Buletin Ilmiah Sarjana Teknik Elektro
Vol. 8, No. 3, June 2026, pp. 913-924
ISSN: 2685-9572, DOI: 10.12928/biste.v8i3.16024

Intrusion Detection System: A Multimodal Analysis-based Machine Learning with Emphasis on Interpretability

Tabark Nasser Abdul Hussein, Ayad Hameed Mousa
College of Computer Science and Information Technology, University of Kerbala, Iraq

ARTICLE INFORMATION

Article History:

Received 25 February 2026
Revised 14 May 2026
Accepted 30 June 2026

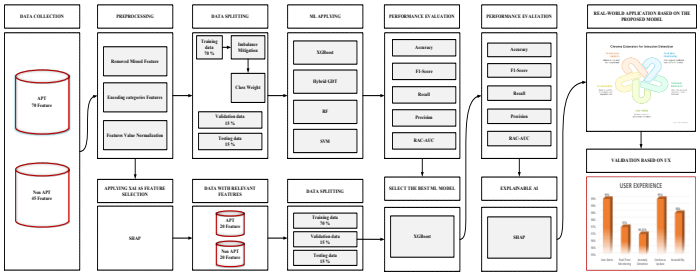
Keywords:

Intrusion Detection;
Intrusion Detection System;
Machine Learning;
Interpretability;
Multimodal Analysis

Corresponding Author:

Tabark Nasser Abdul Hussein,
College of Computer Science and Information Technology,
University of Kerbala, Iraq.
Email:
tabark.n@s.uokerbala.edu.iq

ABSTRACT



Detecting persistent threats, whether APTs or nonAPTs, is a constant challenge in the field of cybersecurity due to the multiplicity of attacks, their stealthy nature, and their multi-stage targeting of information systems over extended periods. The rigor of intrusion detection system selection is measured by its ability to detect these threats in their early stages and by the fundamental characteristics of network traffic. However, due to the large number of characteristics, some may be unrelated or of limited importance in determining the severity of malicious activity. Accordingly, selecting and defining relevant and influential characteristics for intrusion detection has become a necessity, especially in resource-constrained environments. In this paper, a set of machine learning algorithms (XGBoost, Random Forest, Support Vector Machine, Hybrid Decision Tree) was adopted in conjunction with artificial intelligence pre-interpretation (XAI) techniques to develop an intrusion detection model in a resource-constrained environment. The datasets CICAPT-IIoT, CICIoT2023, IoT-23 were used after preprocessing. XAI techniques were employed in two phases: first, during preprocessing to identify key features of the selected datasets, and second, during post-processing for interpretation. A real-world application based on the proposed model was developed to validate its accuracy and applicability in intrusion detection. Extensive testing demonstrated the superiority of the (XGBoost) algorithm with its accuracy (the CICIoT2023 dataset, achieving an F1-score of 0.9925, precision of 0.9933, recall of 0.9920, and an almost perfect ROC-AUC of 0.9999. the CICAPT-IIoT dataset scoring 99.16%, 0.9810 F1 score. And on the IoT-23 dataset accuracy 99.69% and achieved balanced precision, recall and F1-score of 0.9969). The study was distinguished by its reduction of complexity and improved performance of the proposed model.

Document Citation:

T. N. A. Hussein and A. H. Mousa, "Intrusion Detection System: A Multimodal Analysis-based Machine Learning with Emphasis on Interpretability," *Buletin Ilmiah Sarjana Teknik Elektro*, vol. 8, no. 3, pp. 913-924, 2026, DOI: 10.12928/biste.v8i3.16024.

1. INTRODUCION

The IoT is a network of integrated and programmed hardware with sensors that can communicate with devices of varying sizes [1]. The IoT is one of the technologies of the future [2]. The IoT environment has limited resources and energy, and dealing with it requires precision and care to protect it from attacks [3]. There is a pressing need for protection against attacks and vulnerabilities, and for increased security measures linking devices [4]. IDS are among the most important mechanisms for protecting this network, and this system aims to detect attacks early [5][6]. Monitoring data traffic, detecting attacks, and providing early warnings and detection [7]. The system must be able to make decisions immediately and be capable of detecting and responding to attacks [8][9].

This system has a high detection capability and supports technologies that help in accurate detection [10][11]. IDS alone is insufficient, adding ML increases detection accuracy and the ability to distinguish attacks [12]. Improving detection is essential in the IoT network to increase response speed [13][14]. Using ML is crucial for the validity of the model [15][16].

Increase the reliability of the model by making it interpretable [17]. The IDS is responsible for evaluating predictions, the consistency of results, and accuracy [18]. Use interpretation techniques that help when selecting features and providing explanations [19]. In this paper, we propose a unified and interpretable framework for ID in IoT environments [20] based on ML and evaluated across multiple classification settings [21]. The proposed framework not only detects attacks using binary and multiclass classification but also incorporates stepwise analysis of attack behavior integrating decision interpretation mechanisms to enhance transparency and understanding of model outputs. More specifically the contributions of this research are as follows:

- Developing a machine learning based ID framework and evaluating its performance across various classification settings, including binary multiclass, and stepwise classification, reflecting realistic attack scenarios in IoT environments.
- Conducting a comprehensive analysis of the performance of several ML algorithms, including XGBOOST and others [22] to identify the most balanced model in terms of detection accuracy and generalizability across different datasets [23].
- Using interpretability methods to understand model decisions, identify key features, gain insight into attack patterns, aid in decision support.
- The model was implemented practically by creating an extension and application where a link was entered, then analyzed, alerted, and interpreted to provide a classification guide.
- A comprehensive experimental evaluation was presented based on three real-world IoT datasets, using appropriate evaluation metrics to address data imbalances and analyze the results in depth.

2. RELATED WORKS

This study focused solely on predictive performance for intrusion detection systems in IoT networks and confirmed the efficacy of ensemble methods for over 98% accuracy. It was, though, lacking in interpretability, multi-scenario assessments, three- tiered detection methods, and real-world practicality.

This study [24] proposed an enhanced ID model that combines hybrid feature selection methods with XGBoost for improved efficiency of detection. Results from the experiments showed that classification accuracy was greater than 99% after dimensionality reduction, supporting better computational efficiency with minimal loss of performance. However, the study was only validated within a singular experimental context, and did not include explainable artificial intelligence/ cross-dataset robustness testing [25].

XGBoost was used in this study for feature selection on the UNSW-NB15 dataset, which showed an improvement in binary classification accuracy to 90.85% and 77.51% accuracy in multiclass detection after reducing the features from 42 to 19. While the results showed an improvement due to dimensionality reduction, it was still difficult to detect the minority classes [11]. While the authors reviewed several classifiers for multi-class intrusion detection, obtaining approximately 90% accuracy, they showed effective multi-class detection. However, the study lacked feature importance or explainable AI [26].

