SvedbergOpen
DISSEMINATION OF KNOWLEDGE

International Journal of Artificial Intelligence and Machine Learning

Publisher's Home Page: https://www.svedbergopen.com/

Research Paper Open Access

Evaluation of Machine Learning Classification for Intrusion Detection with Explainable AI Implementation

Alvino Kannen Vallian[1*], Rafael Sebastian Wibowo[1], Gede Putra Kusuma[1], Hidayaturrahman[2]

[1]Computer Science Department, BINUS Graduate Program - Master of Computer Science, Bina Nusantara University, Jakarta, Indonesia, 11530.
E-mail: alvino.vallian@binus.ac.id

[2]Computer Science Department, BINUS Graduate Program - Master of Computer Science, Bina Nusantara University, Jakarta, Indonesia, 11530.
E-mail: rafael.wibowo@binus.ac.id,

[3]Computer Science Department, BINUS Graduate Program - Master of Computer Science, Bina Nusantara University, Jakarta, Indonesia, 11530.
E-mail: inegara@binus.edu

[4]Computer Science Department, School of Computer Science, Bina Nusantara University, Jakarta, Indonesia, 11530.
E-mail: hidayaturrahman@binus.edu

Abstract
Classification is a machine learning objective that enables the model to properly output results based on available features. In current times, intrusion detection systems are able to be developed based on machine learning objectives, enabling them to work in conjunction by having the machine learning model able to help identify which type of intrusion set off the intrusion detection systems's alarms. With the trend of cybersecurity becoming increasingly prominent and along comes the development of explainable artificial intelligence, the aspect of transparency in machine learning-driven intrusion detection systems is highlighted due to the trend of explainable artificial intelligence. This work contributes to evaluating on how well does a machine learning model implemented with explainable artificial intelligence is able to perform and how will the output of the explainable artificial intelligence be able to increase the model's transparency. The experiment itself is done with the UNSW-NB15 dataset as a means to replicate the regular day-to-day conditions while also being a certified dataset. As a result, the algorithm best suited for the classification purpose is XGBoost with a value of 0.9767 for the accuracy metric, 0.6207 for macro F1, 0.7063 for macro Recall, 0.5947 for macro Precision, and 0.0024 for Macro FPR as a false alarm metric.

Keywords: Cybersecurity Threat, Intrusion Detection, Artificial Intelligence, Machine Learning, Model Interpretability, Explainable AI, UNSW-NB15 Dataset.

Introduction

Cybersecurity threats are to increase as time goes on with the inevitable integration of technology advancement in every segment of our lives. With the start of the exponential growth in 2021, projecting a massive amount of 623.3 million daily attacks that has increased by more than 100% from the previous year where now it reaches a projected amount at 803 million daily attacks only at the first and second quarters of 2025 (Griffith, 2025). The large amount of cyberattacks continuing to grow every year and with a broader amount attack types, increases our need for more intelligent defense mechanisms in current times.

As a means of solving this problem, machine learning–based intrusion detection systems have emerged as an effective alternative by using data-driven approaches to learn discriminative patterns from large-scale network traffic data (Magdy et al., 2023). Supervised classification models, particularly ensemble and tree-based algorithms, have demonstrated strong detection performance and robustness when applied to complex intrusion detection datasets (Ferrag et al., 2020). Despite their effectiveness, many high-performance machine learning models lack transparency, making it difficult to understand how the input features influence the classification results (Arrieta et al., 2020).

The lack of interpretability presents a significant challenge in cybersecurity applications, where trust, accountability, and human supervision are essential for operational deployment (Arrieta et al., 2020). Security analysts require clear explanations of model decisions to validate alerts, prioritize responses, and ensure compliance with regulatory and organizational policies (Vilone & Longo, 2020). Leading to making Explainable Artificial Intelligence (XAI) becoming an increasingly important research area in recent times. The main reason

for that is that XAI itself is an alternative for researchers or developers that are aiming to improve the transparency and interpretability of machine learning models without sacrificing the performance of the model (Vilone & Longo, 2020).

Among the most widely adopted XAI techniques are SHapley Additive exPlanations (SHAP) and Local Interpretable Model-agnostic Explanations (LIME), both of which are model-agnostic and suitable for complex classifiers (Mane & Rao, 2021). SHAP provides both global and local explanations by quantifying the contribution of each feature to a model's objective whether it is classification or prediction based on cooperative game theory principles (Mane & Rao, 2021). LIME focuses on local interpretability by approximating a complex model with an interpretable surrogate model around a specific instance, enabling instance-level explanation of predictions (Michael & Isreal, 2025).

The effectiveness of machine learning–based systems are highly dependent on the quality and representativeness of the datasets used for training and evaluation. The UNSW-NB15 dataset remains one of the most widely used benchmark datasets for intrusion detection research due to its realistic traffic generation and diverse attack categories (Al-Omari & Al-Haija, 2024). It includes modern attack types such as Exploits, Fuzzers, DoS, Reconnaissance, and Generic attacks, along with rich flow-based features extracted using contemporary network monitoring tools (Kabir et al., 2022).

This research employs four widely used machine learning algorithms—LightGBM, XGBoost, Random Forest, and Artificial Neural Networks (ANN) to accompany the usage of the listed XAI models. Gradient boosting models, such as LightGBM and XGBoost, are particularly effective for IDS tasks because of their ability to model complex nonlinear relationships, handle high-dimensional tabular data, and provide competitive performance with efficient training times (Attia et al., 2020; Geng et al., 2024).

By employing a set of classifiers that consists of two gradient boosting models, one ensemble bagging model, and one deep learning method, this study aims to conduct a thorough comparative evaluation. Furthermore, integrating SHAP and LIME enables transparent interpretation of their decision-making processes, supporting informed deployment in operational cybersecurity environments.

