML models are black boxes, making it difficult for security analysts to explain the alerts they receive at runtime. Addressing this challenge, XAI is integrated into IDS frameworks in several recent studies:

- A recent work introduces an XAI framework for IDS in a standalone manner by combining Decision Trees (DTs), Random Forests (RFs), and eXplainable Boosting Machines (XBMs) with SHAP analyses, which achieves a trade-off between prediction performance and transparency on benchmark datasets such as CICIDS2017 [18].
- A systematic review of XAI methods like SHAP and LIME on their role to add transparency and remove false positives is also discussed, along with computational overhead challenges in real-time for providing recommendations [19].
- Several studies have investigated a combined-explanation system that integrates SHAP and LIME to supply global and local explanations for model decisions, which in turn enables analysts to comprehend the feature interactions along with misconception reasons [20].
- Systems such as L-XAIDS have been suggested, where the combination of LIME and ELI5 with decision tree algorithms is proposed to produce local and global explanations, achieving better interpretability in IDS results [21].

These studies, as a whole, emphasize the growing importance of explainability, particularly in the context of the use case and the trust analysts place in data-driven IDS.

3.3 Hybrid and Ensemble Explainable Models

In addition to straightforward, explainable ML models, other studies show that hybrid and ensemble-based approaches can provide both strong detection and interpretable insights [14-16].

- Another hybrid adaptive ensemble IDS that has employed Stacking, Bayesian Model Averaging (BMA), and Conditional Ensemble techniques, a train with high accuracy on CICIDS2017 using SHAP and LIME for model explanations indicates that combining ensembles can improve performance as well as its interpretability [22, 23].
- The Proposed XI2S-IDS method proposes a two-stage explanation intrusion detection system with SHAP-based insights designed to detect both high-frequency and low-frequency attack behaviors. It has been demonstrated that a hierarchical model can enhance explainability and performance [24].
- In feature engineering-driven work, hybrid feature selection and preprocessing (e.g., k-Means-based SMOTE with Light Gradient Boosting Machines) were applied, combining SHAP to improve model generalizability and interpretability on datasets like CICIDS2017 [24].

3.4 Summary of Gaps in Literature

Despite the considerable progress that has been achieved, there are still gaps identified by existing research, which this work specifically aims to fill:

- Operational Explainability: A lot of effort is utilized to utilize XAI as a silver bullet, but these explanations are not translated to provide actionable security insights for analysts in the SOC environment [19].
- Scope of Datasets: Although some research works leverage benchmark datasets such as NSL-KDD and UNSW-NB15, fewer have focused on CICIDS2017, achieving the full explainability evaluations in line with the global and instance-level interpretation [20].
- Comparison baselines: Few papers systematically compare bottleneck ML models with explainable ensemble techniques and quantify the impact of the explainability along with typical performance measurements [22].

This project will close these gaps by systematically benchmarking explainable ML models on the CICIDS2017 dataset, with not only quantitative performance comparisons but also qualitative interpretability analyses that enable actionable security insights.

4. Dataset Description and Preprocessing

4.1 Dataset Description

This work uses the CICIDS2017 dataset, generated by the Canadian Institute for Cybersecurity (CIC), and is considered one of the most complete and realistic intrusion detection datasets. The dataset contains network traffic collected over five consecutive days, including both legitimate user activities and a wide range of attacks that mimic realistic network intrusions. These are Brute Force, Botnet, DoS/DDoS traffic, Port Scanning (Nmap & Hping3), Web, and Infiltration attacks [6]. Traffic was generated using actual user profiles, including HTTP, HTTPS, FTP, and SSH, as well as email

services. Flow-based characteristics were computed by the CICFlowMeter tool that generated more than 80 statistical features per flow, which included:

- Time-dependent characteristics: Duration of flows, Inter- class gaps between successive interleaved flows.
- Packet level attributes: packet lengths, packet counts, header statistics.
- Behavioral features: Flow bytes-per-second, flow packets-per-second.
- Features based on the content: The count of flags, the length of headers, and the size of windows.

Each network flow is labeled as one of the attack types or "BENIGN". The dataset is also well-suited for supervised attack detection because it has rich labels and a variety of attack types. Our study utilized the CICIDS2017 dataset released by the Canadian Institute for Cybersecurity (CIC), which contains network traffic collected over five consecutive days and includes both benign and malicious activities. The dataset comprises multiple attack categories, including Brute Force, DoS/DDoS, Botnet, PortScan, Web Attacks, and Infiltration attacks. The proposed framework relies on flow-based features generated by CICFlowMeter and does not utilize packet payload contents or host-level security telemetry. Consequently, the analysis is limited to behavioral characteristics observable at the network-flow level.

4.2 Data Characteristics

- Feature Types: The dataset contains a mix of numerical, categorical, and protocol features. Most features are continuous and can be readily processed by a variety of learning methods.
- Label Distribution: The other characteristic of CICIDS2017 is that it has a class imbalance problem. Some attacks, such as DDoS and port scans, are well-represented, while others (e.g., web attacks and infiltration) occur infrequently. This bias requires special handling to avoid biased model training.