Tree-based classifiers have reported a binary classification accuracy of over 98%. This work assesses a number of supervised ML models for IoT intrusion detection. However, the study primarily focused on performance metrics, and did not address interpretability, or consider hierarchical detection approaches [27]. This study implemented XGBoost-based feature ranking on UNSW-NB15, achieving 90.85% accuracy for binary classification and 77.51% for multi-class. Although the study showed that the reduction of dimensions enhanced the performance of the system, the examination of the cross dataset system robustness was not performed [19]. Using the CICIDS2017 dataset, Banadaki (2020) examined XGBoost, Random Forest,

Decision Tree, and Gradient Boosting and reported the highest accuracy of 99.6% and a weighted F1-score of 97.9% obtained using XGBoost. While the overall accuracy is very high, the results showed that class imbalance affects the performance at the macro level and minority-class detection is particularly problematic [28].

Akram et al. proposed a hybrid deep learning ensemble model using MRI data for the classification of Alzheimer's disease. The model, which combines EfficientNetB7, Xception, and MobileNetV3Large, obtained 99.87% and 97.13% accuracy on the ADNI and OASIS datasets respectively, showing great generalization capability. To improve the interpretability of the model, Grad-CAM was used [29]. Verma and Ranga assessed several ML classifiers for IoT-based DoS detection using the CIDDS-001, UNSW-NB15, and NSL-KDD datasets. They found that CART had an accuracy of 96.74%, whereas XGBoost had the best AUC (98.77%). This study evaluated the statistical significance and execution time on the Raspberry Pi and found XGBoost and CART to be models that are both efficient and high-performing [2]. Banaamah and Ahmad suggested an IoT IDS using deep learning and CNN, LSTM, and GRU on the Bot-IoT dataset. LSTM outperformed all other models with an accuracy of 99.8%, precision of 99.7%, a recall of 100%, and an F1-score of 99.8%, compared to 99.7% for CNN and 99.6% for GRU [30].

Although previous works show impressive accuracy in detection, many do not include thorough cross-dataset validation, integration of interpretability, or verification of deployment. This study addresses these issues by developing an adaptive and explainable IDS framework assessed under various classification methods and real-world scenarios

3. PROPOSED METHODOLOGY

The proposed method focuses on achieving high accuracy, explainability, and generalizability to different IoT environments for intrusion detection. In addition to precise classification, the method helps to understand and justify the decisions made by the model, thus improving trust and dependability [31], Methodology as illustrated in Figure 1. This proposed framework includes seven phases: data collection, preprocessing, data splitting, ML applying, development, Performance evaluation, select the outperform model, feature selection based XAI, create relevant datasets, data splitting, applying the outperformed ML model, performance evaluation, and development of real-world application based on the proposed model. In the following sub-paragraphs, the highlighted of stage as conducted.

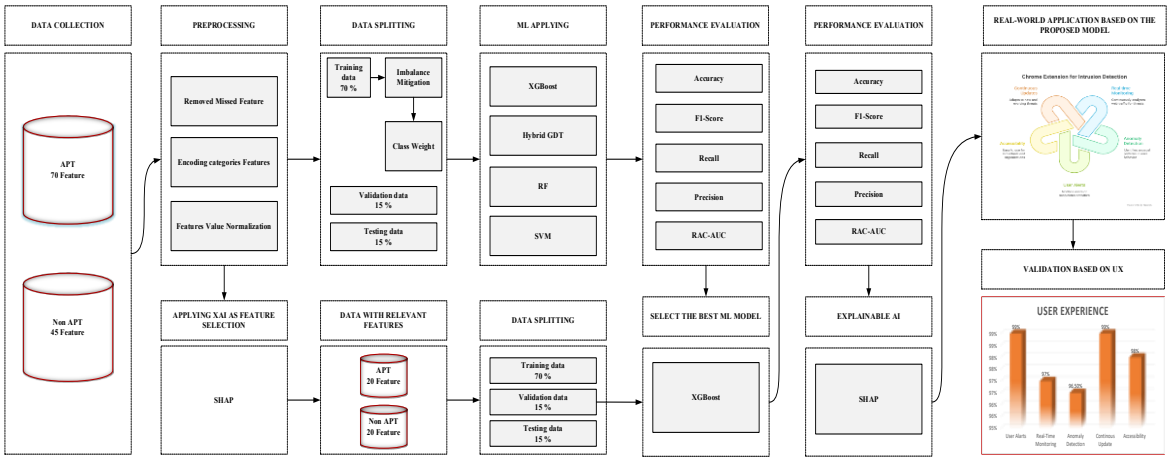


Figure 1. Overall architecture of the proposed explainable intrusion detection framework

3.1. Data Collection Phase

The datasets used in the benchmark evaluation of the suggested model are CICAPT-IIoT, CICIoT2023, and IoT-23. These datasets are chosen, considering their large volume, recent creation, and coverage for multiple attack types for both APT and NON- APT. In the context of this study, three datasets were utilized, Table 1 tabulate the brief description of each.

3.1.1. [CICAPT-IIoT]

The dataset has been developed by the Canadian Institute for Cybersecurity (CIC) for the purpose of understanding APTs in Industrial IoT (IIoT) environments. it has around 15 million records comprising of 70

network features. It presents multi-stage complex attacks which makes it suitable for the evaluation of proposed model for complex attacks. It can be accessed from (https://www.unb.ca/cic/datasets/iiot-dataset-2024.html).

3.1.2. [CICIoT2023]

This dataset contains the simulation of a smart home scenario with various IoT devices. It was created by Canadian Institute for Cybersecurity (CIC) in 2023. The original dataset has 30 million records and contains attacks like DDoS, DoS, Brute Force, and Reconnaissance. For this research, after the 7,845,673 records were processed and cleaned, they were used to enhance the quality and balance of the data. It can be accessed from (https://www.unb.ca/cic/datasets/iotdataset-2023.html).

3.1.3. [IoT-23]

The IoT-23 dataset consists of network traffic data from 23 IoT devices. The total number of records is 6,046,623 which have been labeled and have nearly 45 features for consideration. A unique feature of this dataset is the inclusion of APT and NON-APT attack scenarios along with benign traffic. It was accessed from (https://www.kaggle.com/datasets/surajsooraj26/iot-23). Please ensure that you have referenced this dataset correctly as it is the first of its kind to be created and used in IoT security research [32].

