International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(16s) 2026 pp. 902-825
https://doi.org/10.70917/ijcisim-2026-4621
Article

Explainable Hybrid Framework for Zero Day Web Attack Detection Using Anomaly Detection and Classification Models

Shreya A Patil[1], Anita Harsoor[2]

[1]Department of Computer Science and Engineering, PDA College of Engineering, Kalaburagi
[2]Department of Computer Science and Engineering, PDA College of Engineering, Kalaburagi
Email: shreya.apatil22@gmail.com, anitaharsoor@pdaengg.com

Abstract: The rise in the usage of web applications has elevated the risk of cyber attacks, specifically zero day web attacks. Zero day web attacks are the unseen and unusual attacks which traditional methods fail to detect. This essay's primary goal is to build a system that detects zero day web attacks efficiently. The proposed system utilises CICIDS2017 dataset and applies SMOTE method for creating artificial samples to balance the class. For anomaly detection, the system utilizes unsupervised techniques like autoencoder and isolation forest. The study aims to implement supervised algorithms as well, such as Logistic regression, linear support vector machines, and random forests and XGBoost respectively. Among all these models, XGBoost outperforms with 0.99 accuracy, 0.98 precision, 1.00 recall and 0.99 F1-score. The system classifies the traffic as normal or attack enabling efficient detection of cyber attack. Explainable AI (XAI) techniques like SHaply Addictive exPlanations (SHAP) and Local Interpretable Model-agnostic Explanations (LIME) are applied To enhance the interpretability. It is observed that SHAP is the higher suitable technique for explaining the identification of attacks. The paper mainly focuses on improving the detection capability of the setup and maintaining transparency in decision making. This study marks its contribution in providing important perspectives in the area of cybersecurity for efficient detection of zero day web attacks.
Keywords: Zero day web attack, Anomaly Detection, Classification Models, Explainable AI (XAI)

1. Introduction

The exponential growth in the usage of internet and web-applications has resulted in the huge exchange of data through networks. Improvement in the means of communication and its accessibility is appreciable But this has also brought up the chances of cyber threats and attacks. An increase in the quantity of these attacks like malware, distributed denial of service (DDoS), brute force, cross-site scripting (XSS) and SQL

injection creates a demand for an efficient security system that has the ability to preventing such cyber attacks. In modern cybersecurity, IDS, or Systems for detecting intrusions are crucial for monitoring the network traffic for identifying any kind of suspicious behaviour or any activity trying to gain unauthorised access to the system. These IDS are known for generating alert messages or a report on detecting the malicious activity, allowing the security

analyst to investigate and take necessary precautions as shown in the figure 1.1. Based upon the deployment method, IDS is classified as a Both host-based and intrusion detection systems that are based on networks (NIDS and HIDS, respectively). Both HIDS and NIDS scan network traffic. scans over the hosts and devices to capture the malicious activity. IDS is categorized as a signature-based intrusion detection system depending on the detection technique. and Anomaly based Intrusion Detection System. Signature based attacks are classified by comparing the network with the known attacks, while anomaly based attacks are identified by their deviation from the normal behaviour. This makes anomaly based IDS a suitable system for detecting unknown or unseen attacks.

In spite of so much technical advancement, many IDS still focus on detecting the known attacks and fail to identify the unknown attacks. These unknown attacks are the zero day attacks. They are the attacks where the vulnerabilities present within the software or hardware systems are exploited unknowingly, without being aware of the developer or vendor. Since the flaw is unnoticed by anyone, the system can be exploited for days, months or sometimes for years, unless it is reported to the developer. This type of attack is known as Zero Day because the developer has got zero days to patch the vulnerability. Hence the term "Zero Day" Attack. Because of the unknown patterns of zero day attacks, developing A method for identifying them is a major challenge. The paper majorly focuses on detecting zero day web attacks.

Due to its ability to handle huge datasets and to identify anomalous behaviour, Techniques for machine learning have grown more a lot of importance. For capturing these zero day attacks, it was considered anomaly detection, an unsupervised technique that is more effective in identifying the zero day web attacks. Therefore a hybrid approach is proposed for detecting the zero day web attacks using an unsupervised and supervised model.

The proposed system incorporates CICIDS2017 dataset which includes the network traffic. Over this dataset, preprocessing techniques are implemented for efficient understanding and handling of the data. For developing the zero day detection system, data is prepared by applying various techniques like splitting of dataset into train and test, balancing dataset by SMOTE, feature importance, feature selection and scaling of features. SMOTE technique enables generating synthetic samples for the minority class. Further, techniques for identifying irregularities, like isolation forest and autoencoder models were developed. Looking over the results of anomaly detection, supervised classification models like Logistic Regression, Linear Support Vector Machine, Random Forest and XGBoost were implemented. XAI techniques are applied over the best performing models like LIME and SHAP, which explains The forecast for the output. XAI helps to enhance the transparency and the comprehensibility of the system making the proposed IDS reliable.

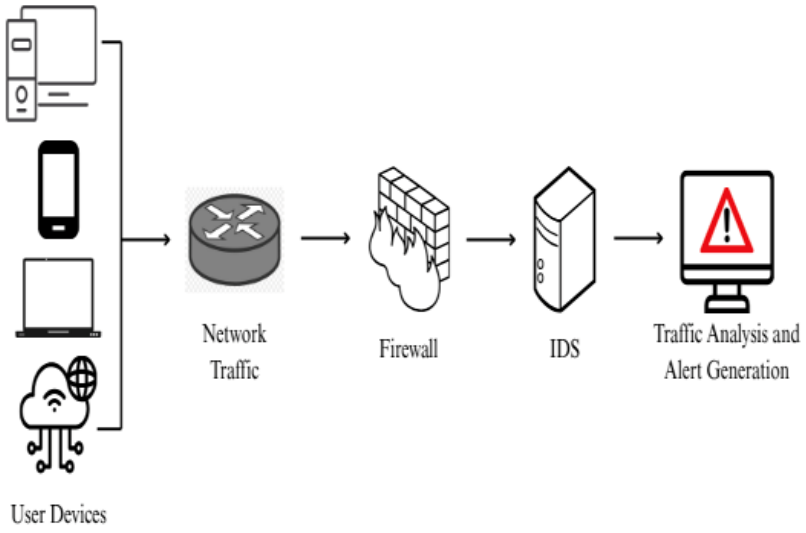

Figure 1.1 Role of IDS

2. Related Work

In this section, we mainly review the work related to the detection of zero day attacks, where the network traffic will be classified as normal or attack.

The frequency of cyberattacks is increasing, and more complicated. The old ways of detecting these attacks are not working well. Thus, scientists are investigating usage of machine learning. as well as deep learning to find zero-day attacks. Recent studies are focusing on finding attack patterns by looking at behavior and learning features.

According to several researchers, using learning models that do not need any supervision to detect zero-day web attacks. For example ZeroWall used a kind of neural network to learn what normal HTTP requests look like and find anomalies[1]. It used a score-based system to find requests that were not typical ones. Another approach used a combination of techniques like LSTM and GRU to detect web attacks like SQL injection and Cross-Site Scripting. These methods were very good at finding anomalies. Did not give many false positives[6].Machine learning is Additionally, becoming accustomed to detect web attacks. A framework that used Random Forest-style supervised learning algorithms was very good at detecting attacks like SQL injection and Cross-Site Scripting. It used benchmark datasets like CSIC2010 and ECML/PKDD 2007[2]. Other techniques like One-Class SVM and Autoencoders are being used to find unseen attack behaviors[3].Recently Deep learning has been used by researchers. and graph-based techniques to improve intrusion detection. For example WebWall used a graph network to detect zero-day attacks in encrypted HTTPS traffic[4]. It looked at packet flow dependencies and

anomaly scores. Another approach used attention-based hybrid models to detect assaults on the Internet of Things settings. Additionally, it used Explainable Artificial Intelligence techniques like SHAP and LIME to make the model more interpretable[7].Being able to describe how the model works is becoming very important. Many studies are Explainable Artificial Intelligence approaches to make the model more transparent and trustworthy. For example frameworks that use SHAP and LIME are helping security analysts understand how the design is making predictions[5],[7],[8]. Reinforcement learning and learning-based Additionally, models are utilized. to handle imbalanced network traffic datasets and detect unknown attacks[8].

Existing systems for detecting intrusions still have a great deal of problems, like getting it sometimes not being able to understand new things being hard to make sense of and having trouble finding new kinds of attacks.

The intrusion detection systems used nowadays are capable of identifying signature-based online dangers, but they aren't as good at identifying zero-day threats.