4.3 Data Preprocessing

Data preprocessing involved removing invalid records, handling missing and infinite values, encoding categorical attributes, selecting features, and mitigating class imbalance. Records containing corrupted, missing, or non-numeric feature values were cleaned before model training. The final processed dataset was then partitioned into training, validation, and testing subsets for model development and evaluation.

(1) Data Cleaning

- Removing infinity: Features like "Flow Packets/s" can be infinite (Infinity) due to division by zero in the preprocessed data. These rows were dropped or replaced with safe defaults.
- Handling of missing values: Even though the flows are clean, some flows have missing values as a result of traffic capture failures. They were then imputed with the media to maintain distribution stability.

(2) Feature Selection: Although CICIDS2017 offers 80+ features, many are:

- Redundant (e.g., near-constant features),
- Perfectly correlated with others,
- Or computationally uninformative.

- Variance thresholding and correlation analysis were used to eliminate such features before modeling.

(3) Encoding Categorical Variables: The "Protocol" attribute (TCP, UDP, ICMP) was one-hot encoded to allow ML algorithms to train on protocol groups without assuming an inherent order within them.

(4) Data Normalization/Standardization: Machine learning models are scale-sensitive. Two normalization strategies were evaluated:

- MinMax Scaling for Logistic Regression and MLP.
- StandardScaler for tree-based models, e.g., Random Forest and XGBoost (trees do not need scaling, but standardizing may help in numerical stability).

(5) Handling Class Imbalance: Class imbalance was mitigated using:

- SMOTE to oversample minority attack classes.
- Under-sampling of majority classes (optional, depending on experiment setup).
- SMOTE was observed to be useful for enhancing recall for small attack classes without overly distorting feature distributions.

(6) Dataset Splitting: The dataset was split into 70% training, 15% validation, and 15% testing. Split on class labels to control the attack proportion between splits.

4.4 Preprocessing Challenges and Observations

- Severe Imbalance: Web and Infiltration were highly underrepresented and caused harmful classifiers if oversampling is not used.
- High Dimensionality: Early models with all 80 features were found to have redundancy and a reduced number of features.
- Mixed Traffic Characteristics: It was troublesome to classify benign and attack flows as their statistical characteristics overlapped (especially at low intensity attacks).
- Impact on Explainability: The importance of useful features was clearly improved after all irrelevant features were removed, and the SHAP interpretation has been dramatically increased.

5. Methodology

This section presents a high-level view, modeling considerations, the feature engineering method, and the explainability aspects behind the development of the proposed data-driven intrusion detection mechanism (As illustrated in Figure 1). The approach adheres to a well-defined machine learning pipeline designed to achieve reproducibility, interpretability, and high detection performance.