Table 1. Dataset

Id	Dataset Name	Environment	No. of Feature	Dataset Type	No. of Sample	Link
1	CICAPT-IIoT	IIoT	70 features	APT	contains normal size 15 million records	https://www.unb.ca/cic/datasets/iiot-dataset-2024.html
2	CICIoT2023	IoT	46 features	NON-APT	7,845,673 records	https://www.unb.ca/cic/datasets/iotdataset-2023.html
3	IoT-23	IoT	45 features	APT, NON-APT	6,046,623 records	https://www.kaggle.com/datasets/surajsooraj26/iot-23

3.2. Data Preprocessing

Quality data is essential for machine learning. The four steps undertaken in the data cleaning include addressing missing values, categorical data, and feature scaling. The steps involved include:

3.2.1. Missing Value Removal

The first step was to get rid of any rows with missing values. Dropna() in the pandas library was used to accomplish this. Removing instances that are incomplete makes machine learning biased. It ensures that only complete data is given to the models. The number of rows that were removed will help determine the effect of the step on the overall size of the dataset.

3.2.2. Categorical Data

All the features that were categorical were changed to numerical values because this is a requirement for machine learning models. For the nominal categories that have no order, one-hot encoding was used with pd.get_dummies. And for the ordinal categories, label encoding was used. This step ensures that the information contained in the categorical variables is preserved and, in addition, that the data accommodated algorithms like XGBoost, Random Forest, and SVM.

3.2.3. Feature Value Normalization

All numerical features were normalized to remove the impact of different feature scales[26]. Standardization (z-score normalization) was carried out using scikit-learns Standard Scaler, which rescales each feature to have zero mean and unit variance. This holds major significance for distance-based models (e.g., SVM) and gradient-based optimizations. It guarantees that no feature, due to its size, suppresses other features.These preprocessing steps have been uniformly applied to the CICAPT-IIoT, CICIoT2023, and IoT-23 datasets. This consistency is essential as it creates a strong basis for the subsequent splitting and modeling steps.

3.3. Data Splitting

Upon finishing preprocessing, each dataset was divided into three parts: training, validation, and testing. A split ratio of 70/15/15 is adopted, where 70% of the data is utilized to train the model, 15% is used for validation (hyperparameter tuning and early stopping), and the remaining 15% is for final evaluation.

Stratified sampling was used based on the target labels for the original class distribution to avoid bias. This method is important for the divided dataset for intrusion detection problems since it ensures each of the divided datasets is consistent with the class distribution of the full dataset.

During training, class imbalance was tackled with the use of class weighting, which means that a higher penalty is assigned to the misclassification of minority class instances. This method improves the model's sensitivity without altering the natural distribution of data. This is preferred to oversampling or undersampling, since those methods would introduce synthetic/ artificial data or lose important data.

3.4. Machine Learning Models

3.4.1. XGBoost

XGBoost, based on decision trees, is an efficient and scalable algorithm based on the principles of gradient boosting and widely used in research and practice with a relatively high level of confidence in the ability to cope with the problems of highly unbalanced data and datasets with high dimensionality and therefore in this research, it has been used as the first step of the classifier for detection of APT, both stage-wise and comprehensive [33]. In the dataset, a stratification of 70/15/15 was used to train, validate, and test the model and maintain the same class distribution across all three datasets; the model was trained on features of the network flow that had been numerically encoded as well as on tokenized. Hyperparameter tuning was performed on the validation set with respect to the learning rate (shrinkage), maximum depth of the tree, number of iterations of the tree, and subsampling of the trees.

XGBoost uses a range of its own built-in regularization techniques (which include both column and row level subsampling, depth control, and the application of L1 and L2 regularization on the weights of the leaves), as well as the regularization in the case of overfitting; and in the situation where the validation loss was no longer improving, the early stopping was set. The model remained stabilized across all the stages of APT with no signs of performance inflation and along with the captured complex non-linear interactions of the features; it was evident that the second-order gradient optimization had been used.

3.4.2. Random Forest

RF is a crowdsourced decision tree learning algorithm that creates a set of independent models and combines their outputs to improve stability and reduce variance. This algorithm demonstrates high efficiency in handling high dimensional and unbalanced data, as well as its ability to represent nonlinear relationships, making it particularly suitable for intrusion detection applications in IoT environments [34][35].

As a baseline ensemble model, Random Forest was trained on stratified data. To address variance, the model's authors used bootstrap aggregation and feature randomness while controls on depth and minimum sample sizes dealt with overfitting. Classifying multi-stage APTs was made stable and robust by majority voting across the trees.

3.4.3. Support Vector Machine

The SVM algorithm relies on maximizing the margin between classes in the attribute space [36], giving it a strong ability to distinguish between complex patterns, especially in high dimensional spaces. This algorithm is widely used in IDS due to its flexibility in handling linear and non- linear separation and its ability to achieve stable performance when high quality labeled data is available [20]. SVM was used as a classical margin-based classifier to determine its usefulness in multi-stage APT detection. The model was trained using a normalized numerical network features and applied a stratified 70/15/15 split to ensure balanced representation across stages [37].

To obtain the best results, it was best to use a kernel-based transformation to capture the nonlinear aspects of the feature space. SVM was, however, noted to have poorer performance in contrast to ensemble-based methods. This is due to the high dimensions and non-linear interactions present in the network traffic of IIoT that are best handled by tree based boosting. The optimization of SVM also became very computationally expensive for the large datasets, making SVM unable to scale the datasets easily.

These findings are consistent with the previous literature surrounding Intrusion IDS where the use of boosting ensemble methods almost always outperforms the margin-based classifiers in complex multi-class intrusion detection scenarios.

3.4.4. Hybrid Decision Tree

To further improve multi-stage APT detection [38], a Hybrid Decision Tree framework merges rule-based hierarchical structuring with data-based tree learning. The framework consists of two layers, with the first layer performing coarse-level classification to determine if observed traffic is normal or an attack [39], while the

second layer is a more refined level classification of the type of attack. This type of hierarchical decomposition helps to further class overlap and the complexity of multi-class discrimination [40].

The decision tree element was developed with a number of encoded network attributes and a decomposition of the data set while using stratification for the data. A combination of both a cap on the depth of the tree and a set number of data points was used to limit overfitting [41]. The hybrid method also organized the classification into multiple layers of sequential decisions to enhance interpretability and minimize misclassification of APT stages that are closely related.

3.5. Select the Best ML Model

For intrusion detection, fore machine learning algorithms were used: XGBoost, Random Forest RF, SVM and Hybrid DT. These algorithms were chosen because they are known to be effective with high-dimensional data and imbalanced classification problems. Each model was trained and validated with a stratified 70/15/15 split (details in Section 3.3) on each of the three preprocessed datasets. Each model's hyperparameters were tuned via grid search on the validation dataset to improve performance.