3. Proposed Approach

The hybrid framework proposed in this project tries to overcome the current system's shortcomings. The suggested system is made in a way making it a more efficient and effective system for detecting zero day web attacks. The system is capable of precisely capturing zero day attacks, precisely web attacks. The idea of applying SMOTE technique on the network data has improved model learning

on the minority class. The system provides a hybrid framework including unsupervised and supervised techniques, rather than focusing on a single model. Various classification models are implemented in order to infer about the most reliable classification model that is suitable for clearly distinguishing between normal and attack traffic, reducing the chances of inappropriate prediction of false predictions i.e false positive rate. Due to the implementation of XAI methods, The model is competent in explain the decisions and is more transparent. The proposed system is designed in a way such that it is flexible enough to incorporate new models and techniques for future work.

System Architecture

Figure 3.1 explains the architecture of the proposed system which leads to efficient detection of zero day web attacks.

1. Data Input: CICIDS2017 dataset is collected for the examination of the network for the detection of the zero day web attack.
2. Data Preprocessing: This step is carried out to handle the collected dataset appropriately in order to avoid missing values, data cleaning and label encoding.
3. Data Preparation: Performed on the dataset enabling it to be efficiently used for detection and classification models.

4. Anomaly Detection: An unsupervised technique, used to detect anomalies present within the network data.
5. Classification Model: A supervised approach, which classifies the network data into attack class or normal class.
6. Evaluation: This enables to provide a clear understanding which model is
7. outperforming and is precisely able to detect the zero day web attacks.
8. Explainable AI: XAI methods provide a clear explanation about the model's decision, enhancing the transparency of the system.

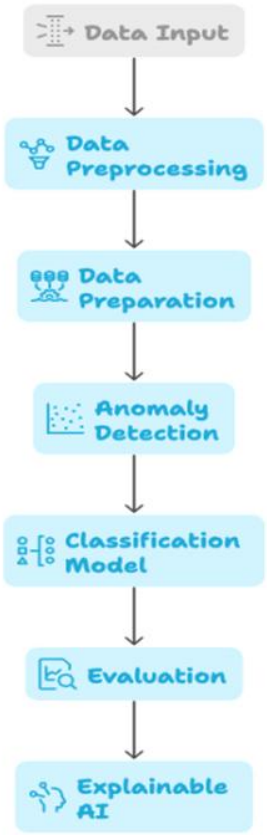

Figure 3.1: System Architecture

The block schematic of the architectural design illustrates the major components of the suggested hybrid framework for zero-day web attack detection.

Algorithm

INPUT: CICIDS2017 Dataset D

OUTPUT: Anomaly Labels

Step 1: Load the dataset D

Step 2: Preprocess the dataset D

Step a: Standardise the column names

Step b: Check for missing and NaN values

Step c: Convert Labels to Binary Values using Label Encoding

0 -> Normal

1 -> Attack

Step 3: Prepare the dataset D for model building

Step d: Split the dataset D into instruction and evaluation data

Step e: Balancing the normal and attack class of dataset D using SMOTE D’

Step f: Normalise the data and implement feature selection

- VarianceThreshold, considering 0.01 as the threshold value

Step g: Convert each and every values of dataset D to comparable scale for feature scaling

- StandardScalar, Mean=0 and Standard Deviation=1

Step 4: Anomaly Detection

Step h: Prepare Autoencoder model based on dataset D

Step i: Implement Model of Isolation Forest with dataset D

Step 5: Classification models

Step j: Build a Logistic Regression model using dataset D’

Step k: Build a Random dataset D' for the forest model

Step l: Build a Linear SVM dataset-based model D’

Step m: Build a XGBoost dataset D' in the model

Step 6: Evaluate and analyse the execution of the supervised models

Step 7: Implement Explainable AI techniques on the best performing model

Step n: Apply LIME method

Step o: Apply SHAP method

4. Methodology

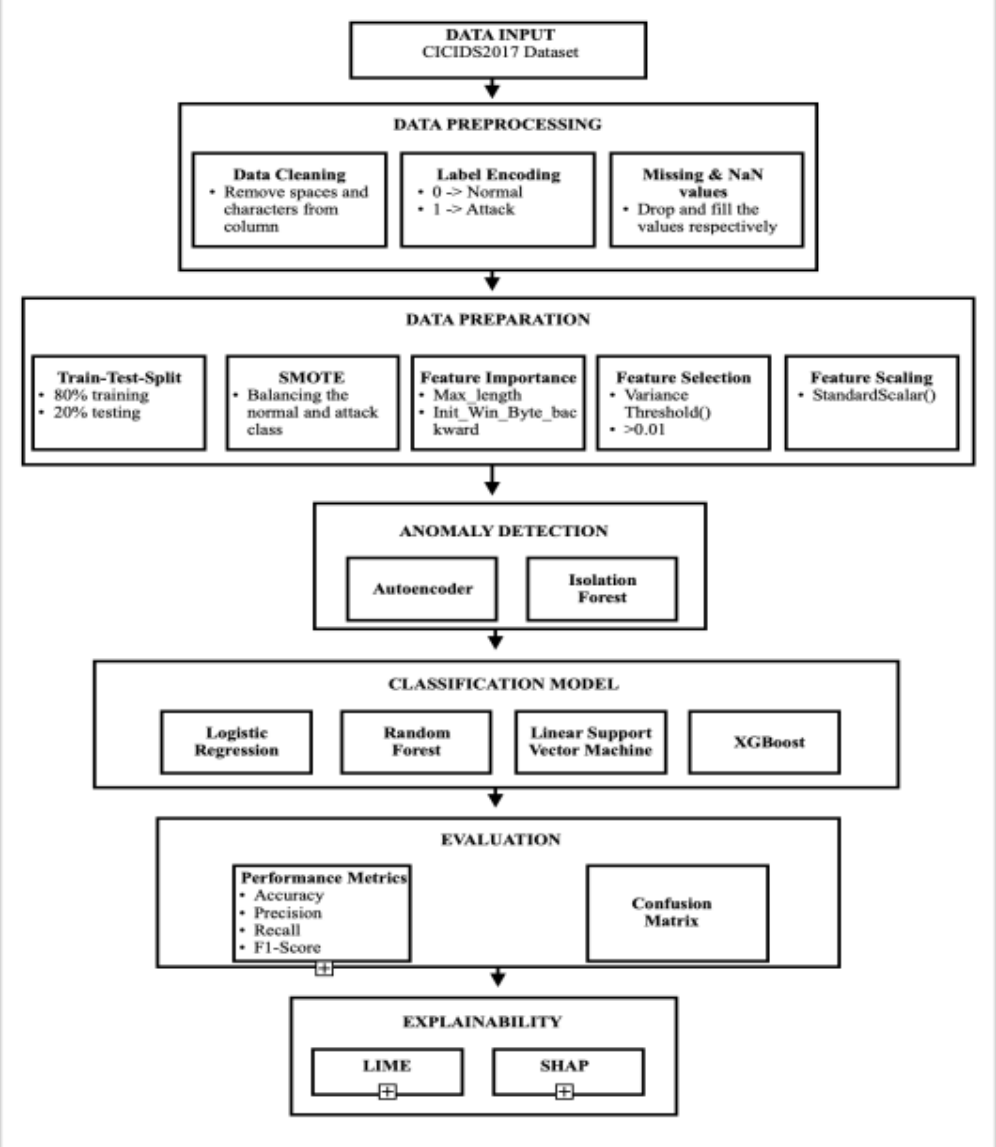

Figure 3.2 Proposed Hybrid Framework for Zero Day Web Attack Detection

Referring to the figure 3.2

Data Input: The dataset utilized for the network analysis is CICIDS2017 which is provided by the Canadian Institute for Cybersecurity (CIC) for the Intrusion Detection Systems (IDS). It consists of network data including normal and attack traffic of all types. The dataset has got 1,70,366 rows and 79 columns.

Data Preprocessing: It is the process where the raw data collected from the dataset is converted to a more clean and structured format making it suitable for further processing. It includes the steps like:

- Data Cleaning: This step helps to deal with the removal of unwanted space and characters within column names allowing standardizing The implementation of feature names.
- Label Encoding: Label Encoding helps in converting categorical data like normal and attack to numerical values like 0 and 1.

- Missing and NaN Values: Checking on missing and NaN values helps to verify the presence of any values that are missing or are mentioned as NaN within the dataset and dropping and filling them respectively.