Therefore, this work focuses on machine learning–based classification using the UNSW-NB15 dataset, accompanied by the implementation of SHAP and LIME for explainable intrusion detection. The objectives of this study are to evaluate the performance of selected machine learning classifiers, analyze model decisions using SHAP and LIME, and assess how explainability enhances the transparency, reliability, and trustworthiness of intrusion detection systems. By combining strong predictive performance with interpretable explanations, this research aims to contribute to the development of explainable and operationally viable machine learning–based cybersecurity solutions.

Literature Review

Mallampati and Seetha conducted a research on enhancing intrusion detection performance by utilizing a hybrid machine learning and explainable AI framework applied to the UNSW-NB15 dataset (Mallampati & Seetha, 2024). Their work is based on handling class imbalance using the capabilities of k-Means SMOTE and proposed a hybrid feature selection pipeline combining Pearson Correlation, Information Gain, and Sequential Forward Feature Selection with LightGBM as the wrapper classifier. The model itself achieved 90.71% accuracy on the UNSW-NB15 dataset and implemented SHAP to assist them in interpreting feature contributions, enabling them to have better understanding of the learned model. Their study highlighted how XAI can improve trust and interpretability by clarifying the reasoning behind predictions in IDS applications based on their available features.

Hermosilla et al. did a research on the forensic relevance of integrating XAI methods consisting of both SHAP and LIME into intrusion detection workflows using the UNSW-NB15 dataset (Hermosilla et al., 2025a). Their work compared two classification models that are XGBoost and TabNet. Using them, they assessed explanation fidelity, stability, and global coherence. XGBoost achieved 97.8% validation accuracy and produced explanations to help better communicate their work. In their work, they presented that both XAI methods, SHAP and LIME, provide complementary interpretability signals, with SHAP offering information on feature attributions while LIME provides localized justification suitable for their purposes, forensic auditing. Their findings helped reinforced that XAI is vital for ensuring the forensic evidence and reasoning of IDS outputs.

Hermosilla et al. did a research on how effective is XAI in their previous work on the forensic relevance of XAI integration on models for intrusion detection that utilizes both SHAP and LIME to XGBoost and TabNet trained on UNSW-NB15 (Hermosilla et al., 2025b). Their study addressed the "black-box" nature of deep learning models by evaluating the consistency and transparency of explanations in forensic contexts. During their process of study, they concluded that tree-based models yield more better explanation patterns in terms of

both stability and interpratibility compared to deep tabular networks. Their work highlighted the importance of explainability in legal and investigative settings, and arrived in the result of where suggesting that SHAP and LIME enables both the researcher and user to a clearer interpretation of reasoning processes behind ML-based intrusion detection systems.

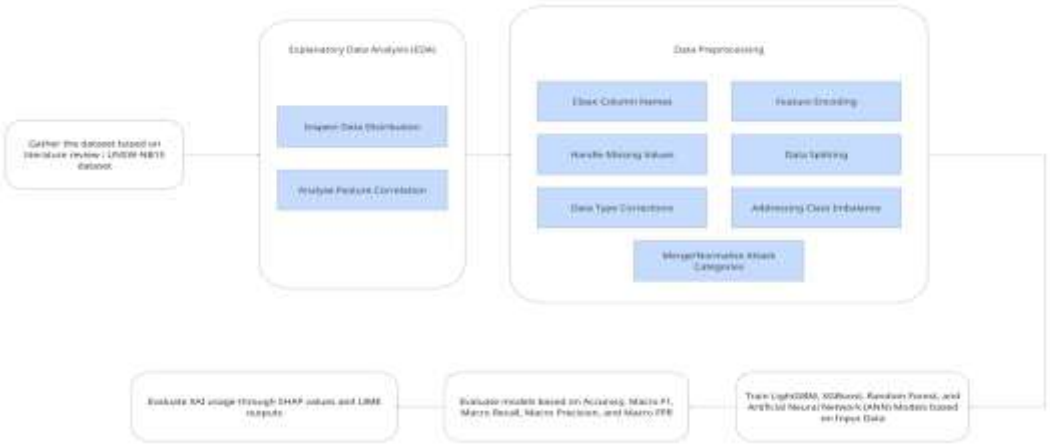
Rehman et al. conducted a research on designing an advanced explainable intrusion detection framework that integrates a more sophisticated preprocessing utilizing two processes that are the Harris Hawks Optimization and stacked ensemble classifiers while also evaluating them on the UNSW-NB15 dataset (Rehman et al., 2025). Their pipeline used three methods of preprocessing that are RobustScaler, QuantileTransformer, and SMOTE to mitigate skewed distributions and class imbalance. Afterwards they are to apply a meta-heuristic feature reduction and ensemble learning by using XGBoost, Support Vector Machine (SVM), and Random Forest. Their model achieved an accuracy of 99.44% on UNSW-NB15. Although not centered on SHAP or LIME, the study itself incorporated both XAI techniques to provide global and local explanation visualizations, enabling deployment in environments requiring transparent and accountable decision-making.

Mohale and Obagbuwa conducted research on integrating several XAI techniques consisting of SHAP, LIME, and ELI5 into machine-learning-based intrusion detection models trained on the UNSW-NB15 dataset (Mohale & Obagbuwa, 2025). Their work evaluated several models including XGBoost, CatBoost, Random Forest, Multilayer Perceptron (MLP), and Logistic Regression. During their work, it is highlighted that both XGBoost and CatBoost achieved a similar and the higher accuracy of all the models used around the 87% mark. The research itself emphasized that XAI methods not only improve interpretability but also help analysts identify indicators of malicious activity. In general, their work demonstrated the advantages of combining predictive performance with transparent interpretability in the operational domain.

Across the reviewed literature, a constant variable appears as a standard for research on ML-based intrusion detection system can be seen. The variable itself is the UNSW-NB15 dataset. From the review, this dataset can be seen as a widely adopted benchmark for evaluating intrusion detection systems due to its diverse attack types and realistic traffic characteristics. Machine learning models that are ranging from both gradient boosting architectures to deep tabular networks are able to consistently achieve strong performance in evaluating the dataset itself. On the other hand, a lacking aspect comes from their transparency. This lack in transparency increases the need on the integration of explainable AI. Both SHAP and LIME, as the commonly used XAI implementations based on the related works, repeatedly appear as essential tools for interpreting predictions. Both methods are used in validating model behavior, while also assisting end users such as analysts in their operational decision-making. To conclude the whole literature review, the studies are able to showcase that combining robust ML classification with XAI significantly enhances reliability, accountability, and transparency in intrusion detection systems.