Across all datasets and classification problems (binary, multi-class, and stage-wise), XGBoost was the best performer compared to RF, SVM and Hybrid DT. Table 2 shows a comparison of in terms of Accuracy, Precision, Recall, F1-Score, and ROC-AUC. In stage-wise APT classification of the CICAPT-IIoT dataset, XGBoost was the best performer, and demonstrated the ability to identify complex non-linear interactions among tier-2 features. XGBoost was the best performer in all metrics because of the built-in class imbalance regulation, class weighting, and regularization.

XGBoost is able to outperform competitors thanks to its unique gradient boosting framework which corrects error of precursor trees in each boosting round. Additionally, the framework can natively deal with missing values and sparse datasets. Furthermore, the selected features using SHAP (described in Section 3.7) enhanced the model's interpretability and generalization. Thus, XGBoost was chosen as the main model for the forthcoming analysis and real-world application in the Chrome Extension.

3.6. Performance Evaluation

The proposed models were evaluated using standard metrics derived from the confusion matrix consisting of four elements: True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN) [42]-[44]. The metrics calculated include:

1. Accuracy is an overall measure of the model and is calculated as the number of correct predictions divided by the number of total predictions.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{1}$$

2. Precision (or Positive Predictive Value) is the measure of the fraction of all positive predictions that are correct.

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

3. Recall (or Sensitivity or True Positive Rate) is the measure of the fraction of all positives that are correctly predicted.

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

4. The F1-Score is the weighted average of Precision and Recall and is especially useful for assessing datasets that are not balanced.

$$F1 - scor = 2 * \frac{Precision * Recall}{Prrecision + Recall} \tag{4}$$

5. False Positive Rate (FPR) measures the percentage of negative instances that are misclassified as positive:

$$FPR = \frac{FP}{FP + TN} \tag{5}$$

6. ROC-AUC (Receiver Operating Characteristic - Area Under the Curve) summarizes the ability of the model's ability to tell the difference among various classification thresholds. It is a measure of the area

under the curve of the True Positive Rate (TPR) and the False Positive Rate (FPR) and may be understood as the likelihood of a positive randomly than a negative randomly selected instance:

$$ROC - AUC = \int_0^1 TPR(FPR)d(FPR) \tag{6}$$

3.7. Features Selection Using SHAP

To start, the XGBoost model was trained on all features available (70 features for the CICAPT-IIoT dataset and 46 features for CICIoT2023). While the model performed detection excellently, we wanted to focus on explainability and cut down the computing costs. To do this we utilized SHAP (SHapley Additive exPlanations) for feature selection.Here's what we did: After training the XGBoost model previously described, SHAP features were computed using TreeExplainer for all features. Features were then ranked according to the mean absolute SHAP value, which represent the mean of the feature's contribution to the model's prediction.

In the context of this study, the authors kept the remaining 20 features, which represented around 95% of the remaining importance (based on the elbow point of the importance curve). We then retrained the XGBoost model on these 20 features.

The retrained model, even with its fewer features, improved explainability and training time while standing up to the full-featured model across all evaluation criteria (Accuracy, Precision, Recall, F1, and ROC-AUC). That most notable result means the 20 features selected are capturing the most critical network flow attributes for both APT and non-APT attacks. Both the SHAP summary and the bar plots in Figure 2, provide evidence. Ultimately, this selection process means that the final model that will be deployed with the Chrome extension will be efficient and explainable, by only using the top features.

Only a portion of the features shows significant influence on the model's predictions [45][46]. The SHAP values in the document corroborate the model's interpretability, as the values are consistently in the same range and push predictions to either the normal or malicious classes [47].

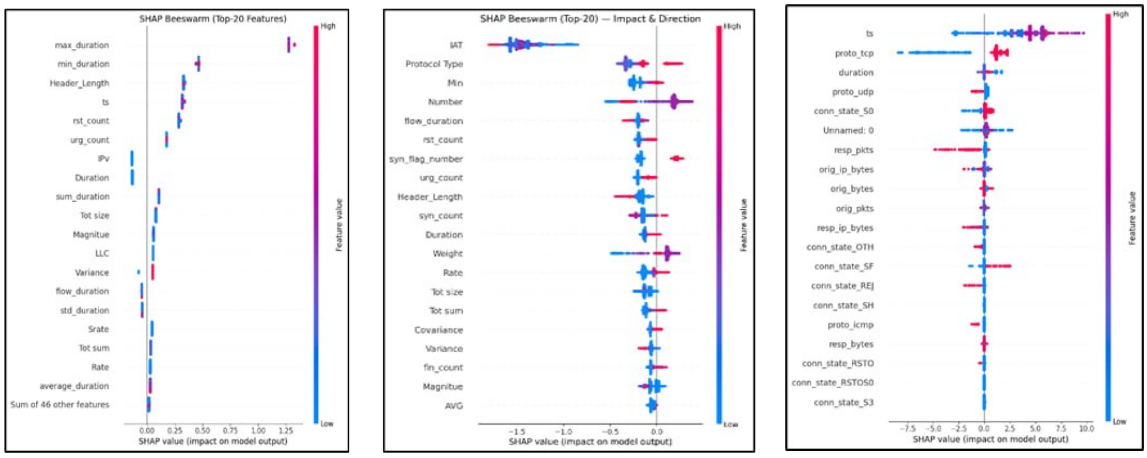


Figure 2. The features affecting model predictions are detailed for all datasets

3.8. Model Explanation and SHAP Validation

Applying SHAP again after finalizing the XGBoost model using the 20 features outlined in Section (3.7), was done to provide some level of reasoning and justification to the model's reasoning in each of the classification tasks (binary (benign vs attack), multi-class (what kind of attack), and in step (phases of APT) classifications. With TreeExplainer, for each prediction during the testing phase, SHAP values are the decomposition of the output into an additive feature attribution, which makes visible the contribution of each feature to the final prediction. As an example, SHAP is used to clarify why a certain flow is considered malicious in a binary classification, provide the classification of a DDoS attack and distinguish it from reconnaissance in a multi-class classification, and to clarify the features associated with certain APT phases such as Recon/InitialAccess and Exfiltration/Impact. The fact that SHAP is able to justify some of these classifications after the data has already been used in the model helps to establish trust with analysts by providing justification and reasoning for each classification, since the model is not designed with transparency in mind.