Data Preparation: This is the most crucial step, as it deals with preparing the data in the suitable manner for the implementation of algorithms. The data preparation process includes the following steps:

- Train-Test Split: This method divides training and testing portions of the dataset in order to train the model and evaluate it on the unseen data respectively.
- SMOTE: Synthetic Minority Oversampling Technique is applied
- on the dataset for the purpose of balance the class imbalance. SMOTE basically uses KNN algorithm and creates synthetic samples of the attack class and balances the dataset so that the model to better effectively understand the assault class.
- Feature Importance: This method is useful for gaining the insights on which features are impacting the classification of network data.
- Feature Selection: This technique helps to only deal with relevant features. Variance Threshold method is employed for selecting the valid variables. It calculates the variance of every characteristic and discards the one with low value.
- Feature Scaling: Feature Scaling is done to be able to transform all the values of the features onto a similar scale. Standard Scaler method is employed in order to scale the values which converts the data into Mean=0 and Standard Deviation=1.

$$X_{scaled} = (X - means \div standarddeviation$$

Anomaly Detection: This method of unsupervised machine learning aids to detect anomalies occurring within the system. Since detecting zero day web attacks deals with capturing unknown attacks, anomaly detection is considered to be an appropriate method. To identify the machine learning model is best at identifying the abnormalities, two different kinds of models were used.

- Autoencoder: An autoencoder model for deep learning is mostly used for anomaly identification. This neural network framework thoroughly studies the normal behaviour of network usage enabling it easy to detect any kind of deviations from its ordinary actions. It

basically tries to reassemble the incoming information and then calculates the reconstruction error. The reconstruction error gives the difference between the input data and the output generated by the autoencoder model. If the error is greater than the threshold assumed, then the specific sample is regarded as belonging to the attack class.

- Isolation Forest: This tree based technique which rather than studying the normal actions of the whole network dataset simply isolates the information that act as anomalies from the normal data points using random partitioning method by building random forest tree model. This facilitates the direct isolation of the data points. rather than studying the behavior of traffic data.

Classification Models: This module includes developing models for machine learning under supervision classification using multiple algorithms. This is performed in order to study which classification model accurately predicts the zero day web attack. These classification models use the SMOTE data to train the model and classify whether the sample considered belongs to normal class or attack class. 0 indicates the normal class and 1 indicates attack class.

- Logistic Regression: Logistic Regression is a model for linear classification that predicts the probability of attack and works perfectly in the case of binary classification. This approach serves as a foundation for assessing the effectiveness of the other advanced classification models developed.
- Random Forest: This model refers to creating multiple decision trees and

then combining the outcomes of all the trees to achieve the final prediction. These types of the models that work best for classification or regression type of problems. The Random Forest model generates many trees, each of which is trained at random using the data. Every tree makes the prediction based on the training completed and the majority of the predicted value is considered as the output.

- Linear Support Vector Machine: primarily utilized for regression and classification. Support Vector Machine's concept is to use the optimal boundary to divide the data into distinct classes. The term "hyperplane" describes this border. If the boundary is linearly able to separate the data cleanly into different classes, if the hyperplane is simply a straight line or flat, then The term "Linear Support Vector Machine" refers to this type of model. Linear Support Vector Machine is different from Support Vector Machine. is advantageous as it is computationally cheaper While interacting with enormous data.
- XGBoost: This model refers to Extreme Gradient Boosting. It is a kind of model suitable for classification and regression. XGBoost is the advanced form of Gradient Boosting. Unlike Random Forest that creates independent multiple trees, XGBoost creates the trees in a sequential manner. Every time a new tree is created, it rectifies the errors present in the previous trees providing a scope of improvement at every step. As the model progresses, It only concentrates on the mistakes, resulting in a fine-tuning of the model to make accurate predictions. Hence, XGBoost is regarded as the most advanced and powerful between the other machine learning algorithms.

Evaluation: This phase enables examination of the suggested system's performance and evaluation of the models' accuracy in identifying zero-day web attacks.

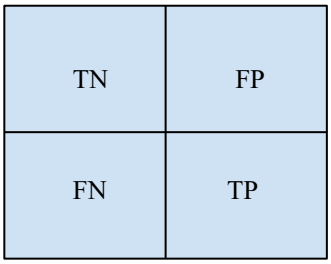

Figure 3.3 General Layout of Confusion Matrix

Performance metrics:

The parameters that are typically considered to assess the machine learning models' performance are included in performance metrics. The following are the evaluation criteria that were used:

Accuracy: This is a gauge of the model's accuracy. It calculates how many predictions the model correctly calculates among the total samples.

$$Accuracy = (TP + TN) \div (TP + TN + FP + FN)$$

Precision: This gives the measure of how many predictions are actually correct among the total predictions made.

$$Precision = TP \div (TP + FP)$$

Recall: Recall is a measure that gives the percentage of real attacks that the model anticipated. Additionally, it is referred to as detection rate.

F1-score: It is the precise harmonic mean and recall, creating a balance between both the parameters.

$$F1 - score = 2 * [(Precision * Recall) \div (Precision + Recall)]$$

Where,

TP = True Positives (Correctly Predicted Attack Traffic)

TN = True Negatives (Correctly Identified Normal Traffic)

FP = False Positives (Normal Traffic falsely classified as Attack Traffic)

FN = False Negatives (Attack Traffic falsely classified as Normal Traffic)

Confusion Matrix: A confusion matrix is plotted To evaluate the efficacy of the classification models for comparing the actual and predicted labels. It gives An illustration of the labels enabling easy computation of performance metrics.

Explainability: This module includes Explainable AI (XAI) methods that are advantageous for improving the suggested system's transparency. With the help of XAI techniques, the system produces an explanation of every predicted output describing why the model predicts “this” output. It is also useful for gaining the insights on the features that are influential in detecting the zero day web attacks thus increasing the trust on the proposed system and assisting network analysts to make quick decisions in case of attack detection. These Explainable AI techniques are implemented after building the machine learning models to interpret the classification of network traffic and to gain insights about the impactful features that made the model to predict the attack class for that particular network instance.

SHapely Additive exPlanations: This XAI technique is widely known as

SHAP. It is known as a game theory based model. SHAP provides the explanation for the output predicted by the machine learning model by assigning values to each feature contributing in predicting the output. SHAP gives the information on both global and local importance using summary plot and waterfall plot respectively.

Local Interpretable Model-agnostic Explanations: This XAI technique is generally called LIME. As the name implies, this method is only helpful in explaining the local importance i.e it is able to provide the explanation of the predicted output for a single instance and not regarding the entire dataset, unlike global importance in SHAP.

5. Results and Discussion

This section speaks about the end result and gives the detailed insight of the output achieved by the proposed system.

While processing the data, It has been noted that the CICIDS2017 dataset has got 1,70,366 rows and 79 columns of data. Among which 1,68,186 rows are benign traffic, 1507 are web attack brute force, 652 are web attack XSS and 21 are web attack SQL injection. The data cleaning yields 1,70,231 valid rows They are beneficial for further processing. Train_Test_Split splits the dataset into nearly 1,36,184 rows of training data and 34,047 testing data, both including 78 columns. Before applying SMOTE technique, the dataset includes 1,34,440 values for normal class and only 1,744 values for attack class creating a huge imbalance, but SMOTE balances the dataset by generating synthetic samples making certain that the two classes' values are equal, or 1,34,440 values for every class as depicted in the below figures 4.1(a) and 4.1(b).

Fig: 4.1 Class Distributions. (a) Before SMOTE (b) After SMOTE

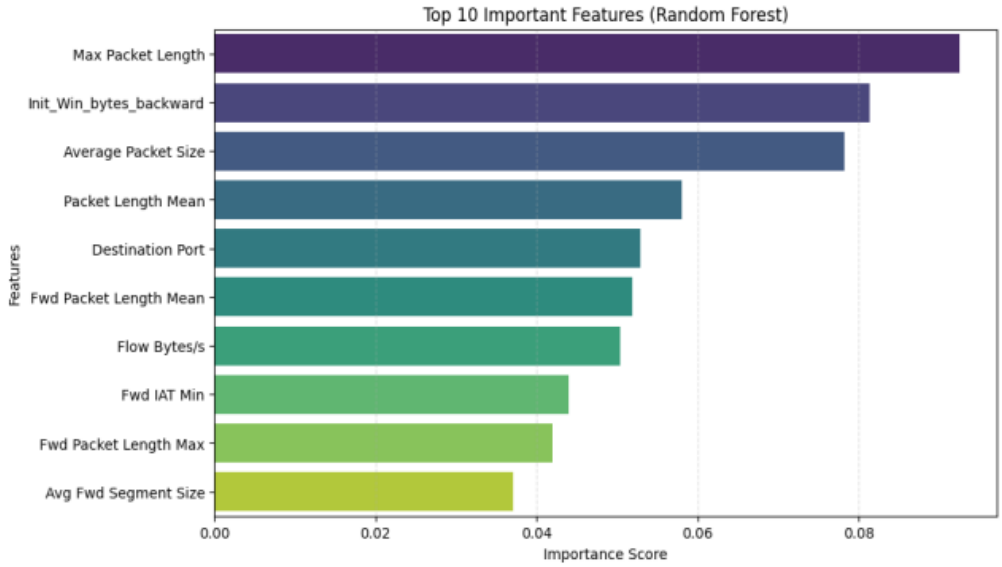

Fig: 4.2 Feature Importance Plot

The feature selection process using VarianceThreshold method nearly eliminates 12 features allowing only 66 features to be considered for further processing. The scores and graphical representation of the feature importance is as depicted in the below figure 4.3.