Methodology

Figure 1. Methodology Flowchart



3.1Dataset

The UNSW-NB15 dataset is a dataset that is free to use for academic purposes. The dataset itself contains data in the form of raw network packets that is created by the IXIA PerfectStorm tool from the Cyber Range Lab in UNSW Canberra. The utilization of tool itself is done through generating a combination of both real usual

network activity and synthetically made contemporary attack behaviors to be present. Additional usage of the Tcpdump tool is to capture a sum of 100 gigabytes (GB) of raw network traffic. The dataset itself has nine branches of attack present: Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode and Worms. To further increase the details present in the dataset, the authors utilize The Argus, Bro-IDS tools to develop twelve algorithms. The developed algorithms generate a total of 49 features along with the class label present in the UNSW-NB15_freatures.csv file. All the work present in the dataset can be credited to the authors of the dataset and UNSW Canberra (Moustafa & Slay, 2015, 2016).

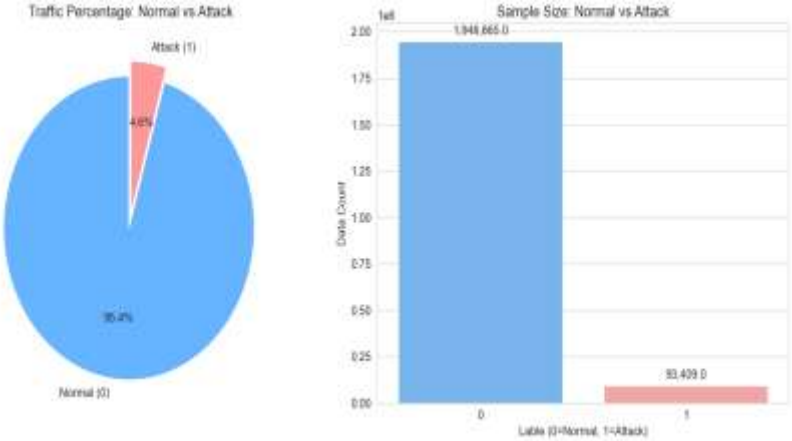
3.2Explanatory Data Analysis

During the explanatory data analysis (EDA) phase of the dataset, this work conduct EDA to understand the underlying structure of the UNSW-NB15 dataset. This process involves visualizing class distribution to identify potential imbalance dataset and analyzeing feature correlations.

3.2.1 Target Class Distribution

The distribution and balance of the dataset will play a critical factor to the model's performance later on. This work visualize the distribution of the primary target which is normal category and attack category to assess the level of imbalance between these two.

Figure 2. Target Class Distribution



As illustrated in Figure 2, the dataset has a severe class imbalance. The normal traffic dominates the dataset by taking 95% portion of the dataset, while attack traffic only takes about 5% portion of the dataset. This large difference between the two classes number suggests that the model may be biased to the majority class if not handled correctly.

3.2.2 Attack Category Distribution

To gain deeper insight into the attack class, this work further analyzed the attack class distribution show on Figure 3.