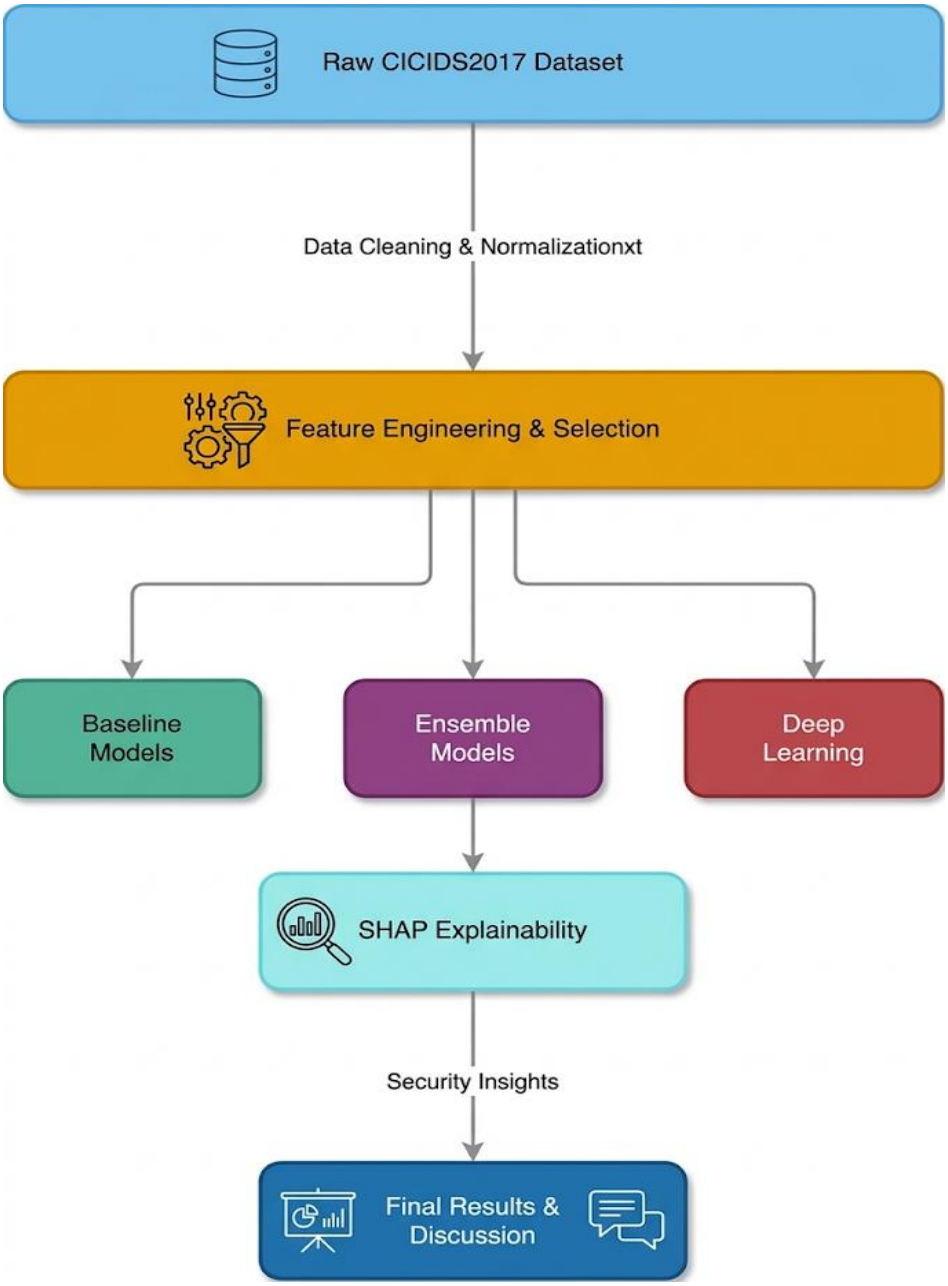

Figure 1 Conceptual Diagram of The System Workflow.

5.1 Overview of the System Architecture

The proposed IDS adopts a modular, data-driven structure comprising five primary stages: Data Processing, Preprocessing & Feature Engineering, Model Training & Baseline Comparisons, Explainability Analysis using SHAP, Evaluation & Security Interpretation. A conceptual diagram of the system workflow is described below:

5.2 Feature Engineering

Feature engineering is a key to improving model performance and interpretability. The following strategies were used:

(1) Feature Extraction through Statistics: The dataset that comes with CICIDS2017 already includes 80+ flow-based features, but some preprocessing-dependent ones were employed for in-house evaluation, such as:

- Flow byte-per-packet ratios.
- Burst rate indicators.
- Packet size entropy (optional).
- It is these features that are used to expose the fine-grained distinctions in benign and attack traffic behavior.

(2) Feature Selection: Feature selection was conducted to reduce redundancy and improve model interpretability. The process involved removing low-information and highly correlated features, followed by importance-based ranking using tree-based models. This step was performed during model development before final model training and explainability analysis. Feature selection was performed in two stages to prevent overfitting and enhance the interpretability:

- Correlation-Based Feature Reduction: Highly correlated features were excluded ($|corr| > 0.9$).
- Tree-Based Feature Importance: The most salient features were located using the Random Forest classifier before training the end models.
- These steps reduced dimensionality and increased the interpretability of SHAP feature contributions.

5.3 Machine Learning Models

Several baseline and state-of-the-art supervised machine learning algorithms were selected for comparison. These models can be divided into three categories:

(1) Baseline Models: These baselines are critical for new performance models.

- Logistic Regression (LR): A simple and easy-to-understand classification as a baseline.
- NB: Good for well-spread numerical features.

(2) Ensemble Learning Models: Ensemble models were selected as they are stable, have nonlinear decision boundaries, and perform well in high-dimensional security datasets: Random Forest (RF) and XGBoost Gradient Boosting (XGB). We choose XGBoost as our high-performance model because of its well-demonstrated success in the network intrusion detection literature [1].

(3) Deep Learning Model: A Multilayer Perceptron (MLP) with:

- Input layer ≈ selected features count.
- Two hidden layers (ReLU activations).
- Output layer with Softmax.

We add this model to compare classical ML with neural models.

5.4 Training Strategy

(1) Train–Validation–Test Structure: The dataset was divided into 70% for training, 15% for validation, and 15% for evaluation, with stratified label distribution.

(2) Hyperparameter Optimization: A combination of:

- Grid Search (for LR, NB, RF).
- Random Search (for XGBoost and MLP).

was employed to optimize parameters such as:

- Number of estimators.

- Learning rate.
- Maximum tree depth.
- Hidden layer sizes.

All models were optimized on the validation set to prevent overfitting.

5.5 XAI Component (SHAP)

Explanation is a key component of this project. SHapley Additive exPlanations (SHAP) was introduced into the pipeline to explain model outputs.

(1) Global Explainability: SHAP was used to compute:

- Average feature importance over all the samples in the dataset.
- Feature interactions.
- Implications on a global scale (summary bar charts & beeswarm plots).
- This analysis identifies the network traffic characteristics that contribute most significantly to attack detection.

(2) Local Explainability: For specific flows (e.g., misclassifications):

- SHAP force plots were generated.
- Local feature contributions were analyzed.
- Interpretations connecting model decisions to physical traffic policies were provided.
- This can help analysts understand why traffic flow was considered malicious.