Table: 1 Feature Importance

Feature Number	Feature	Importance
36	Max Packet Length	0.092608
54	Init_Win_bytes_backward	0.081331
45	Average Packet Size	0.078198
37	Packet Length Mean	0.058016
0	Destination Port	0.052842
8	Fwd Packet Length Mean	0.051939
14	Flow Bytes/s	0.050378
24	Fwd IAT Min	0.043956
6	Fwd Packet Length Max	0.041975
46	Avg Fwd Segment Size	0.037071

Using the preprocessed data and training dataset, multiple machine learning models like Autoencoder, Isolation Forest for Detecting Anomalies and Logistic Regression, Random Forest, Linear SVM

and XGBoost are developed. The Analyzing the overall effectiveness the above mentioned models can be analyzed using the following diagram (figure 4.2)

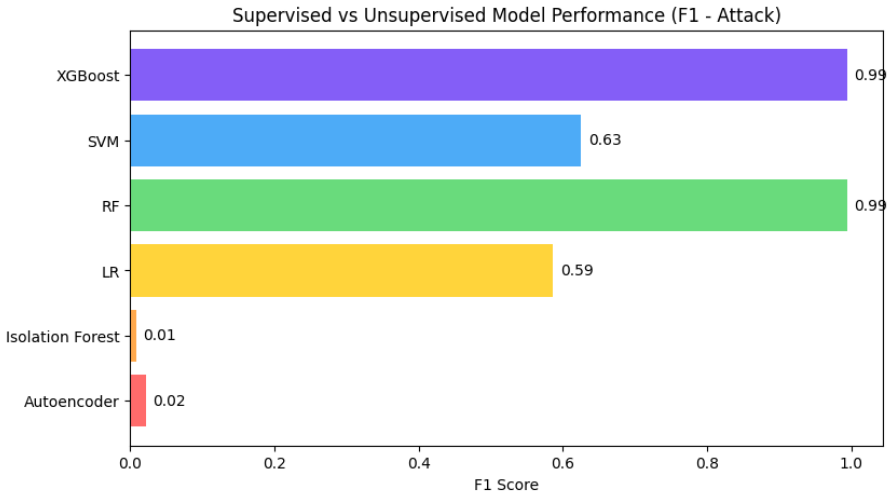

Fig: 4.3 Model Performance Supervised vs Unsupervised

From the figure 4.3, it It is evident that among all the implemented machine learning models, F1-Score for XGBoost and Random Forest model is the highest, demonstrating that these designs are real. performing well facilitating the correct detection of zero day web attacks in this experiment. But the F1-score of both the models are making it difficult to choose which model is the best. In order to conclude about the finest algorithm suitable for this project, it is necessary to study the details of performance metrics and confusion matrix of every model implemented. To comprehend the confusion matrices of the implemented models, the following prerequisites are important. The below figure illustrates the layout Consider the confusion matrix

produced by different models for machine learning with distinct labels, as seen in figure 3.3.

As the detection of zero day web attack deals with unlabeled data and is likely to fall under the category of anomaly detection. Anomaly detection includes the autoencoder model and isolation forest model.

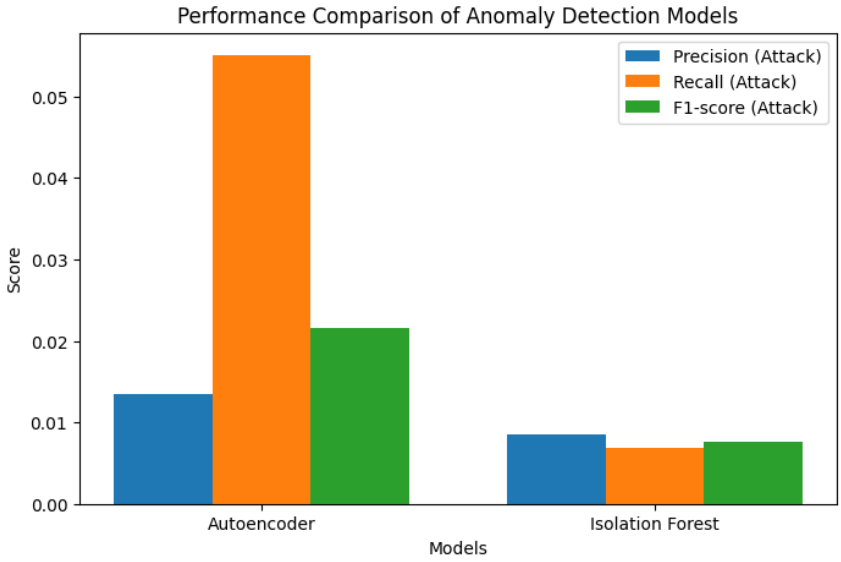

Figure 4.4 Performance Comparison of Anomaly Detection Models

The above figure serves as a summary how well the unattended gadget performs and what methods were used to detect ambiguity. It might be true that the total performance of the company isolation forest is disappointing and the scores of the metrics in autoencoder are very unstable.

The performance of the autoencoder model can be observed from the following reconstruction error plot (figure 4.5), confusion matrix (figure 4.6) and performance metric chart. The below plot gives a clear description regarding the poor performance of the autoencoder model. The X-axis indicates the Reconstruction error values and Y-axis indicates the Number of samples. The red dashed line indicates the threshold value, where the left region of the line indicates low error and right region indicates higher error. It is apparent that the model is efficiently reconstructing the normal data as the majority of the normal data samples are concentrated at 0, but the plot is failing to describe the attack class. This demonstrates that the concept is skilled. not able to differentiate between the normal and attack class due to their similar behaviour and also the overlapping threshold line and blue bar of reconstructed normal class describes the high probability of misclassification leading to rise in the quantity of false negatives and false positives.

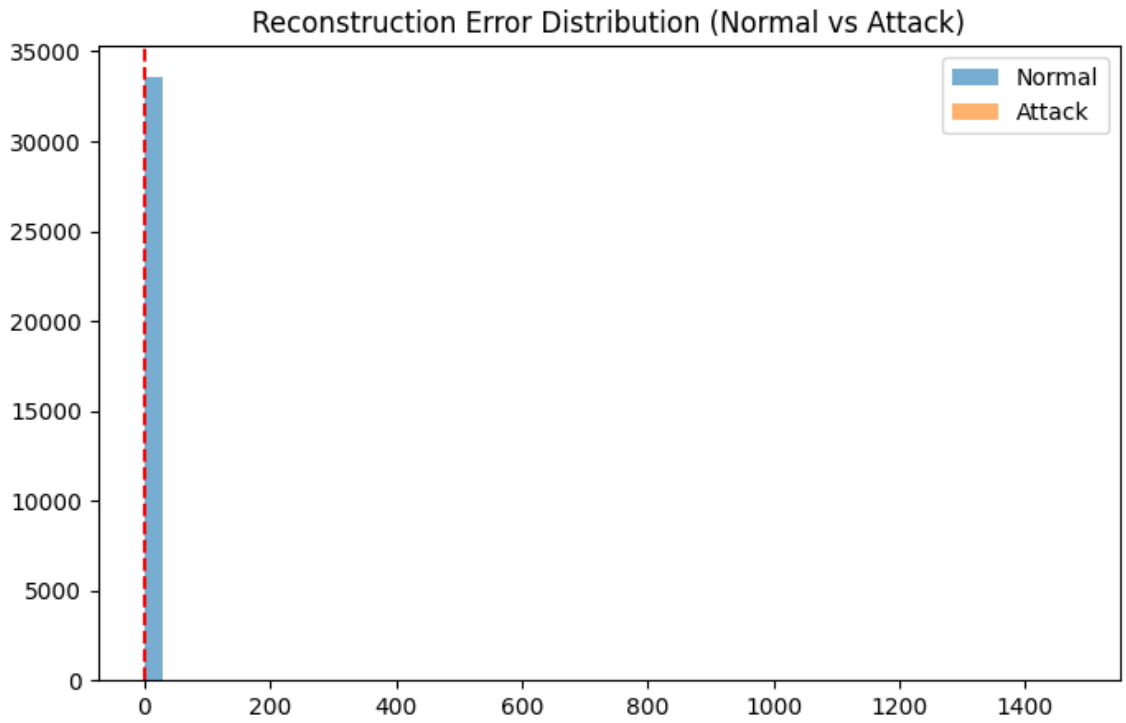

Figure 4.5 Distribution of Reconstruction Error

Since the output of the autoencoder model is converted to the binary format, a confusion matrix is generated for the same as displayed in the figure 4.6. Referring to the figures 3.3 and 4.6, it can be assessed that 412 numbers of attack traffic are being detected as normal and 1785 numbers of normal traffic are being detected as attack. Also, the below chart gives a clear description regarding the classifier model's performance. It demonstrates how accurate the type is. is 93.54%, but it is also observed that for the attack class the precision is only 0.01. This huge difference is unavoidable and hence it proves that the Model autoencoder is not fit for detecting the zero day web attack for the current experimentation.