Figure 3. Attack Category Distribution

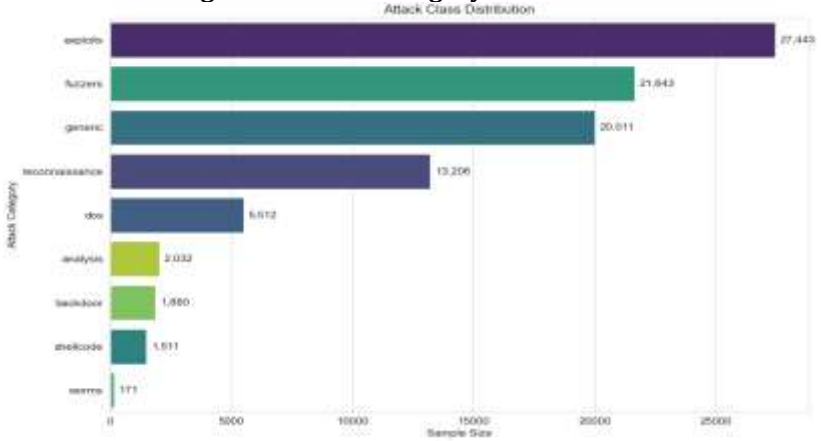


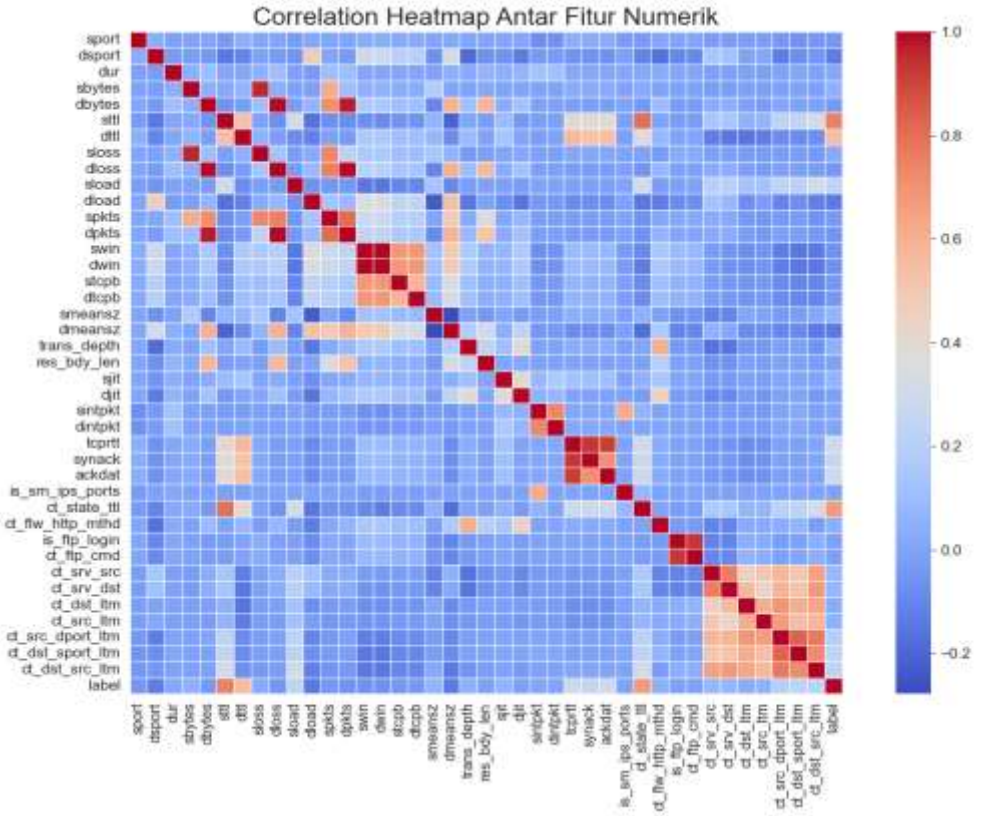
Figure 3 illustrates the attack category distribution which is also imbalanced. The attack category are dominated by exploits and worms attack category is extremely rare compared to the other attack category. As

worms has a very small sample size, this may become a challenge for the model to learn and classify the worm attack category effectively.

3.2.3 Feature Correlation Analysis

A correlation matrix shown by Figure 4 will be generated to understand the relationship between each features where the dark red indicates a positive relation and dark blue indicates a negative relation.

Figure 4. Feature Correlation Matrix



The heatmap reveals some clusters starting from the positive relation of the traffic volume where spkts (source packets) will increase alongside dpkts (destination packets), sbytes (source bytes), and dbytes (destination bytes). There is also a strong relation between tcprtt, synack, and ackdat because of a mathematical correlation where tcprtt is the sum of synack and ackdat. There is also a positive correlation connection counter cluster (ct) in the down-right part of the matrix where it contains the connection of the same addresses in.

The strategic decision based on the analysis is to keep using all the features even though there are strong relation in the matrix. This work use all the features because of the model selection used are dominated by tree-based model such as XGBoost, LightGBM, and Random Forest. Even though, this work also use ANN, to get an ideal and valid performance benchmark, all the models will be trained using the same dataset.

3.3Data Preprocessing

During the data preprocessing phase of the dataset, this work employs several methods of preprocessing to apply to the UNSW-NB15 dataset. Among them are: data cleaning, data transformation, addressing class imbalance, and data encoding (Fan et al., 2021). This step is done due to increase the efficiency of model training and to further prepare the data for the purpose of maximizing the results of model training.

3.3.1 Data Cleaning

With the first data preprocessing method, data cleaning, the dataset itself is to be cleared of missing values and inconsistent feature naming. In this case, the method used for normalizing the inconsistent features present in the attack categories column and inconsistent data typing is manual mapping, this method helps ensure the efficiency of model training due to having less columns with the same purpose contents as others while also ensuring the present data stays compatible with their type. Through the fruits of data cleaning, the dataset itself can be clear of minor inconveniences such as incomplete values, wrong data types, and inconsistent column naming.

3.3.2 Data Transformation

The second data preprocessing method to be employed to this work is data transformation. This method is employed to make the model to better understand the inputted data. In this segment is where the dataset undergoes feature encoding. The type of encoding applied in this work is label encoding. Label encoding is considered one of the most straightforward attempts at encoding, where it utilizes the boolean method of true or false (Bolikulov et al., 2024). Label encoding is often used in classification objectives to encode the target column where the results only have 2 outcomes.

In addition to that, both normalizing attack categories and class imbalance is handled with their own respective methods. While on the other hand, handling class imbalance for this work is done through the addition of SMOTE. SMOTE is applied due to the dataset itself being imbalanced in its target column. With the usage of SMOTE. that is to generate synthetic data, this preprocessing method is utilized to reduce bias during model training (Kivrak et al., 2024).

3.3.3 Data Splitting

The method of data splitting applied in the work is the stratified splitting method. Stratified data splitting is a sampling method designed to preserve the original class distribution of a dataset across the training, validation, and test partitions, ensuring that each subset reflects the same proportion of labels as the full dataset (Sechidis et al., 2011). The downside of this splitting method is where in classification problems with imbalanced datasets, random splitting may inadvertently place too many rare-class samples in a single subset, resulting in biased model training and misleading evaluation.

In this work, the implementation of stratification is done in the template of 70% train set, 15% validation set, and 15% test set. The reasoning behind the split is due to maintaining a proportional distribution of each class across all subsets while also simultaneously optimizing the balance between model learning, hyperparameter tuning, and achieving an unbiased performance evaluation.

In this scheme, 70% of the data is presented to the training set to ensure that the model that is later used will be exposed to a sufficiently large and representative sample of each class. The reason behind this is that it is especially important in imbalanced classification tasks where minority classes may already be underrepresented. The 15% validation set provides a substantial yet not excessive subset for hyperparameter optimization, early stopping, and model selection, while still also still preserving class distribution through stratification to prevent distorted feedback during tuning. Lastly, the remaining 15% will be used as a test set that serves as an independent holdout sample for estimating the model's generalization performance under the same class proportions found in the original dataset. The test set helps in ensuring that the result of the evaluation metrics remain reliable and unbiased.

3.4Machine Learning Models

3.4.1 LightGBM

LightGBM is a gradient boosting framework designed for high-performance tree-based learning. The algorithm is effective for large-scale and high-dimensional tabular datasets. Different from the traditional gradient boosting decision tree (GBDT) algorithms that grow trees level-wise, LightGBM uses a different approach where it adopts a leaf-wise growth strategy. This approach is done on how splits are made on the leaf where it provides the maximum reduction in loss. The algorithm itself also has the characteristic to prevent overfitting in the model. LightGBM incorporates hyperparameters such as num_leaves, max_depth, and min_data_in_leaf and is able to be further tuned help the model better understand the dataset (Ke et al., 2017).