In the global SHAP importance analysis (Table 3, Section 4.2), the features are ranked based on their mean absolute SHAP values. It can be seen ts (timestamp), Header_Length, and urg_count carry the most weight across all predictions. The local explanations shed more light on particular samples (e.g., correctly classified attack samples vs. misclassified attack samples) and show the values of the features that influence the model most. A case in point, the correctly classified attack sample demonstrated the most influence from ts, urg_count, and Header_Length, and the misclassified sample showed the most influence from the flow_active_time and flow_duration features. Such case studies show that the model is not a black box and its decisions match known patterns of IoT attacks.

Model SHAP interaction values enhance SHAP by yielding pairwise feature dependencies (e.g., flow_duration and Header_Length), which adds a different type of interpretability. Having SHAP to interpret features and model explains why the proposed system has a multi-faceted approach to explainability, which is needed for real-world applications like the Chrome extension. The visual results from the SHAP summary, force, and interaction plots in (Section 4.2) parallel the model's reliability and support the conclusions claimed,As show in the following Figure 3.

Top local features driving this prediction:

	feature	feature_value	shap_value
0	ts	1.701604e+09	1.819248
1	Header_Length	6.600000e+02	0.527332
2	flow_active_time	4.590759e+01	-0.338431
3	flow_duration	4.590759e+01	-0.260087
4	AVG	7.028571e+01	0.086071
5	urg_count	0.000000e+00	-0.059819
6	Srate	2.178289e-01	0.012763
7	ack_flag_number	1.000000e+00	0.001979
8	max_duration	1.701604e+09	0.000000
9	sum_duration	1.701604e+10	0.000000

Figure 3. Top features prediction

4. EXPERIMENTS AND RESULTS

4.1. Performance Evaluation of Full Features

An evaluation was undertaken using the datasets' available features initially. The performance of four ML models—XGBoost, Random Forest (RF), Hybrid Decision Tree, and Support Vector Machine (SVM)—is given in Table \ref{tab:full_features}. XGBoost recorded the highest Precision and F1-score, alluding to its best performance through the models provided for the multi-class intrusion detection problem. The performances of Random Forest and Hybrid Decision Tree were closely ranked, and due to the high-dimensional feature space, SVM recorded the lowest performance metrics because of its high sensitivity. Therefore, XGBoost will be used as the best-performing baseline model and will be used for further optimization and reason, as in Table 2.

4.2. Model Interpretation Using SHAP

All datasets showed the XGBoost model to be the best performing and therefore the most appropriate model to be the most appropriate model to be the primary classifier. In regard to SHAP-based feature selection (top 20 features), performance was consistent or improved: results were consistent in CICIoT2023; performance was improved in CICAPT-IIoT; and results were again consistent in IoT-23. This verifies that the features selected are robust and sufficient, as in the following Table 3.

Table 3 shows XGBoost performance pre and post the top-20 features selection through SHAP. From the results, it is observed that the accuracy is maintained (or in some instances, it is improved) when the feature space is reduced, substantiating the effectiveness of SHAP when it comes to dimensionality reduction.

Table 2. Results for all features in Dataset Collection

Dataset	Model	Accuracy	F1-scor	Precision	Recall	ROC-AUC
CICIoT2023	**XGBoost**	**0.8534**	**0.9925**	**0.9933**	**0.9920**	**0.9999**
	Random Forest	0.8174	0.9898	0.9910	0.9891	0.9999
	Hybrid-Decision Tree	0.8031	0.9784	0.9867	0.9729	0.9994
	SVM	0.0754	0.6538	0.0317	0.0314	0.5988
CICAPT-IIoT	**XGBoost**	**0.9916**	**0.9810**	**0.9955**	**0.9920**	**1.0000**
	Random Forest	0.9286	0.9453	0.8317	0.9891	0.9981
	Hybrid-Decision Tree	0.9286	0.8900	0.9614	0.9729	0.9877
	SVM	0.7059	0.4767	0.3673	0.6380	0.8596
ON IoT-23	**XGBoost**	**0.9969**	**0.9969**	**0.9969**	**0.9969**	**0.9985**
	Random Forest	0.9970	0.9781	0.9970	0.9970	0.9985
	Hybrid-Decision Tree	0.9968	0.9967	0.9968	0.9968	0.9964
	SVM	0.9630	0.9449	0.9274	0.9630	0.0633

Table 3. Results of the top 20 features in dataset collection

Dataset	Model	Accuracy	F1-scor	Precision	Recall	ROC-AUC
CICIoT2023	**XGBoost Before SHAP (All Features)**	0.8534	0.9925	0.9933	0.9920	0.9999
	XGBoost After SHAP (Top-20)	**0.8564**	**0.9923**	**0.9932**	**0.9917**	**0.9999**
CICAPT-IIoT	**XGBoost Before SHAP (All Features)**	0.9832	0.9829	0.9848	0.9832	0.9968
	XGBoost After SHAP (Top-20)	**0.9882**	**0.9882**	**0.9885**	**0.9882**	**0.9951**
ON IoT-23	**XGBoost Before SHAP (All Features)**	0.9969	0.9969	0.9969	0.9969	0.9985
	XGBoost After SHAP (Top-20)	**0.9970**	**0.9969**	**0.9969**	**0.9969**	**0.9985**

4.3. Comparison with State-of-the-Art Studies

Our framework can be contextualized with recent studies on IoT intrusion detection as shown in Table 4. Although other studies successfully demonstrate a strong performance on binary classifications, they do not do as strong of an evaluation on other classifications. Our XGBoost + SHAP model, on the other hand, is able to perform on par or better than other models in evaluations regardless of the task type, thereby showing its the ability to be generalizable and the ability to be robust on a number of different datasets. Table 4 illustrates the results.

The details in Table 4 compares the latest studies in IoT intrusion detection with the framework being proposed. Previous works made huge successes with binary detection, but their performance evaluations are limited and in comparison, to classical classification evaluations. The framework being proposed here does better in general and offers more competitive results with more sophistication in classification frameworks—like step-wise and hierarchical classification. The consistently good performance across varying datasets confirms the proposed XGBoost-methods generality and lucidity.

Table 4. Visualize the compression of the proposed model with the most-relevant current studies.