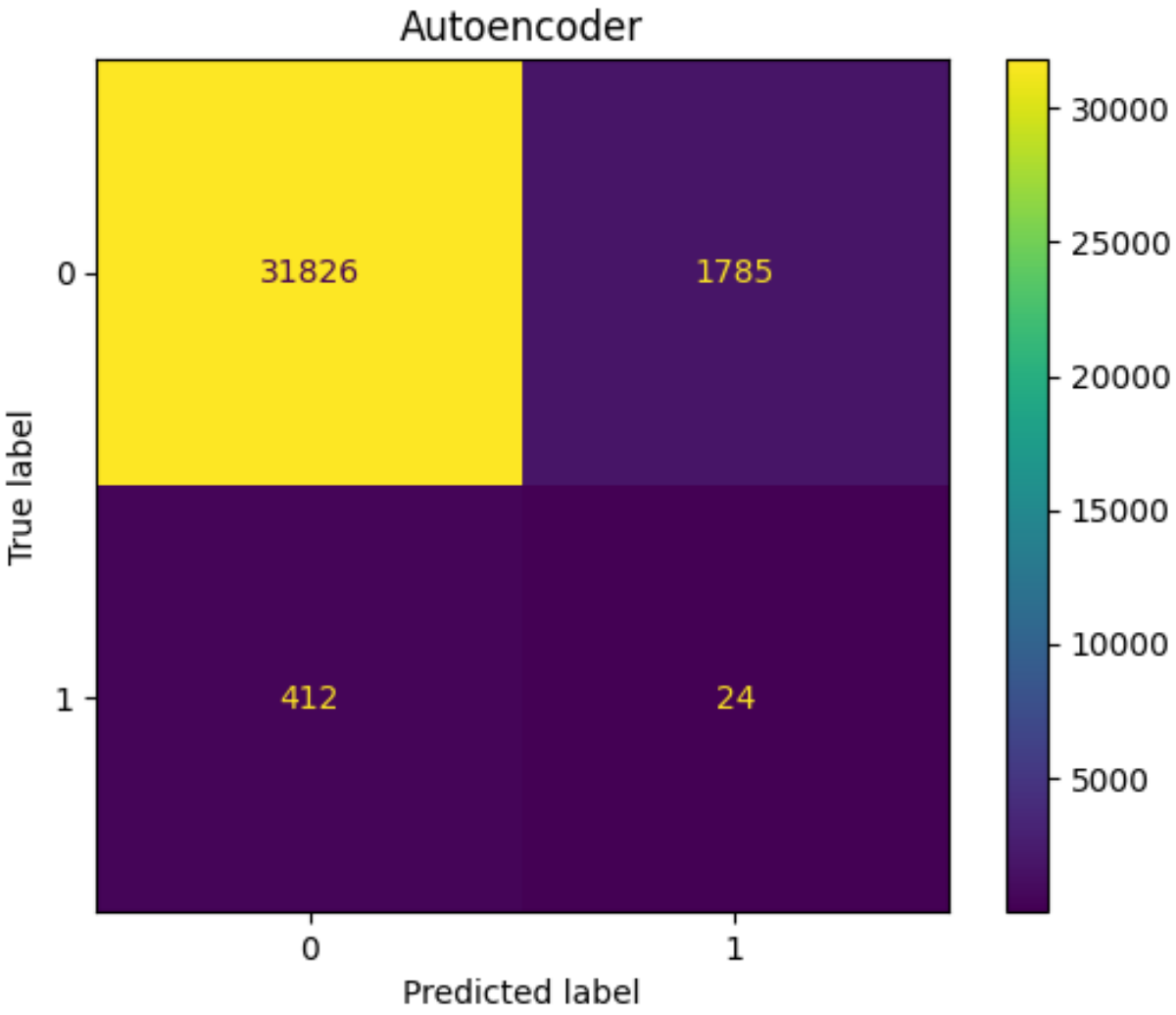

Figure 4.6 Autoencoder Confusion Matrix

Table 2: Performance of Autoencoder model Autoencoder Accuracy: 0.9354715540282551

	precision	recall	f1-score	support
0	0.99	0.95	0.97	33611
1	0.01	0.06	0.02	436
accuracy			0.94	34047
macro avg	0.50	0.50	0.49	34047
weighted avg	0.97	0.94	0.95	34047

Since the autoencoder model failed, the other model implemented for anomaly detection is isolation forest. This model's performance can also be assessed by the score distribution plot of the model, confusion matrix and the performance indicators displayed in the below figures 4.7, 4.8.

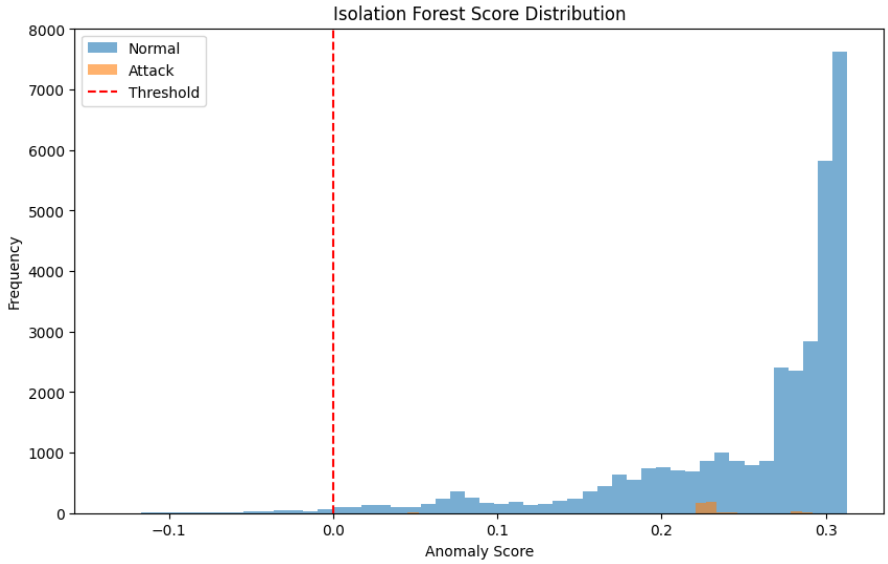

Figure 4.7 Score Distribution of Isolation Forest

The above figure illustrates the sample distribution of community traffic that occurs gives insights on the anomaly score. The red dashed line indicates the threshold. The left region of the threshold i.e the lowest scores indicate the anomalies and the right region of threshold having higher scores indicate the normal traffic. It is evident that though the isolation forest The model can identify the attacks (orange band), it is ineffective in clearly differentiating them from the normal traffic. Referring to the figures 3.3 and 4.8, it can be assessed that 433 numbers of attack traffic are being detected as normal and 351 numbers of normal traffic are being detected as attack. This demonstrates that the false positive and false negative rates are high. It can be clearly assessed that the precision, recall and f1-score values of the attack class are 0.01, highlighting the fact the isolation forest is the poorest performing model as compared to the autoencoder model.