3.4.2 XGBoost

XGBoost is an optimized distributed gradient boosting algorithm built on decision trees. It employs second-order optimization, explicitly using both first and second-order derivatives to refine its loss minimization process. This function helps enable the algorithm by making it more stable and precise in learning compared to using a first-order gradient boosting. The algorithm works by constructing trees level-wise and employs advanced system optimization techniques such as sparsity-aware split finding, weighted quantile sketching for approximate split points, and parallelized tree construction (Chen & Guestrin, 2016).

In this work, several parameters found in the algorithm such as eta, max_depth, subsample, and colsample_bytree are further tuned to optimize model performance across the dataset's diverse features. XGBoost also implements regularization terms (in the form of L1 and L2 penalties) directly within the objective function, controlling model complexity and promoting generalization on heterogeneous traffic patterns observed in the UNSW-NB15 dataset.

3.4.3 Random Forest

Random Forest (RF) is an ensemble learning algorithm that constructs a multitude of decision trees using bagging method and random feature selection. Each generated tree is trained on a bootstrap sample of the

original dataset, and at each node, only a random subset of features is considered for splitting. This process implements a form of diversity into the ensemble and significantly reduces variance, reducing risk of overfitting. The prediction is computed through majority voting across all trees, ensuring stability even when handling noisy or redundant features in the UNSW-NB15 dataset. Additionally, Random Forest provides internal measures of feature importance, which will later complement the XAI techniques applied later in this work. Due to its non-parametric nature and overall strong performance across classification applications, RF is able to serve as an essential baseline model (Breiman, 2001).

3.4.4 Artificial Neural Network

The Artificial Neural Network implemented in this work is a Multi-Layer Perceptron (MLP) architecture composed of an input layer, one or more hidden layers with nonlinear activation functions, and an output layer configured for multi-class classification. MLPs approximate complex decision boundaries by learning hierarchical representations of input features through iterative optimization using backpropagation and gradient-based updates. When applied to the UNSW-NB15, regularization strategies, including dropout, L2 weight decay, and early stopping, are applied to prevent overfitting, given the high dimensionality of the dataset (Heaton, 2018).

3.5Metrics for Evaluation

In this work, model performance evaluation on the UNSW-NB15 dataset is done through several metrics that are Accuracy, Macro F1, Macro Precision, Macro Recall, and Macro False Positive Rate (FPR). These metrics are selected to provide a balanced evaluation under the imbalanced class distribution characteristic of the UNSW-NB15 dataset.

3.5.1 Accuracy

Accuracy measures the proportion of correctly classified instances relative to the total number of samples.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad \textbf{(1)}$$

As seen on Equation 1, where TP and TN denote true positives and true negatives, and both FP and FN denote false positives and false negatives respectively. Although the accuracy metric is widely used due to its intuitive interpretability, accuracy becomes unreliable in imbalanced datasets such as the UNSW-NB15 dataest. This can happen due to the class imbalance factor, where dominant classes can inflate the value even though it has poor performance on minority attack categories (Powers, 2020).

3.5.2 Macro Precision

The macro Precision metric evaluates the ability of a classification model to avoid false positives per class, and then it being averaged equally across all classes regardless of class frequency.

$$\text{Macro Precision} = \frac{1}{\text{K}}\sum_{\text{i}=1}^{\text{K}} \frac{\text{TP}_\text{i}}{\text{TP}_\text{i} + \text{FP}_\text{i}} \quad \textbf{(2)}$$

The macro Precision metric is essential in IDS because false alarms (FP) incur operational costs. By giving equal weight to minority classes, this metric ensures the model does not achieve an unrealistic high score due to overfitting (Sokolova & Lapalme, 2009).

3.5.3 Macro Recall

The macro Recall metric measures the proportion of attack types correctly detected. High Macro Recall is indispensable in intrusion detection where failing to detect attacks (FN) can result in catastrophic security breaches. Since UNSW-NB15 includes rare attacks such as Shellcode, Worm, or Backdoor, Macro Recall ensures each attack category contributes equally to the evaluation (Powers, 2020).

$$\text{Macro Recall} = \frac{1}{\text{K}}\sum_{\text{i}=1}^{\text{K}} \frac{\text{TP}_\text{i}}{\text{TP}_\text{i} + \text{FN}_\text{i}} \quad \textbf{(3)}$$

3.5.4 Macro F1

The macro F1 Score is the mean of both the macro Precision and macro Recall, computed per class and then averaged again (Vechtomova, 2009).

$$\text{Macro F1} = \frac{1}{\text{K}}\sum_{\text{i}=1}^{\text{K}} 2 \cdot \frac{\text{Precision}_\text{i} \cdot \text{Recall}_\text{i}}{\text{Precision}_\text{i} + \text{Recall}_\text{i}} \quad \textbf{(4)}$$

The average of both macro Precision and macro Recall heavily weighs extreme imbalances between them, making macro F1 an optimal metric when assessing cybersecurity models such as IDS that must balance false alarms (FP) with missed detections (FN).

3.5.5 Macro FPR

Macro False Positive Rate calculates the behavior of the model to incorrectly label normal everyday traffic as attacks. To find the macro FPR of a model, its initial FPR needs to found first (Chicco & Jurman, 2020).

$$\mathrm{FPR_i} = \frac{\mathrm{FP_i}}{\mathrm{FP_i} + \mathrm{TN_i}} \quad \textbf{(5)}$$

Using the FPR found on Equation 5, the Macro FPR averages this value across all classes using Equation 6.

$$\text{Macro FPR} = \frac{1}{\mathrm{K}} \sum_{\mathrm{i=1}}^{\mathrm{K}} \mathrm{F\,PR_i} \quad \textbf{(6)}$$

In the context of an IDS, a low FPR score is considered vital. Large sum of false alarms overwhelm security analysts, decrease trust in the IDS, and increase operational costs. Macro FPR's averaging through FPR scores, help ensures rare attack types do not distort the assessment of false alarm sensitivity in a system-wide manner.