Ref.	Model / Approach	Year	Dataset	Classification Type	Accuracy	F1-Score	Notes
[48]	Curriculum Learning + XAI	2025	CICAPT-IIoT	Binary	98%	96%	APT-focused dataset
[49]	Curriculum Learning + XAI	2025	CICIoT2023	Binary	98%	97%	IoV attacks
[50]	Hybrid Voting (RF + XGB + AdaBoost)	2025	IoT-23	Binary	99.99%	99.99%	Hybrid ensemble
[50]	Hybrid Voting (RF + XGB + AdaBoost)	2025	IoT-23	Multi-class	99%	99%	Multi-class evaluation
	Proposed XGBoost (This Study)		**CICAPT-IIoT**	**Stage-wise APT**	**98.82%**	**98.82%**	**Stage-level classification**
	Proposed XGBoost (This Study)		**IoT-23**	**Hierarchical (Binary → APT → Stage)**	**99.69%**	**99.69%**	**Hierarchical detection**
	Proposed XGBoost (This Study)		**CICIoT2023**	**Multi-class**	**99.25%+**	**99.25%+**	**Large-scale IoT dataset**

4.4. Real-World Deployment Using IoT-23 Model

To test how the framework works in reality, we implemented an XGBoost model using the IoT-23 dataset by making a Chrome extension. The extension serves as a tool for monitoring visits to pages by the user, including the capturing of URLs to pages. For each URL, a number of features are captured and engineered (e.g. length of the URL, number of subdomains, special characters, and url entropy). The features are then sent to a cloud server where a backend REST API is configured, and the model is trained to make inferences in real time. The model is ready to make predictions and classify a URL as either benign or malicious and with a score of how confident the model is in its prediction, which is then shown to the user through an informative pop up. This way, users are able to make decisions about which sites to visit. The extension also anonymizes predictions

to show further logged information, exemplifying the seamless integration of the proposed IDS with the users browsing routine, in contrast to the proposed theoretical and academic cybersecurity solutions. The next Figure 4 shows the extension and the application.

The browser extension presents a streamlined interface for URL-based and flow-based analyses, as illustrated in Figure Y. The captured input is sent to a backend API where a model powered by ML Technologies performs real-time inference. The predictive outputs contain details on whether an attack was detected, the attack's severity, and degree of confidence, as well as the attack's category (e.g., MITRE classification). This architecture provides the optimal balance between user interface and intrusion detection engine interaction, while conserving computational resources.

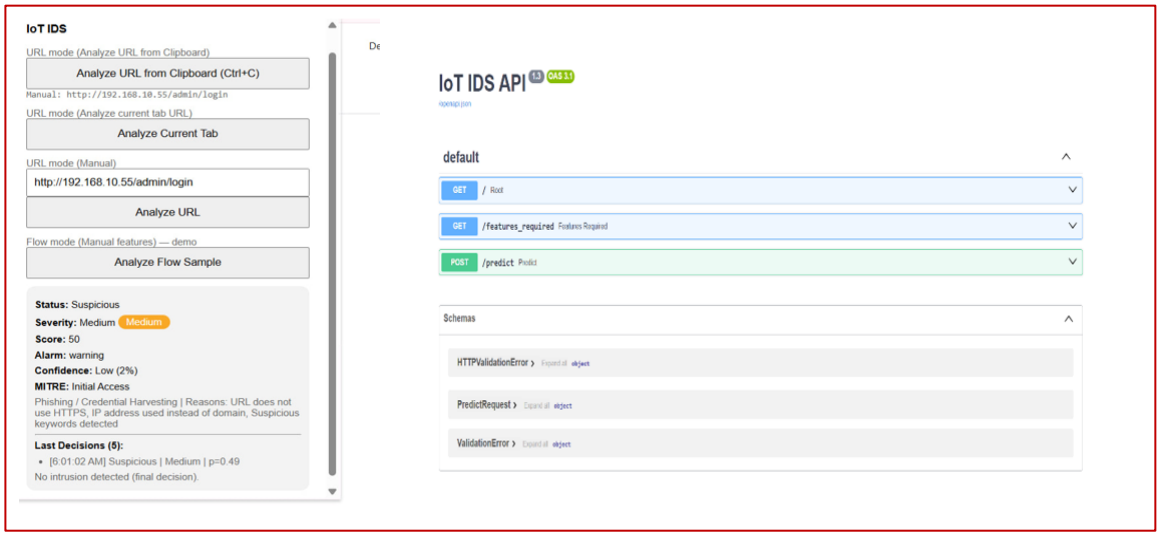


Figure 4. Illustrates the user interface of the developed Chrome extension and how it interacts with the back-end API

5. CONCLUSION

The contribution of this paper is an explainable IDS designed specifically for heterogeneous IoT ecosystems. The framework has been meticulously benchmarked with the datasets CICIoT2023, CICAPT-IIoT, and IoT-23, which cover multi-class, step, and level detection. Among the baseline models, XGBoost was the best performer with an accuracy of 99.69% on IoT-23 and 98.82% on CICAPT-IIoT, demonstrating solid generalization across different patterns of attacks. The combination with SHAP adds transparency to the models concerning the selection and justification of features, allowing for a reduction of the set of features by 65% and maintaining the detection accuracy. Practicability has been shown with a lightweight chrome extension that performs real-time detection of URLs through an inference engine on the back end. The proposed framework captures a good balance among accuracy, interpretability, and efficiency. This is especially needed for the analysis of real-world IoT security implementations. The extension of this framework for adversarial attacks through a resilient and continuously adaptive approach to learning will be developed along with the framework's potential adaptability to edge devices with limited resources.

REFERENCES

[1] K. V. V. N. L. Sai Kiran, R. N. K. Devisetty, N. P. Kalyan, K. Mukundini, and R. Karthi, "Building a Intrusion Detection System for IoT Environment using Machine Learning Techniques," *Procedia Comput. Sci.*, vol. 171, no. 2019, pp. 2372–2379, 2020, https://doi.org/10.1016/j.procs.2020.04.257.

[2] A. Verma and V. Ranga, "Machine Learning Based Intrusion Detection Systems for IoT Applications," *Wirel. Pers. Commun.*, vol. 111, no. 4, pp. 2287–2310, 2020, https://doi.org/10.1007/s11277-019-06986-8.

[3] M. Stoyanova, Y. Nikoloudakis, S. Panagiotakis, E. Pallis, and E. K. Markakis, "A Survey on the Internet of Things (IoT) Forensics: Challenges, Approaches, and Open Issues," *IEEE Commun. Surv. Tutorials*, vol. 22, no. 2, pp. 1191–1221, 2020, https://doi.org/10.1109/COMST.2019.2962586.

[4] M. ElKashlan, M. S. Elsayed, A. D. Jurcut, and M. Azer, "A Machine Learning-Based Intrusion Detection System for IoT Electric Vehicle Charging Stations (EVCSs)," *Electron.*, vol. 12, no. 4, pp. 1–17, 2023, https://doi.org/10.3390/electronics12041044.

[5] H. Jiang, Z. He, G. Ye, and H. Zhang, "Network Intrusion Detection Based on PSO-Xgboost Model," *IEEE Access*, vol. 8, pp. 58392–58401, 2020, https://doi.org/10.1109/ACCESS.2020.2982418.