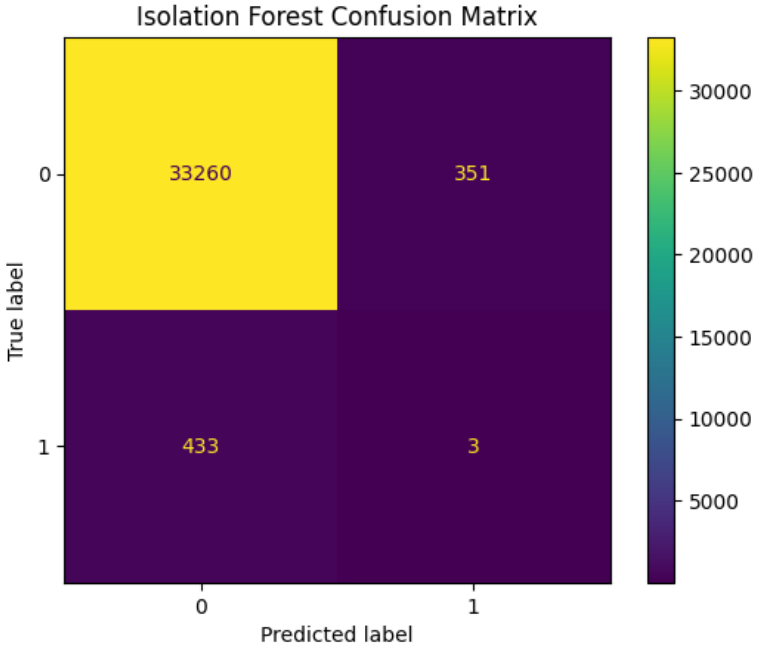

Figure 4.8 Isolation Forest Confusion Matrix

Performance of Isolation Forest model

Accuracy: 0.976973007900843

Classification Report:

Table 3: Performance of Isolation Forest model

	precision	recall	f1-score	support
0	0.99	0.99	0.99	33611
1	0.01	0.01	0.01	436
accuracy			0.98	34047
macro avg	0.50	0.50	0.50	34047
weighted avg	0.97	0.98	0.98	34047

Therefore, it can be interpreted that even an isolated forest is not a suitable model for the suggested intrusion detection system. Consequently, it can be strongly concluded that detecting zero day web attacks using anomaly detection techniques is not effective for the CICIDS2017 dataset. techniques for machine learning supervised are utilized to build different classification models in order to study the efficiency of the prototype in classifying the normal and malicious behaviour of the network traffic.

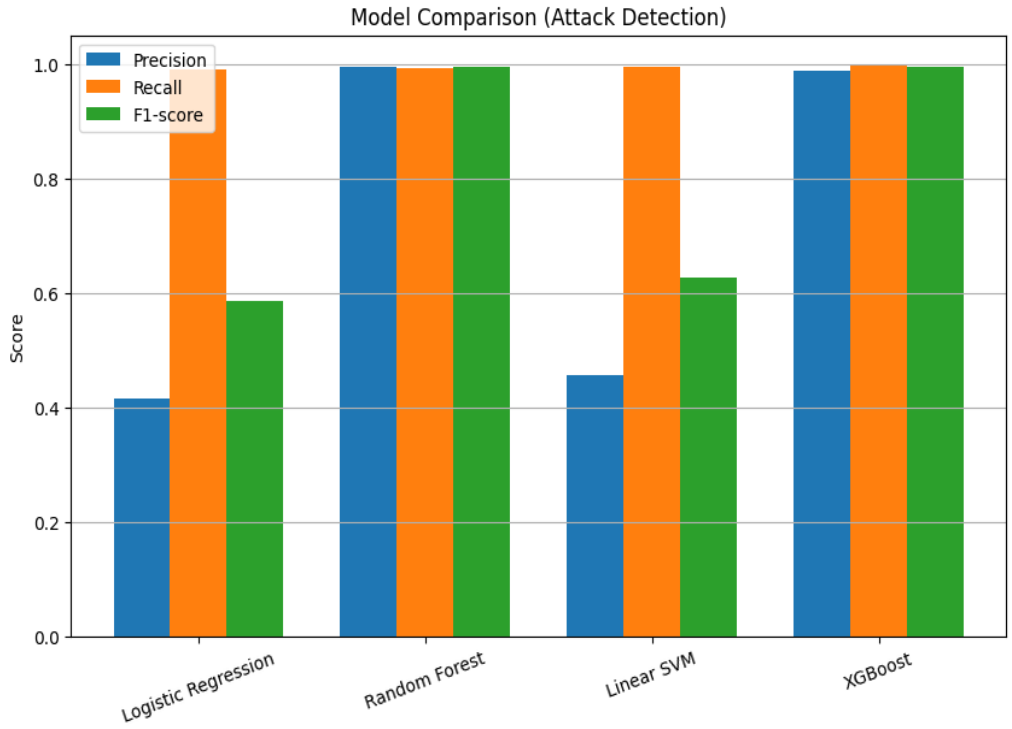

Figure 4.9 Performance Comparison of Classification Models

The above figure gives a complete understanding regarding the performance of the classification models and It might be observed that Random Forest and XGBoost model are able to capture the anomalous behaviour of the network traffic. The below is the summary chart of the Linear Regression, The classification model's performance

may be evaluated. XGBoost algorithms describing their accuracy, precision, and understand and f1-score. This enables us to easily identify the top performing model.

Table 4: Performance of Classification Models

	Model	Accuracy	Precision (Attack)	Recall (Attack)	F1-Score (Attack)
0	Logistic Regression	0.9821	0.4162	0.9908	0.5862
1	Random Forest	0.9999	0.9954	0.9931	0.9943
2	Linear SVM	0.9847	0.4559	0.9954	0.6254
3	XGBoost	0.9999	0.9887	1.0000	0.9943

From the above chart, it is noticed that XGBoost is the most superiorly performing model as compared to additional classification algorithms that are accurate 99%, precision of 98.87%, recall of 99.99% ~ 100% and f1-score of 99.43%.

The classification model's performance may be evaluated. analysed by the confusion matrices and the performance metrics chart as given below.

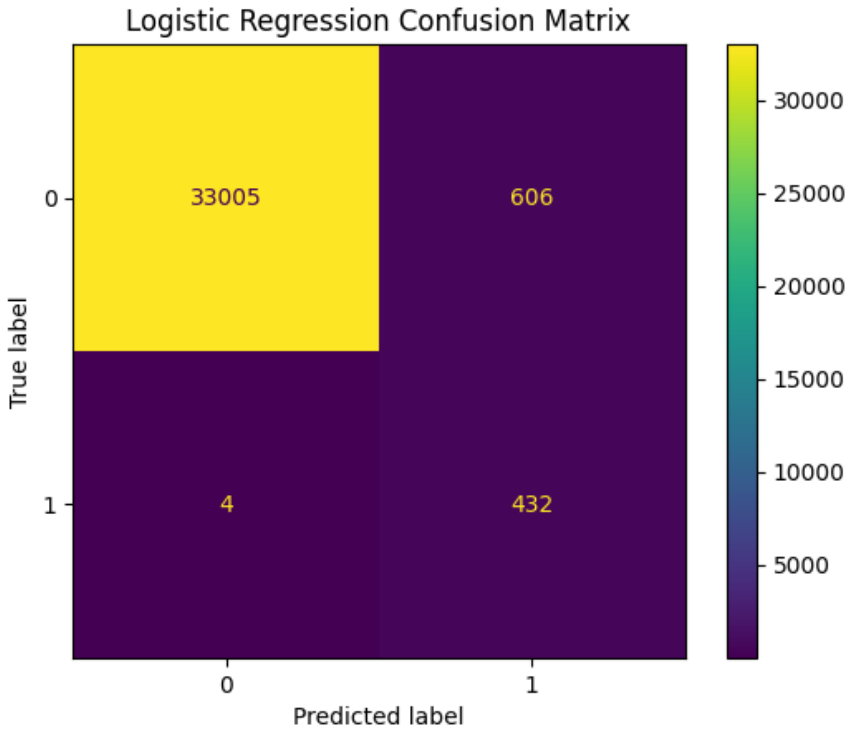

Figure 4.10 Logistic Regression Confusion Matrix

Referring to figure 3.3 and considering the above figure, It is clear that the amount of the most hazardous parameter, i.e the false negative, is only 4. But still Logistic Regression is an inefficient model because the false positive number is 606, which makes the suggested system very annoying. Additionally, the logistic regression efficiency measures mentioned describes that the precision of attack class is merely 0.42 and f1-score is 0.59. Thus, it can be stated that Logistic Regression is not a suitable classification model to be considered.