Result and Discussion

4.1Train and Validation Results

This section represents the model's performance during training and validation process using the optimal hyperparameter configuration.

4.1.1 Final Model Configuration

The four models used in this work were trained using the best parameter combination obtained from the previous optimization stage. Table 1 displays the final configuration used to generate this training models.

Table 1. Model Hyperparameters

Model	Parameter	Value
XGBoost	n_estimators	2824
	max_depth	11
	learning_rate	0.0757
	min_child_weight	3
	gamma	2.0116
	subsample	0.9900
	colsample_bytree	0.7189
	reg_alpha	1.4136
	reg_lambda	1.2082
LightGBM	n_estimators	1971
	learning_rate	0.0442
	max_depth	12
	num_leaves	41
	min_split_gain	0.3542
	min_child_samples	43
	subsample	0.6465
	subsample_freq	5
	colsample_bytree	0.9846
	reg_alpha	2.2374
	reg_lambda	2.4847
Random Forest	n_estimators	297
	max_depth	20
	min_samples_split	6
	min_samples_leaf	3
	max_features	sqrt
ANN	n_layers	2
	n_units_l0	77
	n_units_l1	54
	activation	relu
	alpha	0.0000
	learning_rate_init	0.0036

4.1.2 Model Stability Evaluation

To ensure the model does not suffer extreme overfitting, a stability evaluation was performed by comparing the model's training and validation performance. Table 2 displays the model's performance in training and validation.

Table 2. Training and Validation Set Performance

Model	Train F1	Val F1	Gap	Status
XGBoost	0.8790	0.6498	0.2292	Overfitting Risk
LightGBM	0.8705	0.6513	0.2193	Overfitting Risk
Random Forest	0.8892	0.6107	0.2785	Overfitting Risk
ANN	0.7986	0.5739	0.2247	Overfitting Risk

Based on the performance metrics observed in the validation phase, several critical insights regarding the model's behavior on imbalanced data were identified. A significant gap in train and validation F1 score was found in all of the models with a gap around 0.2 where the validation F1 score is lower than the train F1 score. The models achieved a high train F1 score by correctly predicting the majority attack class and the lower validation F1 score actually represents the actual difficulty in predicting the minority attack class in the validation set.

In terms of generalization stability, the tree-based ensemble models such as XGBoost and LightGBM outperformed the ANN model. The ANN has the lowest train and validation F1 score, showing that the neural network architecture struggles more in transiting from synthetic training data using SMOTE to real-word attack scenario data in the validation data.

4.2Testing Results

This section represents the model's performance using the **Testing Data**, which consists of unseen data during training and validation process. This phase aims to measure the system's performance in real-work attack scenario.

4.2.1 Model Testing Performance

Table 3 represents the model's performance in testing dataset.

Table 3. Testing Set Performance

Model	Accuracy	Macro F1	Macro Recall	Macro Precision	Macro FPR (False Alarm)
XGBoost	0.9767	0.6207	0.7063	0.5947	0.0024
LightGBM	0.9764	0.6127	0.6965	0.5863	0.0024
Random Forest	0.9762	0.5938	0.7040	0.5620	0.0024
ANN	0.9734	0.5650	0.6930	0.5271	0.0027

Based on the testing results, all models achieved a very high accuracy score of around 0.97. However, high accuracy in this results is misleading as there are a high class imbalance, where the majority class is significantly higher than the minority class. Therefore, a model can achieve a high accuracy by correctly predicting the majority class while failing to predict the minority class.

As the result, this work prioritize Macro F1 Score and Macro Recall Score as the model's performance indicator over Accuracy. These metrics treat all classes equally providing a more honest assessment of the model's performance in predicting the minority class. By evaluating this metrics, the best model is **XGBoost** with a Macro F1 Score of 0.6207 and a Macro Recall of 0.7063. This shows that **XGBoost** is the best model in recognizing minority attacks which is a crucial aspect in intrusion detection system.

4.3XAI Results

To ensure trust and transparency, explainable AI or XAI is used to interpret the model's decision-making process. This work implements ooth SHAP and LIME to reveal the key features in the intrusion detection system.

4.3.1 Global Feature Importance (SHAP)

The SHAP summary plot in Figure 5 shows the features ranked by their impact on the model's output across all classes.

Figure 5. Global Feature Importance

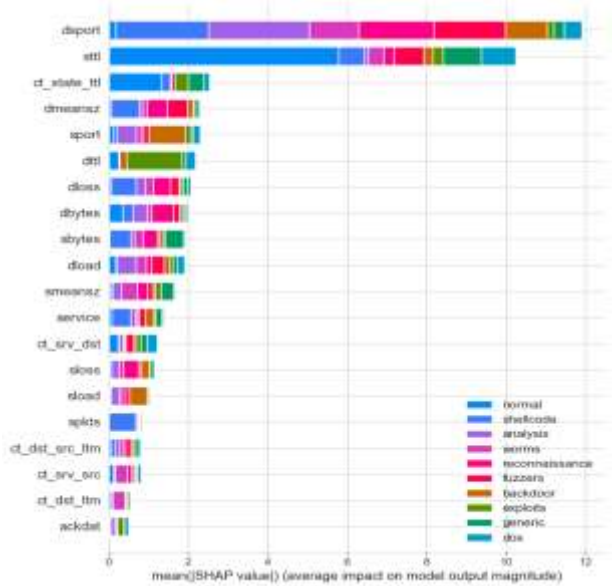


Figure 5 shows that the top dominant feature is dsport (destination port number), which is ranked as the most influential predictor globally. The multi-colored bar shows that dsport is critical for distinguishing between multiple attack types. The second top dominant feature is sttl (source to destination time to live), this feature has a large impact on the normal class as it has the largest portion of the bar. This shows that the model relies heavily on sttl values to validate benign traffic.