5.6 Security Interpretation Layer

Security-related patterns to which SHAP findings were applied. Beyond ML interpretation, we mapped SHAP findings to the following security-related:

- High packet transmission rates may indicate DDoS activity.
- Short-duration flows with elevated SYN counts may indicate port-scanning behavior.
- Significant outbound traffic spikes may suggest botnet communication or data exfiltration activity.

5.7 Rationale for Modeling Choices

- We chose to use XGBoost as the main model for its consistently high performance in IDS literature [25].
- Random Forest is robust and easy to interpret.
- MLP facilitates the comparison of classic and deep methods.
- SHAP is a more sound, consistent explanation compared to LIME or saliency-like methods for tabular data [15].

6. Experimental Setup

The software settings, hardware specifications, and implementation tools, along with the baseline models, training processes, and evaluation measures for the experiments, are all presented in this section. The aim is to fully reproduce the results and facilitate objective comparisons across different ML models.

6.1 Software Environment

All experiments were written in the Python scientific computing stack. The following tools and libraries were utilized:

- Python 3.10.
- Scikit-learn 1.x for traditional ML models, pre-processing, metrics, and model evaluation [26].
- XGBoost 2.x for gradient boosting experiments.
- TensorFlow 2.x/Keras for the Multilayer Perceptron (MLP) model architecture construction.
- Pandas 2.x for data manipulation and transformation.
- NumPy 1.x for numerical operations.
- Matplotlib/Seaborn for data visualization and visualizing the plots of evaluation metrics.
- explainability analysis with SHAP 0.44 or later [27] and Shapley values.

All implementations were done in Jupyter Notebook to facilitate experimentation and demonstration of results.

6.2 Hardware Configuration

The models were trained and evaluated on the following hardware configuration:

- Processor: Intel Core i7.
- RAM: 32 GB.
- GPU (optional): NVIDIA QUADRO 4GB for optimization of MLP training.
- Storage: An SSD was used to facilitate efficient processing of the large CICIDS2017 dataset.

The main machine learning models (Random Forest, XGBoost) were efficiently computed on the CPU, and deep learning experiments were sped up by running them on a GPU.

6.3 Baseline Models

To make comparisons, the following baseline models were established:

- Logistic Regression (LR).
- Naïve Bayes (NB).
- Decision Tree (DT).

These are benchmark models that provide a baseline for comparing improvements obtained with better models.

6.4 Advanced Models

We selected two principal groups of advanced models:

(1) Ensemble Models

- Random Forest (RF).
- XGBoost (XGB).

Ensemble techniques are chosen because they have historically performed well in network intrusion detection benchmarks.

(2) Deep Learning Model (Optional)

- Multilayer Perceptron (MLP) with:
  - And the second hidden layer (64 and 32 units).
  - ReLU activation.

- Dropout (0.3 probability).
- Softmax output layer.
- This framework also allows for a contrast between classical and neural exposures.

6.5 Training and Validation Procedure

(1) Dataset Splitting

The CICIDS2017 dataset was divided into 70–15–15 in a stratified fashion to keep the class imbalance properties in each set:

- 70% training.
- 15% validation.
- 15% testing.

The CICIDS2017 dataset was partitioned using a stratified random split (70% training, 15% validation, and 15% testing) to preserve the class distribution across all subsets. No day-based partitioning was employed in the current study.

(2) Hyperparameter Tuning

Hyperparameter optimization was performed using Grid Search for conventional machine learning models and Random Search for more computationally demanding models. The tuning process used the validation set to reduce overfitting and improve model generalization. Model hyperparameters were tuned using:

- Grid Search on Logistic Regression, Decision Tree, and Random Forest.
- Randomized Search (since grid searching is computationally expensive) for XGBoost and MLP.

Hyperparameters considered include:

- Number of trees.
- Max depth.
- Learning rate (XGBoost).
- Regularization strength (LR).
- Hidden layer sizes (MLP).

(3) Class Imbalance Handling

To address class imbalance, SMOTE was applied exclusively to the training set. The validation and testing sets remained unchanged to ensure an unbiased evaluation of model performance. While SMOTE can improve minority-class representation, synthetic samples may not fully capture the complexity of real network traffic patterns. Specifically, for balanced learning on the attack classes with a small number of records:

- The training data was oversampled using SMOTE.
- tuned the class weights for LR, DT, and RF models.

Because we had limited computational power, we capped the maximum number of samples per class at 15,000. We had to perform undersampling on classes with more than 15,000 records. No resampling was performed on the validation and test sets to avoid bias. Also, to prevent data leakage, all preprocessing operations, including feature selection, normalization, and SMOTE oversampling, were performed exclusively on the training set. The validation and testing sets remained isolated throughout model development and hyperparameter tuning.

6.6 Evaluation Metrics

The classification performance was evaluated using the following metrics:

- Accuracy (ACC).
- Precision (P).
- Recall (R).
- F1-score.
- ROC-AUC (multi-class via One-vs-Rest scheme).