The Logistic Regression model's performance

LR Accuracy: 0.982083590331013

Table 5: Performance of Logistic Regression model

	precision	recall	f1-score	support
0	1.00	0.98	0.99	33611
1	0.42	0.99	0.59	436
accuracy			0.98	34047
macro avg	0.71	0.99	0.79	34047
weighted avg	0.99	0.98	0.99	34047

Below is The matrix of perplexity in a random forest model. Referring to figure 1, It may be inferred that the false both favorable and unfavorable rates are decreased to 2 and 3 respectively. This indicates that the Random Forest classification model is appropriate for classifying the attack and normal class.

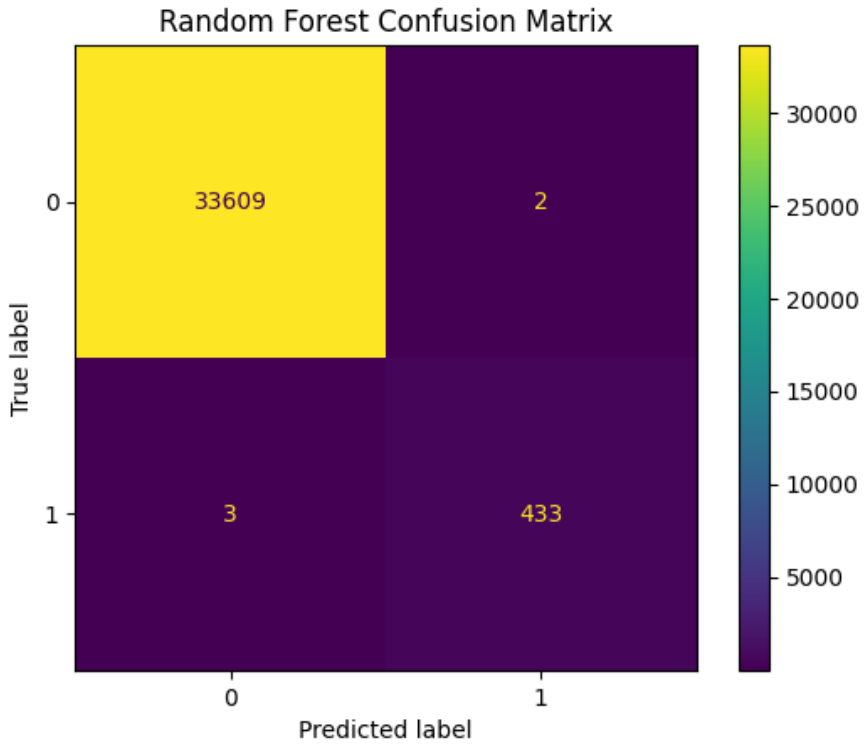

Figure 4.11 Linear SVM Confusion Matrix

The linear SVM's confusion matrix is as seen in the figure. model. It is observed that even if the model has only two samples that meet the false positive and false negative criteria. has got 518 samples, making the system ineffective. Also, The model's performance metrics demonstrate is able to achieve only 0.46 of precision and f1-score of 0.63 for the

attack class which is too high to neglect. This shows that this classification For the suggested system, the model is unreliable.

Performance of Linear Support Vector Machine model

Linear SVM Accuracy: 0.9847269950362734

Table 6: Performance of Model of a Linear SVM

	precision	recall	f1-score	support
0	1.00	0.98	0.99	33611
1	0.46	1.00	0.63	436
accuracy			0.98	34047
macro avg	0.73	0.99	0.81	34047
weighted avg	0.99	0.98	0.99	34047

The matrix of misunderstanding for the XGBoost model is as displayed in the below figure 4.12. The issue of both false negatives and false positives has completely disappeared in this case. There are 0 samples for false negatives and 5 samples under false positives. Also, measures for the model's performance describes that it has achieved precision of 0.99, recall of 1.00 and f1-score of 0.99. Thus, it can be stated XGBoost is the outperforming model, as the model is clearly able to distinguish between the attack class and normal class and is effective enough to detect the zero day web attacks.

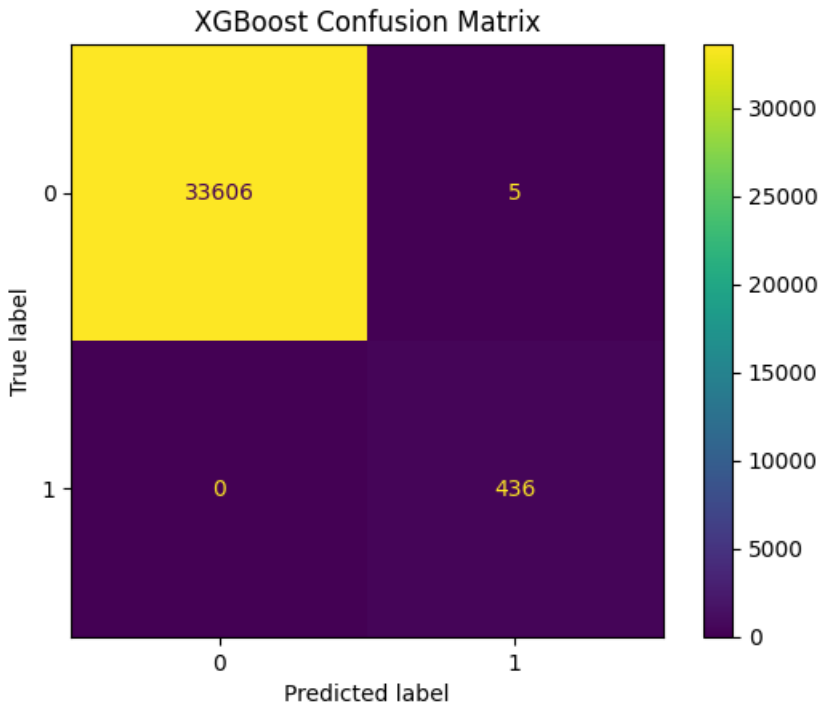

Figure 4.12 XGBoost Confusion Matrix

Performance of SVM model

XGB Accuracy: 0.999853144183041

Table 7: Performance of Linear SVM model

	precision	recall	f1-score	support
0	1.00	1.00	1.00	33611
1	0.99	1.00	0.99	436
accuracy			1.00	34047
macro avg	0.99	1.00	1.00	34047
weighted avg	1.00	1.00	1.00	34047

By analysing the execution of all the implemented machine learning models, it can be strongly stated that tree based classification models i.e Random Forest and XGBoost algorithms were suitable for proposed system, but XGBoost is the model that outperformed serving the purpose of detecting and classifying the zero day web attacks for this experimentation.

Table 8: Feature Contribution by LIME

	Feature Condition	Contribution
0	Init_Win_bytes_backward <= -1.00	-0.103524
1	min_seg_size_forward <= 20.00	-0.100070
2	Destination Port <= 80.00	0.096075
3	Fwd PSH Flags <= 0.00	-0.047211
4	Bwd IAT Std <= 0.00	0.039817
5	Idle Mean <= 0.00	-0.029653
6	Active Min <= 0.00	-0.029344
7	Init_Win_bytes_forward <= 20.00	-0.027219
8	0.00 < Max Packet Length <= 113.00	-0.026196
9	Flow IAT Max <= 23246.75	0.024980

The above chart is Manufactured by the LIME technique of Explainable AI that provides the explanation regarding the features contributing the prediction of network instances globally. The negative values describe that those features are contributing towards the normal class while the positive values are in the support of the attack class. LIME basically provides details about the contribution of features for the overall dataset i.e it gives information over the globally important features. It fails to specify about the single network instance to be considered for the prediction of output. Hence SHAP technique is also used besides LIME to be able to acquire the insights on both global and local features.

The below figure 4.13 represents the summary plot of SHAP technique of Explainable AI generated for the XGBoost model which gives the details of global importance. The X-axis denotes the SHAP values and the Y-axis

denotes features that have their contribution in predicting the output. In the event that the feature score is towards positive SHAP value, it indicates the attack and if the score is moving towards negative SHAP values, then it indicates normal traffic. The color Red denotes both high and low feature values. It is evident from the above chart that the features Init_Win_Bytes_backwards, Destination Port, min_seg_size_forward and Fwd IAT Min possess the highest contribution in predicting the output of the framework.