[6] M. Mohammadi *et al.*, "A comprehensive survey and taxonomy of the SVM-based intrusion detection systems," *J. Netw. Comput. Appl.*, vol. 178, no. January, p. 102983, 2021, https://doi.org/10.1016/j.jnca.2021.102983.
[7] N. Elsayed, Z. S. Zaghloul, S. W. Azumah, and C. Li, "Intrusion Detection System in Smart Home Network Using Bidirectional LSTM and Convolutional Neural Networks Hybrid Model," *Midwest Symp. Circuits Syst.*, vol. 2021-Augus, pp. 55–58, 2021, https://doi.org/10.1109/MWSCAS47672.2021.9531683.
[8] G. Dorado, S. Gálvez, and M. del P. Dorado, "Computer firewalls: security and privacy protection for Mac—review," *Big Data Inf. Anal.*, vol. 6, no. 0, pp. 1–11, 2021, https://doi.org/10.3934/bdia.2021001.
[9] M. A. Umar, Z. Chen, and Y. Liu, "A Hybrid Intrusion Detection with Decision Tree for Feature Selection," *Inf. Secur. An Int. J.*, pp. 1–16, 2021, https://doi.org/10.11610/isij.4901.
[10] G. Kocher and G. Kumar, "Machine learning and deep learning methods for intrusion detection systems: recent developments and challenges," *Soft Comput.*, vol. 25, no. 15, pp. 9731–9763, 2021, https://doi.org/10.1007/s00500-021-05893-0.
[11] S. M. Kasongo and Y. Sun, "Performance Analysis of Intrusion Detection Systems Using a Feature Selection Method on the UNSW-NB15 Dataset," *J. Big Data*, vol. 7, no. 1, 2020, https://doi.org/10.1186/s40537-020-00379-6.
[12] M. Asif, S. Abbas, M. A. Khan, A. Fatima, M. A. Khan, and S. W. Lee, "MapReduce based intelligent model for intrusion detection using machine learning technique," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 34, no. 10, pp. 9723–9731, 2022, https://doi.org/10.1016/j.jksuci.2021.12.008.
[13] A. Alrefaei and M. Ilyas, "Using Machine Learning Multiclass Classification Technique to Detect IoT Attacks in Real Time," *Sensors*, vol. 24, no. 14, 2024, https://doi.org/10.3390/s24144516.
[14] A. S. AL-Aamri, R. Abdulghafor, S. Turaev, I. Al-Shaikhli, A. Zeki, and S. Talib, "Machine Learning for APT Detection," *Sustain.*, vol. 15, no. 18, 2023, https://doi.org/10.3390/su151813820.
[15] A. R. Gad, A. A. Nashat, and T. M. Barkat, "Intrusion Detection System Using Machine Learning for Vehicular Ad Hoc Networks Based on ToN-IoT Dataset," *IEEE Access*, vol. 9, pp. 142206–142217, 2021, https://doi.org/10.1109/ACCESS.2021.3120626.
[16] E. E. Abdallah, W. Eleisah, A. F. Otoom, and I. I. Ahmed, "Intrusion Detection Systems using Supervised Machine Learning Techniques : survey," *Procedia Comput. Sci.*, vol. 201, pp. 205–212, 2022, https://doi.org/10.1016/j.procs.2022.03.029.
[17] J. Ables *et al.*, "Eclectic Rule Extraction for Explainability of Deep Neural Network based Intrusion Detection Systems," *arXiv preprint arXiv:2401.10207*, 2024, https://doi.org/10.1016/j.procs.2022.03.029.
[18] P. Sharon Femi, K. Ashwini, A. Kala, and V. Rajalakshmi, "Explainable Artificial Intelligence for Cybersecurity," *Wirel. Commun. Cybersecurity*, pp. 149–174, 2023, https://doi.org/10.1002/9781119910619.ch7.
[19] S. A. Memon, U. K. Wiil, and M. Shaikh, "Explainable Intrusion Detection for Internet of Medical Things," *Int. Jt. Conf. Knowl. Discov. Knowl. Eng. Knowl. Manag. IC3K - Proc.*, vol. 3, no. Ic3k, pp. 40–51, 2023, https://doi.org/10.5220/0012210300003598.
[20] I. Malashin, V. Tynchenko, A. Gantimurov, and V. Nelyub, "Support Vector Machines in Polymer Science: A Review," *Polymers*, vol. 17, no. 4, p. 491, 2025, https://doi.org/10.3390/polym1740491.
[21] M. Wang, K. Zheng, Y. Yang, and X. Wang, "An Explainable Machine Learning Framework for Intrusion Detection Systems," *IEEE Access*, vol. 8, pp. 73127–73141, 2020, https://doi.org/10.1109/ACCESS.2020.2988359.
[22] S. S. Dhaliwal, A. Al Nahid, and R. Abbas, "Effective intrusion detection system using XGBoost," *Inf.*, vol. 9, no. 7, 2018, https://doi.org/10.3390/info9070149.
[23] Y. Hosain and M. Çakmak, "XAI-XGBoost: an innovative explainable intrusion detection approach for securing internet of medical things systems," *Sci. Rep.*, vol. 15, no. 1, pp. 0–17, 2025, https://doi.org/10.1038/s41598-025-07790-0.
[24] S. Hizal, U. Cavusoglu, and D. Akgun, "A novel deep learning-based intrusion detection system for IoT DDoS security," *Internet of Things (Netherlands)*, vol. 28, 2024, https://doi.org/10.1016/j.iot.2024.101336.
[25] B. Xu, L. Sun, X. Mao, R. Ding, and C. Liu, "IoT Intrusion Detection System Based on Machine Learning," *Electron.*, vol. 12, no. 20, 2023, https://doi.org/10.3390/electronics12204289.
[26] V. Malele and T. E. Mathonsi, "Testing the performance of Multi-class IDS public dataset using Supervised Machine Learning Algorithms," *arXiv preprint arXiv:2302.14374*, 2023, https://doi.org/10.48550/arXiv.2302.14374.
[27] N. Y. Jien, M. Tahir, M. Dabbagh, Y. K. Meng, and A. Farooq, "Performance Evaluation of Machine Learning Algorithms for Intrusion Detection in IoT Applications," *4th IEEE Int. Conf. Artif. Intell. Eng. Technol. IICAIET 2022*, pp. 1–6, 2022, https://doi.org/10.1109/IICAIET55139.2022.9936863.
[28] Y. M. Banadaki, "Evaluating the performance of machine learning algorithms for network intrusion detection systems in the internet of things infrastructure," *J. Adv. Comput. Sci. Technol.*, vol. 9, no. 1, pp. 14–20, 2020, https://doi.org/10.14419/jacst.v9i1.30992.
[29] S. Akram, M. A. Iqbal, M. Rashid, M. S. Bhatti, and B. Fida, "A Hybrid Deep Learning Framework for Early-Stage Alzheimer's Disease Classification From Neuro-Imaging Biomarkers," *IEEE Access*, vol. 13, pp. 101662–101680, 2025, https://doi.org/10.1109/ACCESS.2025.3574039.
[30] A. M. Banaamah and I. Ahmad, "Intrusion Detection in IoT Using Deep Learning," *Sensors*, vol. 22, no. 21, 2022, https://doi.org/10.3390/s22218417.
[31] M. Bacevicius, A. Paulauskaite-Taraseviciene, G. Zokaityte, L. Kersys, and A. Moleikaityte, "Comparative Analysis of Perturbation Techniques in LIME for Intrusion Detection Enhancement," *Mach. Learn. Knowl. Extr.*, vol. 7, no. 1, pp. 1–18, 2025, https://doi.org/10.3390/make7010021.