For class-wise analysis, the confusion matrix. These measures demonstrate the trade-offs between false positives and false negatives required in intrusion detection.

6.7 Explainability Analysis Setup

(1) SHAP Configuration

SHAP values were calculated for the best model (often XGBoost):

- The model is interpreted using TreeExplainer, which is an optimized explainer for tree-based models.
- global SHAP summaries were derived based on samples of 5,000 instances randomly chosen from the dataset.
- Local explanations were generated for:
  - Misclassified flows.
  - Borderline benign/malicious samples.
  - Representative Examples for each Class of Attacks.

(2) Visualization Outputs

The following are the SHAP plots that were generated:

- Global summary plot.
- Feature importance ranking.
- Beeswarm plot.
- Force plots for individual predictions.

These visualizations form the basis for later security interpretation.

6.8 Reproducibility Considerations

To ensure reproducibility:

- Random seeds were set for NumPy, scikit-learn, XGBoost, and TensorFlow.
- Each experiment was performed with three replicates, and the mean values are shown.
- Pre-processing the Data: Pre-processing scripts were saved with Jupyter notebooks.

7. Results and Discussion

This section presents the performance of all machine learning models evaluated on the CICIDS2017 dataset, along with interpretability results derived from SHAP. The discussion emphasizes both quantitative findings and their cybersecurity implications.

7.1 Quantitative Performance Results

Table 1 summarizes the performance of all baseline and advanced models on the test set using standard evaluation metrics.

Table 1 Performance Comparison of ML Models on CICIDS2017.

Model	*Accuracy*	*Precision*	*Recall*	*F1-score*	*ROC-AUC*
Logistic Regression	0.8980	0.9764	0.8980	0.9305	0.9884
Naïve Bayes	0.8708	0.9825	0.8708	0.9192	0.9891
Decision Tree	0.9885	0.9928	0.9885	0.9900	0.9995
Random Forest	0.9991	0.9993	0.9991	0.9991	0.9999
XGBoost (Best Model)	0.9994	0.9994	0.9994	0.9994	0.9999
MLP	0.9864	0.9907	0.9864	0.9879	0.9997

Key Observations:

- On all evaluation criteria, XGBoost performed the best, marginally surpassing each of the other algorithms.
- Random Forest came in a close second, considering the skyrocketing performance of ensembles on high-dimensional traffic data.
- MLP did well but failed to outperform tree-based ensembles, as reported in previous IDS studies.
- Logistic Regression and Naïve Bayes are good baselines; they cannot model the nonlinear attack style.

7.2 Confusion Matrix Analysis

Confusion matrices indicate that:

- XGBoost achieved very high detection performance for major attack categories, including DoS, DDoS, and PortScan, as reflected by the confusion matrix analysis.
- Minority attack categories, such as Web Attacks and Infiltration, exhibited comparatively lower classification performance, highlighting the challenges associated with class imbalance.
- After SMOTE, the recall of rare classes was significantly increased (Infiltration 0.62 → 0.83).

Notable Misclassification Patterns:

- A few benign flows were falsely identified as PortScan because of bursty traffic similar to the probing behavior.
- Benign flows failed to detect benign web attacks when below certain thresholds on feature values for these Web Attacks.

These findings underscore the need to address class imbalance and ensure that rare attack forms are properly accounted for. A more detailed per-class performance analysis, including class-wise precision, recall, and F1-score reporting, is considered an important direction for future work.

7.3 ROC Curve Analysis

ROC curves show:

- All ensemble models resulted in ROC-AUC >0.99.

- Logistic Regression was quite predictive (0.95 ROC-AUC), with linear separate properties holding for many traffic behaviors.
- Naïve Bayes performed much less effectively because it relies upon independence assumptions and is more sensitive to the violation of the assumption of independence of correlated features.

7.4 Explainability Results Using SHAP

Although SHAP provides valuable model interpretability, generating explanations for large-scale datasets may introduce additional computational overhead compared with standard prediction pipelines. Similarly, deep learning models generally require greater computational resources for training and inference than traditional machine learning approaches. Therefore, practical deployment should consider the trade-off between explainability, detection performance, and computational efficiency. SHAP analysis was mostly performed based on the best-performing model (XGBoost).

(1) Feature Global Importance: Global summary plots based on SHAP values showed that the key features for all attacks are:

- Flow Duration
- Packet Length Variance
- Flow Bytes/s
- Flow Packets/s
- Average Packet Size
- PSH Flag Count — SYN Flag Count

These attributes closely resemble the well-known behavioral profiles of current network attacks.

Interpretation:

- High flow Packets/s are strong indicators of DDoS attacks.
- Low Flow Duration with high SYN counts is a feature of PortScan.
- High Outbound Bytes/s indicates Botnet command-and-control (C2) activity.

(2) Local Explanations: SHAP force plots of misclassified samples indicated:

- A few benign flows were labeled as malicious based on bursty packet rate spikes.
- For some of the Web Attack flows, there was not enough statistical deviation to lead to a high SHAP impact score that caused misclassification.