4.3.2 Feature Impact Analysis (SHAP)

To better understand the directional relationship between the features and the model prediction, beeswarm plot is used as shown in Figure 6 specifically for the normal class. This visualization displays how specific feature affect the probability of classifying normal traffic.

Figure 6. Feature Impact Analysis (Normal Class)

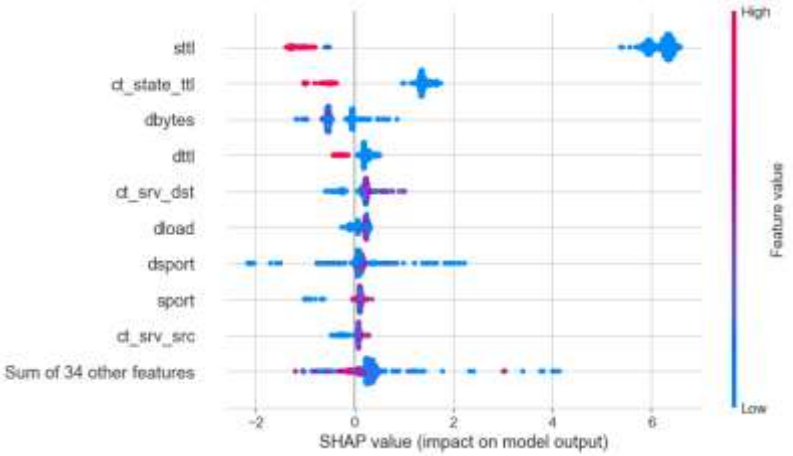


Figure 6 shows that ttl (Time to Live) such as sttl, ct_state_ttl, and dttl are the most decisive features as it has the farthest separation where low value (blue dot) are clustered heavily on the positive side and the high value (red dot) on the negative side. This shows that the model learned a strong rule where traffic with low ttl is most likely to be categorized as normal.

4.3.3 Local Interpretation (LIME)

To validate the global feature interpretation and feature impact analysis by SHAP, LIME was implemented to interpret the model decision-making process for individual data. Figure 7 shows the LIME interpretation for a specific test data (Sample 10) which was classified as normal.

Figure 7. LIME Local Interpretation (Normal Class)

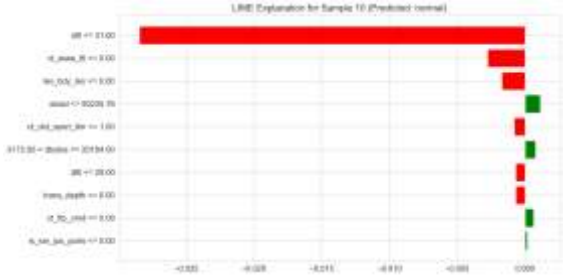


Figure 7 justifies SHAP's interpretation as the ttl in Sample 10 are all on the negative side which means they have a low value. This aligns with Figure 6 as it shows that normal class usually has low ttl values. The same interpretation between LIME and SHAP proves that the model does not rely on random noise and really learned to classify the attack category correctly.

Conclusion and Future Works

This work has succefully implemented and evaluated four machine learning models XGBoost, LightGBM, Random Forest, and ANN on the UNSW-NB15 dataset to detect intrusion, integrated with explainable AI (XAI) technique to further interpret the model decision-making process.

The model evaluation results conclude that while all the models achieved a high accuracy, this metric proved to be misleading as the data is severely imbalanced. As result, the metrics chosen to evaluate the models performance are macro F1-score and macro Recall as a more reliable metrics.

By evaluating the macro F1-score and macro Recall score, the model has a decreased in the model's performance from training, validation, and testing dataset. From the scores of the four models, the best model is XGBoost with a macro F1-score of 0.6207 and macro Recall score of 0.7063. These scores shows that XGBoost can detect minority class better than the other model eventhough by a small margin. In the intrusion detection system this is critical as wrong classification of attack can lead to major problems.

The integration of XAI (SHAP and LIME) also provide essential transparency and interpret of the model result and decision-making process. SHAP global interpretation identified that dsport (destination port) and sttl (source to destination time to live) as the most influential feature. Furthermore, the LIME local interpretation successfully validate that the model predict and decision making is valid and not based on some random noise. Specifically, LIME analysis revealed that low sttl value is a strong indicator of a normal traffic.

Despite these achievements, there are several thing that can be improved for future research. First, future research can handle imbalanced dataset better to further reduce the difference between training and validation performance. Second, future research may investigate deep feature selection that might help to reduce overfitting in the model. Third, future research may try to do hyperparameter tuning to specifically targeting the macro F1-score and macro Recall score to further boosts the detection of rare attack category such as worms.

References

1. Al-Omari, M., & Al-Haija, Q. A. (2024). Performance analysis of machine learning-based intrusion detection with hybrid feature selection. Computer Systems Science and Engineering, 48(6), 1537–1555. https://doi.org/10.32604/csse.2024.056257
2. Arrieta, A. B., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., Garcia, S., Gil-Lopez, S., Molina, D., Benjamins, R., Chatila, R., & Herrera, F. (2020). Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. Information Fusion, 58, 82–115. https://doi.org/10.1016/j.inffus.2019.12.012
3. Attia, A., Faezipour, M., & Abuzneid, A. (2020). Network intrusion detection with xgboost and Deep learning algorithms: An evaluation study. 2020 International Conference on Computational Science and Computational Intelligence (CSCI), 138–143. https://doi.org/10.1109/csci51800.2020.00031
4. Bolikulov, F., Nasimov, R., Rashidov, A., Akhmedov, F., & Cho, Y.-I. (2024). Effective methods of categorical data encoding for artificial intelligence algorithms. Mathematics, 12(16), 2553. https://doi.org/10.3390/math12162553
5. Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5–32. https://doi.org/10.1023/a:1010933404324