[32] H. Ghani, S. Salekzamankhani, and B. Virdee, "Statistical and Multivariate Analysis of the IoT-23 Dataset : A Comprehensive Approach to Network Traffic Pattern Discovery," vol. 2028, pp. 1–22, 2025, https://doi.org/10.3390/jcp5040112.
[33] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785-794, 2016, https://doi.org/10.1145/2939672.2939785.
[34] H. A. Salman, A. Kalakech, and A. Steiti, "Random Forest Algorithm Overview," *Babylonian Journal of Machine Learning*, vol. 2024, pp. 69–79, 2024, https://doi.org/10.58496/BJML/2024/007.
[35] A. Khanan, Y. Abdelgadir Mohamed, A. H. H. M. Mohamed, and M. Bashir, "From Bytes to Insights: A Systematic Literature Review on Unraveling IDS Datasets for Enhanced Cybersecurity Understanding," *IEEE Access*, vol. 12, pp. 59289–59317, 2024, https://doi.org/10.1109/ACCESS.2024.3392338.
[36] A. Halbouni, T. S. Gunawan, M. H. Habaebi, M. Halbouni, M. Kartiwi, and R. Ahmad, "Machine Learning and Deep Learning Approaches for CyberSecurity: A Review," *IEEE Access*, vol. 10, pp. 19572–19585, 2022, https://doi.org/10.1109/ACCESS.2022.3151248.
[37] E. Altulaihan, M. A. Almaiah, and A. Aljughaiman, "Anomaly Detection IDS for Detecting DoS Attacks in IoT Networks Based on Machine Learning Algorithms," *Sensors*, vol. 24, no. 2, 2024, https://doi.org/10.3390/s24020713.
[38] C. Do Xuan, "Detecting APT attacks based on network traffic using machine learning," *J. Web Eng.*, vol. 20, no. 1, pp. 171–190, 2021, https://doi.org/10.13052/jwe1540-9589.2019.
[39] D. Upadhyay, J. Manero, M. Zaman, and S. Sampalli, "Intrusion Detection in SCADA Based Power Grids: Recursive Feature Elimination Model with Majority Vote Ensemble Algorithm," *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 3, pp. 2559–2574, 2021, https://doi.org/10.1109/TNSE.2021.3099371.
[40] B. R. Kikissagbe and M. Adda, "Machine Learning-Based Intrusion Detection Methods in IoT Systems: A Comprehensive Review," *Electron.*, vol. 13, no. 18, 2024, https://doi.org/10.3390/electronics13183601.
[41] E. Alhajjar, P. Maxwell, and N. Bastian, "Adversarial machine learning in Network Intrusion Detection Systems," *Expert Syst. Appl.*, vol. 186, pp. 1–25, 2021, https://doi.org/10.1016/j.eswa.2021.115782.
[42] A. Heidari and M. A. Jabraeil Jamali, "Internet of Things intrusion detection systems: a comprehensive review and future directions," *Cluster Comput.*, vol. 26, no. 6, pp. 3753–3780, 2023, https://doi.org/10.1007/s10586-022-03776-z.
[43] T. R. Mahesh, V. Vivek, and V. Kumar, "Implementation of Machine Learning-Based Data Mining Techniques for IDS," *Int. J. Inf. Technol. Res. Appl.*, vol. 2, no. 1, pp. 7–13, 2023, https://doi.org/10.59461/ijitra.v2i1.23.
[44] Y. K. Saheed and M. O. Arowolo, "Efficient Cyber Attack Detection on the Internet of Medical Things-Smart Environment Based on Deep Recurrent Neural Network and Machine Learning Algorithms," *IEEE Access*, vol. 9, pp. 161546–161554, 2021, https://doi.org/10.1109/ACCESS.2021.3128837.
[45] R. Younisse, A. Ahmad, and Q. Abu Al-Haija, "Explaining Intrusion Detection-Based Convolutional Neural Networks Using Shapley Additive Explanations (SHAP)," *Big Data Cogn. Comput.*, vol. 6, no. 4, 2022, https://doi.org/10.3390/bdcc6040126.
[46] V. Z. Mohale and I. C. Obagbuwa, "Evaluating machine learning-based intrusion detection systems with explainable AI: enhancing transparency and interpretability," *Front. Comput. Sci.*, vol. 7, pp. 1–23, 2025, https://doi.org/10.3389/fcomp.2025.1520741.
[47] S. Neupane *et al.*, "Explainable Intrusion Detection Systems (X-IDS): A Survey of Current Methods, Challenges, and Opportunities," *IEEE Access*, vol. 10, pp. 112392–112415, 2022, https://doi.org/10.1109/ACCESS.2022.3216617.
[48] S. Narkedimilli, A. Sai, and S. D., "Enhancing IoT Network Security through Adaptive Curriculum Learning and XAI," in *Proceedings of the 17th ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec '24)*, pp. 1–2, 2024, https://doi.org/10.1109/FNWF66845.2025.11317635.
[49] K. S. Adewole, A. Jacobsson, and P. Davidsson, "Intrusion Detection Framework for Internet of Things with Rule Induction for Model Explanation," *Sensors*, vol. 25, no. 6, pp. 1–27, 2025, https://doi.org/10.3390/s25061845.
[50] M. A. Akif, I. Butun, A. Williams, and I. Mahgoub, "Hybrid Machine Learning Models for Intrusion Detection in IoT: Leveraging a Real-World IoT Dataset," *arXiv preprint arXiv:2502.12382*, 2025, https://doi.org/10.1109/ISNCC62547.2024.10759058.