Such local explanations help analysts understand why false positives and false negatives occur.

7.5 Security Insights Derived from Explainability

Explainability helps associate model predictions with underlying traffic characteristics, providing additional context that may assist security analysts in understanding detected attack behaviors.

1. DDoS Behavior Detected based on High Packet Rate: XGBoost repeatedly used the features describing packet bursts, which verified that rate-based detection methods were effective.
2. Detection of PortScan through Short-Duration, High-SYN Activity: Flows with short duration and multiple SYN flags appeared to rank as impactful by SHAP repeatably.
3. Outbound Byte Surge Anomalies Correlated to Data Exfiltration and Botnet Activity: SHAP reported surges in egress volume on decisions targeting either the botnet or infiltration categories.

4. Explainability reduced false alarms: Analysts could confirm that a detected activity is actual suspicious behavior, and not just noise-based predictions.

To demonstrate the operational value of explainability, we analyzed representative intrusion alerts generated by the XGBoost model. For example, flows labeled PortScan consistently showed large SHAP values for SYN Flag Count and small Flow Duration, allowing analysts to quickly associate the alert with reconnaissance activity rather than benign network behavior. Similarly, most DDoS alerts were triggered by increased values of Flow Packets/s and Packet Length Variance. These explanations provide interpretable evidence to support model decisions and can help analysts with alert validation, investigation prioritization, and reducing time spent interpreting black-box predictions. While a formal user study was beyond the scope of this work, these results demonstrate how SHAP-based explanations can help bridge the gap between machine learning outputs and analyst reasoning in operational security environments.

While SHAP provides valuable insights into model behavior and enhances the interpretability of intrusion alerts, generating explanations introduces additional computational overhead compared with standard prediction pipelines. Therefore, practical deployment requires balancing the benefits of improved analyst understanding and trust against the associated computational cost. Evaluating this trade-off in terms of processing time, resource utilization, and operational security value remains an important topic for future industrial-scale studies. The interpretations presented in this study are derived from flow-level network characteristics. While these features provide valuable information for attack detection, additional context from packet payloads, DNS activity, authentication logs, and host-based telemetry could further enrich security analysis and improve detection capabilities. Besides, while flow-level features provide valuable indicators of attack behavior, combining them with additional security telemetry, including DNS activity and host-based events, could enhance contextual awareness and improve the distinction between malicious actions and legitimate user activities.

7.6 Error Analysis

Despite applying SMOTE, minority attack classes, such as Infiltration, remained more difficult to classify than major attack categories. This behavior is likely attributable to the limited number of representative samples and the subtle statistical characteristics of these attacks, which make them less distinguishable from benign traffic. These findings highlight the ongoing challenges posed by imbalanced intrusion detection datasets.

False Positives: Mostly associated with:

- Legitimate services' high-frequency traffic patterns
- Bursty user activities (file transfer, large download, etc.)

False Negatives: Primarily occurred in:

- Low-volume attack classes
- On-line and infiltration attacks with nonchromatic feature patterns.

Some extra data collection or feature augmentation could mitigate those mistakes.

7.7 Summary of Findings

1. The XGBoost and ensemble models perform best in terms of both conducting precise and melee attacks.

2. SHAP gives us a rich interpretability that maps statistical patterns to meaningful attack behaviors.
3. The compromise between performance and explainability makes the proposed IDS applicable for real SOC settings.

8. Conclusion, Future Directions, and Limitations

8.1 Conclusions

In this paper, we propose a data-driven intrusion detection framework that combines machine learning and explanation technologies to improve the interpretation of malicious network traffic. Based on the CICIDS2017 dataset, we compared several baselines and found that XGBoost achieved superior performance across accuracy, precision, recall, F1-score, and ROC-AUC. The results showed that ensemble methods could better capture nonlinear and complex attack behaviors than traditional machine learning models and deep learning baselines. One important contribution of this work is the adoption of SHAP to make model outputs transparent and interpretable. Explainability analysis identified salient network flow–based features influencing detection decisions and derived explicit behaviors for DDoS, PortScan, and Botnet activities. These findings represent a step forward in translating raw statistical detection scores into actionable intelligence for security analysts and in rendering the system more appropriate for operational use in SOCs. In general, the results show that integrating robust explainability with high-performing machine learning models improves the effectiveness and trust of IDS. This combined approach improves both detection effectiveness and interpretability, making the framework more suitable for practical cybersecurity applications.