6. Chen, T., & Guestrin, C. (2016). XGBoost. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 785–794. https://doi.org/10.1145/2939672.2939785
7. Chicco, D., & Jurman, G. (2020). The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. BMC Genomics, 21(1). https://doi.org/10.1186/s12864-019-6413-7
8. Fan, C., Chen, M., Wang, X., Wang, J., & Huang, B. (2021). A review on data preprocessing techniques toward efficient and reliable knowledge discovery from building operational data. Frontiers in Energy Research, 9. https://doi.org/10.3389/fenrg.2021.652801
9. Ferrag, M. A., Maglaras, L., Moschoyiannis, S., & Janicke, H. (2020). Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study. Journal of Information Security and Applications, 50, 102419. https://doi.org/10.1016/j.jisa.2019.102419
10. Geng, Y., Hu, H., Ge, Z., & Lian, Z. (2024). Network intrusion detection algorithm based on LIGHTGBM model and improved particle swarm optimization. 2024 IEEE Cyber Science and Technology Congress (CyberSciTech), 67–74. https://doi.org/10.1109/cyberscitech64112.2024.00021
11. Griffith, Charles. 2025. The Latest 2025 Ransomware Statistics (updated June 2025). https://aag-it.com/the-latest-ransomware-statistics/
12. Heaton, J. (2017). Ian Goodfellow, Yoshua Bengio, and Aaron Courville: Deep learning. Genetic Programming and Evolvable Machines, 19(1–2), 305–307. https://doi.org/10.1007/s10710-017-9314-z
13. Hermosilla, P., Berríos, S., & Allende-Cid, H. (2025a). Explainable AI for forensic analysis: A comparative study of shap and lime in intrusion detection models. Applied Sciences, 15(13), 7329. https://doi.org/10.3390/app15137329
14. Hermosilla, P., Díaz, M., Berríos, S., & Allende-Cid, H. (2025). Use of explainable artificial intelligence for analyzing and explaining intrusion detection systems. Computers, 14(5), 160. https://doi.org/10.3390/computers14050160
15. Kabir, M. H., Rajib, M. S., Rahman, A. S., Rahman, Md. M., & Dey, S. K. (2022). Network intrusion detection using UNSW-NB15 dataset: Stacking Machine Learning Based Approach. 2022 International Conference on Advancement in Electrical and Electronic Engineering (ICAEEE), 1–6. https://doi.org/10.1109/icaeee54957.2022.9836404
16. Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T.-Y. (2017). LightGBM: A highly efficient gradient boosting decision tree. Advances in Neural Information Processing Systems, 30. https://www.researchgate.net/publication/378480234_LightGBM_A_Highly_Efficient_Gradient_Boosting_Decision_Tree
17. Kivrak, M., Avci, U., Uzun, H., & Ardic, C. (2024). The impact of the smote method on machine learning and ensemble learning performance results in addressing class imbalance in data used for predicting total testosterone deficiency in type 2 diabetes patients. Diagnostics, 14(23), 2634. https://doi.org/10.3390/diagnostics14232634
18. Magdy, M. E., Matter, A. M., Hussin, S., Hassan, D., & Elsaid, S. A. (2023). Anomaly-based Intrusion Detection System based on feature selection and majority voting. Indonesian Journal of Electrical Engineering and Computer Science, 30(3), 1699. https://doi.org/10.11591/ijeecs.v30.i3.pp1699-1706
19. Mallampati, S. B., & Seetha, H. (2024). Enhancing intrusion detection with explainable AI: A transparent approach to network security. Cybernetics and Information Technologies, 24(1), 98–117. https://doi.org/10.2478/cait-2024-0006
20. Mane, S., & Rao, D. (2021). Explaining network intrusion detection system using explainable AI framework. arXiv preprint arXiv:2103.07110. https://doi.org/10.48550/arXiv.2103.07110
21. Michael, I., & Isreal, O. (2025). Visualization and interpretability of ensemble-based intrusion decisions in SD-VANETs using SHAP and LIME.
22. Mohale, V. Z., & Obagbuwa, I. C. (2025). Evaluating machine learning-based intrusion detection systems with explainable AI: Enhancing transparency and Interpretability. Frontiers in Computer Science, 7. https://doi.org/10.3389/fcomp.2025.1520741
23. Moustafa, N., & Slay, J. (2015). UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 Network Data Set). 2015 Military Communications and Information Systems Conference (MilCIS), 1–6. https://doi.org/10.1109/milcis.2015.7348942

24. Moustafa, N., & Slay, J. (2016). The evaluation of NETWORK ANOMALY DETECTION SYSTEMS: Statistical analysis of the UNSW-NB15 Data Set and the comparison with the KDD99 data set. Information Security Journal: A Global Perspective, 25(1–3), 18–31. https://doi.org/10.1080/19393555.2015.1125974
25. Powers, D. M. W. (2020). Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation. arXiv preprint arXiv:2010.16061. https://doi.org/10.48550/arXiv.2010.16061
26. Rehman, A., Saba, T., Jamjoom, M. M., Al-Otaibi, S., & Khan, M. I. (2025). Advances in machine learning for explainable intrusion detection using imbalance datasets in cybersecurity with Harris Hawks optimization. Computers, Materials & Continua, 86(1), 1–15. https://doi.org/10.32604/cmc.2025.068958
27. Sechidis, K., Tsoumakas, G., & Vlahavas, I. (2011). On the stratification of multi-label data. Lecture Notes in Computer Science, 145–158. https://doi.org/10.1007/978-3-642-23808-6_10
28. Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. Information Processing & Management, 45(4), 427–437. https://doi.org/10.1016/j.ipm.2009.03.002
29. Vechtomova, O. (2009). Introduction to Information Retrieval: Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze (Stanford University, Yahoo! Research, and University of Stuttgart) Cambridge: Cambridge University Press, 2008, XXI+482 pp; Hardbound, ISBN 978-0-521-86571-5, $60.00. Computational Linguistics, 35(2), 307–309. https://doi.org/10.1162/coli.2009.35.2.307
30. Vilone, G., & Longo, L. (2020). Explainable artificial intelligence: A systematic review. arXiv preprint arXiv:2006.00093. https://doi.org/10.48550/arXiv.2006.00093