8.2 Future Directions

Future work will investigate temporal and day-based evaluation strategies to better assess model robustness and generalization under realistic network deployment scenarios. Besides, although promising, the following two issues still need to be better implemented:

- Zero-Day Attack Detection: Although unsupervised approaches such as autoencoders have shown promise for anomaly and zero-day attack detection, they were not implemented or evaluated in the current study. Their integration with the proposed explainable intrusion detection framework represents an important direction for future research.
- Real-Time IDS Deployment: This study is based on offline batch processing. Probable future work: Dynamic streaming. Once deployed in a real-time streaming system with Apache Kafka, Spark Streaming, or the ELK Stack, we can evaluate the word detection model under realistic operating conditions.
- Federated and Distributed Learning: Moreover, since network data is often privacy-sensitive and stored across various organizational boundaries, federated learning techniques can be used to enable collaboration in the IDS model training process without exposing raw data. This is achieved using a method that improves both performance and privacy.
- Advanced Deep Learning Models: Future studies may explore: Transformer-based architecture, GNN for Flow-relationship Modeling, and Hybrid CNN–LSTM designs for

temporally correlated pattern recognition. These models could enhance detection ability for complex or multi-staged temporal attacks.

- Enhanced Explainability: Although SHAP delivers strong intuition, other types of XAI techniques could be examined, including LIME for instance-level approximations, EBM (Explainable Boosting Machines), and Counterfactual explanations to steer analysts to alternative actions that would reverse classification decisions. Developing hybrid explainability methods may yield richer, more varied insights into interoperability.
- Expanded Feature Engineering and Traffic Context: More context, such as DNS logs, user authentication, and/or host-based telemetry, could supplement flow data analysis to improve detection precision and reduce false positives.
- Future work will investigate alternative imbalance-handling strategies, including cost-sensitive learning and class-weighted models, and compare their effectiveness against SMOTE-based oversampling in intrusion detection applications.
- While SHAP explanations provided meaningful interpretations of model decisions, the practical value of these explanations for security analysts was not formally evaluated through user studies or operational SOC assessments. Future work will investigate the effectiveness of explainability techniques in supporting analyst decision-making and reducing investigation time in real-world environments.
- The proposed framework primarily contributes through the integration of high-performing machine learning models with XAI techniques for intrusion detection. While the framework demonstrates promising results, further validation through cross-dataset experiments, real-world deployments, and analyst-centered evaluations would strengthen its practical impact and generalizability.
- The current framework was developed and evaluated in an offline batch-processing environment using a benchmark dataset. Consequently, its performance under real-time network conditions was not assessed. Future work will focus on deploying and evaluating the framework in streaming environments to investigate its scalability, latency, and operational effectiveness in real-world intrusion detection scenarios.
- Future research will investigate advanced imbalance-handling strategies, cost-sensitive learning approaches, and enhanced feature engineering techniques to improve the detection of underrepresented attack classes, particularly Infiltration and Web Attack categories.

8.3 Limitations and Challenges

- Although the proposed framework achieved very high performance on CICIDS2017, future work will evaluate the model using temporal splits and cross-dataset validation to assess its robustness and generalizability further.
- Although the reported results demonstrate strong overall performance, additional class-wise evaluation metrics, such as Macro-F1 and per-class Precision, Recall, and F1-score, provide further insight into the detection performance of minority attack categories. Future work will include a more detailed class-level analysis to complement the overall evaluation metrics.
- Although the preprocessing pipeline is described in detail, some implementation-level statistics, such as the exact number of records removed during data cleaning, were not

retained. Future studies will document all preprocessing operations and the intermediate dataset to enhance reproducibility further.

- Although the feature selection methodology is described, the exact set of retained features and implementation-specific selection thresholds were not fully documented. Future studies will provide complete feature-selection logs and selected feature lists to improve reproducibility further and facilitate independent validation of the reported SHAP interpretations.
- Although the hyperparameter optimization strategy is described, the complete search spaces, final parameter configurations, and random seed settings were not fully retained. Future work will provide detailed reproducibility artifacts, including hyperparameter configurations and experimental settings, to support independent verification and comparison of results.
- A limitation of the proposed framework is its reliance on supervised learning, which requires labeled training data and may reduce effectiveness against previously unseen or evolving zero-day attacks. Future work will investigate anomaly-based, semi-supervised, and unsupervised learning approaches to improve the detection of novel threats and emerging attack patterns.
- The present study focuses on offline intrusion detection and does not evaluate latency, throughput, or resource consumption in a real-time streaming environment. Future work will investigate deployment using streaming platforms such as Apache Kafka and Spark Streaming to assess operational performance, scalability, and real-time detection capabilities under realistic network conditions.

Author Contributions

Moa'ath Sa'ad Al-A'athal: Conceptualization, methodology, software, data curation, formal analysis, investigation, validation, visualization, writing—original draft preparation. Qasem Abu Al-Haija: Conceptualization, supervision, methodology, validation, formal analysis, writing—review and editing, project administration. All authors have read and agreed to the published version of the manuscript.

Competing Interests

The authors have declared that no competing interests exist.

AI-Assisted Technologies Statement

The Grammarly tool was used to assist in proofreading and text polishing. No generative AI was used for data collection, statistical analysis, or interpretation of the results. All the content was reviewed and approved by the authors.

References

1. Sommer R, Paxson V. Outside the closed world: On using machine learning for network intrusion detection. Proceedings of the 2010 IEEE Symposium on Security and Privacy; 2010 May 16-19; Oakland, CA, USA. New York, NY: IEEE. pp. 305-316.

2. Shwayat D, Abu Al-Haija Q, Alma'aitah A. Resource-aware ML framework for multi-level cross-layer and cross-protocol attack detection in IoMT. J Supercomput. 2026; 82: 291.
3. Adadi A, Berrada M. Peeking inside the black-box: A survey on explainable artificial intelligence (XAI). IEEE Access. 2018; 6: 52138-52160.
4. Tavallaee M, Bagheri E, Lu W, Ghorbani AA. A detailed analysis of the KDD CUP 99 data set. Proceedings of the 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications; 2019 July 08-10; Ottawa, ON, Canada. New York, NY: IEEE. pp. 1-6.
5. Moustafa N, Slay J. UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). Proceedings of the 2015 Military Communications and Information Systems Conference (MilCIS); 2015 November 10-12; Canberra, ACT, Australia. New York, NY: IEEE. pp. 1-6.
6. Sharafaldin I, Lashkari AH, Ghorbani AA. Toward generating a new intrusion detection dataset and intrusion traffic characterization. Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP 2018); 2018 January 22-24; Funchal, Madeira, Portugal. Setúbal, Portugal: Science and Technology Publications. pp. 108-116.
7. Tjoa E, Guan C. A survey on explainable artificial intelligence (XAI): Toward medical XAI. IEEE Trans Neural Netw Learn Syst. 2020; 32: 4793-4813.
8. Al-Haija QA, Alsulami AA. Fast anomalous traffic detection system for secure vehicular communications. Intell Converg Netw. 2024; 5: 356-369.
9. Mukherjee B, Heberlein LT, Levitt KN. Network intrusion detection. IEEE Netw. 1994; 8: 26-41.
10. Al-Haija QA, Droos A. A comprehensive survey on deep learning-based intrusion detection systems in the Internet of Things (IoT). Expert Syst. 2025; 42: e13726.
11. Yin C, Zhu Y, Fei J, He X. A deep learning approach for intrusion detection using recurrent neural networks. IEEE Access. 2017; 5: 21954-21961.
12. Javaid A, Niyaz Q, Sun W, Alam M. A deep learning approach for network intrusion detection system. EAI Endorsed Trans Sec Saf. 2016; 3: 21-26.
13. Goodfellow I, Bengio Y, Courville A. Deep learning. Cambridge, UK: MIT Press; 2016.
14. Gunning D, Stefik M, Choi J, Miller T, Stumpf S, Yang GZ. XAI—Explainable artificial intelligence. Sci Robot. 2019; 4: eaay7120.
15. Lundberg SM, Lee SI. A unified approach to interpreting model predictions. Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017); 2017 December 04-09; Long Beach, CA, USA. Red Hook, NY: Curran Associates Inc. pp. 4768-4777.
16. Abu Al-Haija Q, Al-Tamimi A. Secure aviation control through a streamlined ADS-B perception system. Appl Syst Innov. 2024; 7: 27.
17. Zhao Q, Wang F, Wang W, Zhang T, Wu H, Ning W. Research on intrusion detection model based on improved MLP algorithm. Sci Rep. 2025; 15: 5159.
18. Ikram I, Huma Z. An explainable AI approach to intrusion detection using interpretable machine learning models. Eur Vantage J Artif Intell. 2024; 1: 57-66.
19. Mohale VZ, Obagbuwa IC. A systematic review on the integration of explainable artificial intelligence in intrusion detection systems to enhance transparency and interpretability in cybersecurity. Front Artif Intell. 2025; 8: 1526221.
20. Grabowski A, Xu S. Advancing cybersecurity practice: Explainable machine learning for network intrusion detection. J Cybersecur Educ Res Pract. 2025; 2025: 25.

21. Muhammad AE, Yow KC, Bačanin-Džakula N, Khan MA. L-XAIDS: A LIME-based explainable AI framework for intrusion detection systems. Cluster Comput. 2025; 28: 654.
22. Ahmed U, Jiangbin Z, Almogren A, Khan S, Sadiq MT, Altameem A, et al. Explainable AI-based innovative hybrid ensemble model for intrusion detection. J Cloud Comp. 2024; 13: 150.
23. Khan MF, Hassan MM. Explainable AI and machine learning models for transparent and scalable intrusion detection systems. J Inf Syst Eng Manage. 2024; 9: 1576-1588.
24. Al-Fayoumi M, Alhijawi B, Al-Haija QA, Armoush R. XAI-PhD: Fortifying trust of phishing URL detection empowered by Shapley additive explanations. Int J Online Biomed Eng. 2024; 20: 80-101.
25. Korstanje J. Gradient boosting with XGBoost and LightGBM. In: Advanced forecasting with Python. Berkeley, CA: Apress; 2021. pp. 193-205.
26. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: Machine learning in Python. J Mach Learn Res. 2011; 12: 2825-2830.
27. Abu Al-Haija Q, Alsulami AA, Alturki B, Alnabhan M. Securing the internet of flying things (IoFt): A proficient defense approach. In: Proceedings of Ninth International Congress on Information and Communication Technology. Singapore: Springer Nature; 2024. pp. 469-479.