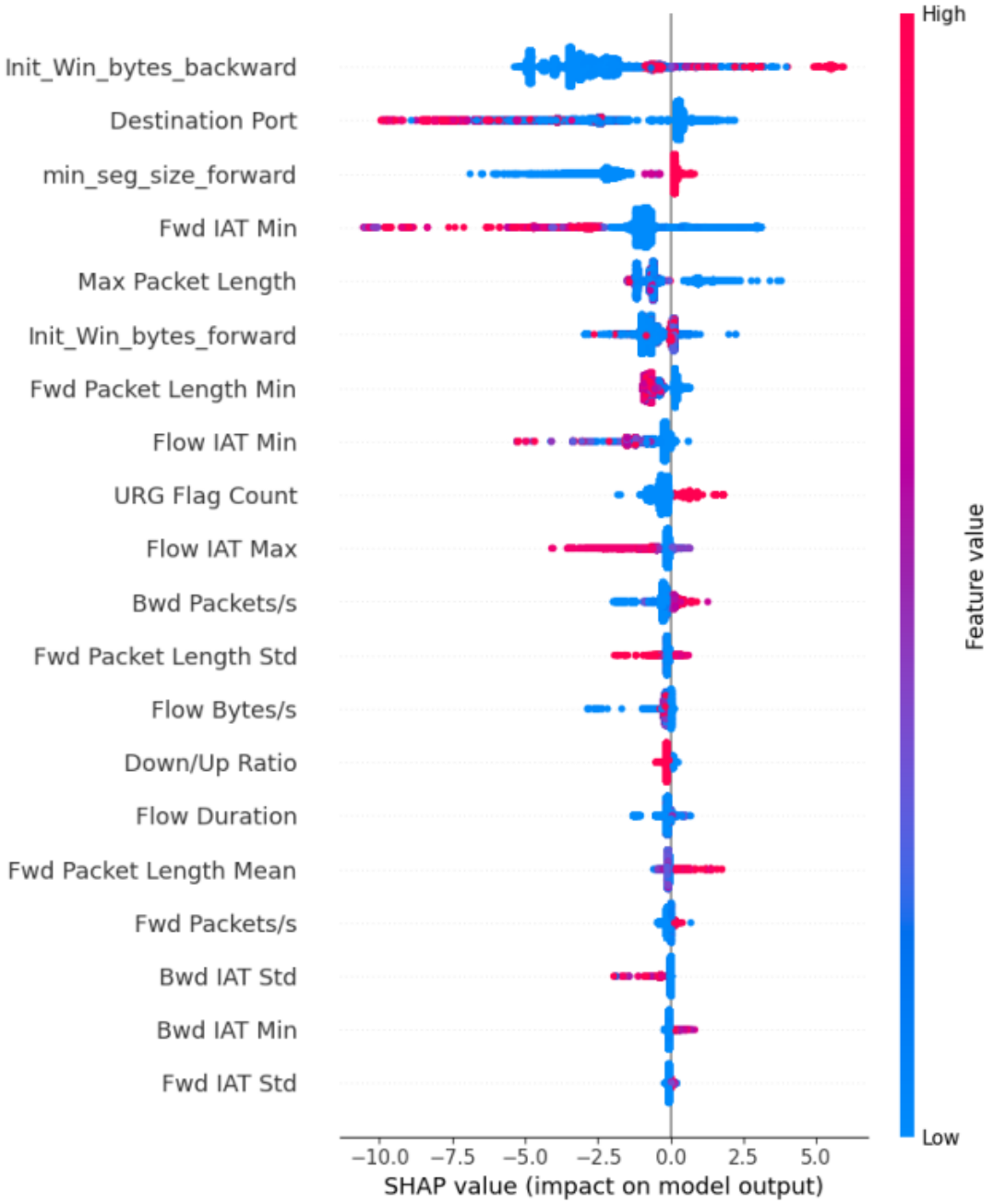

Figure 4.13 Global Feature Importance of SHAP

The figure 4.14 gives the details of strongly influencing features for a single sample of the network traffic data for the prediction of the output. The graph is called the waterfall plot of SHAP method that explains about the contributing features for a single sample prediction. This graph speaks about the local importance unlike the summary plot that describes the global features. E[f(X)] is referred to as base value i.e the starting point of the prediction process

before considering the contribution of features and f(X) is the final prediction value that is calculated after considering all the influencing features. The large negative f(X) denotes that the specific sample is favouring one particular class.

The below figure is the waterfall plot of a network instance belonging to the normal class. The blue bars represent that the model has classified the sample to be behaving normally and feature 54 and 26 is contributing majorly for the classification.

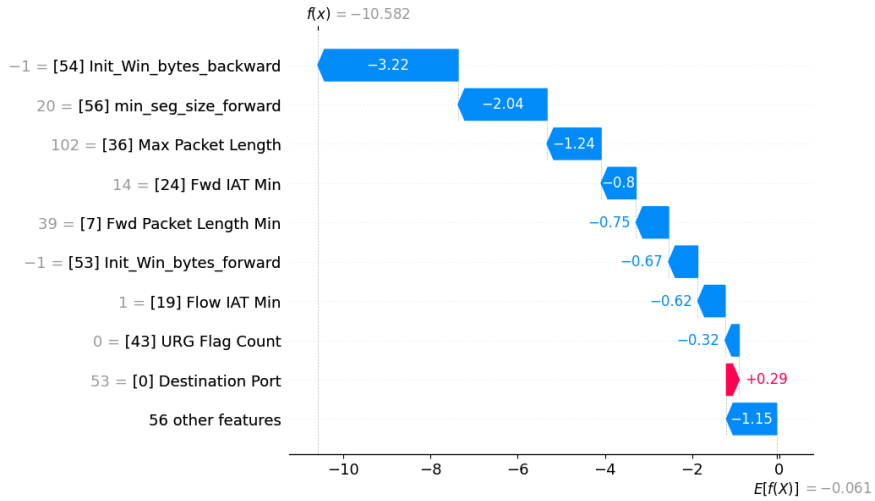

Figure 4.14 Waterfall plot for Normal Sample

Similarly, the below figure is the waterfall plot of a network instance belonging to the attack class. The red bars represent that the model has classified the sample to be malicious and feature 54 and 24 is contributing majorly for the classification.

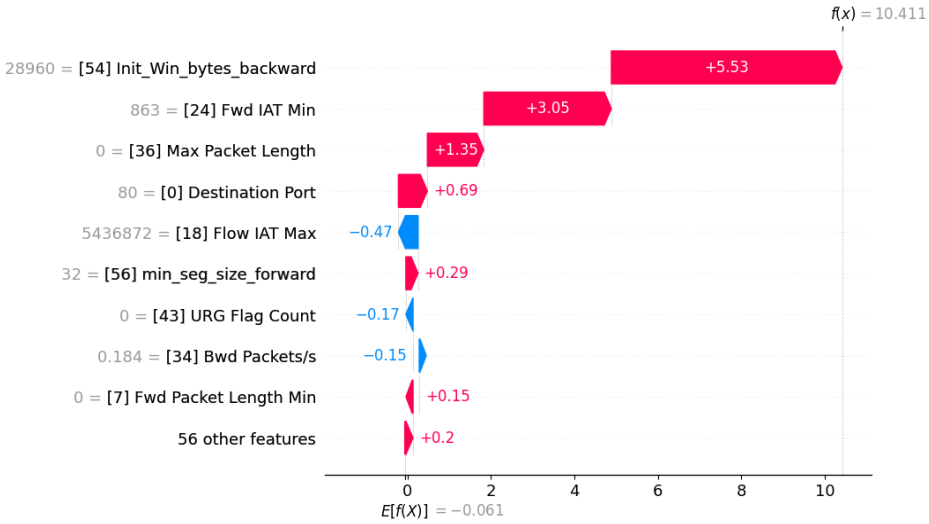

Figure 4.15 Waterfall Plot of Attack Sample

By the overall examination of the results achieved by the experimentation Thus, it may be said that though anomaly detection models are theoretically strong enough to capture the anomalous behavior of the network traffic to capture the unseen attacks, but due to its inability of differentiating between normal and attack class for CICIDS2017 dataset, it serves as a limitation. Therefore, classification models were implemented and the XGBoost model was observed to be obtaining noticeably improved outcomes for this experimentation whose prediction capability was further explained by LIME and SHAP methods, thus serving the purpose of the proposed system.

6.Conclusion

The paper presents about developing an intrusion detection system capable of detecting zero day web attacks by following a hybrid approach of artificial intelligence techniques i.e including both unsupervised and supervised algorithms and implementing the model with explainable AI methods.

The process began with uploading the CICIDS2017 dataset and handling it with multiple data preprocessing techniques. Later the dataset was prepared using various methods like splitting the dataset for training and testing, balancing the

dataset using SMOTE, checking out on feature importance and feature selection and carrying out feature scaling methods. For anomaly detection, autoencoder and isolation forest technique was developed and for classification, logistic regression, random forest, linear support vector machine and xgboost algorithms were implemented. By analysing the performance of all the implemented models, it can be observed that xgboost is the outperforming model suitable for correctly classifying normal and attack traffic of the network data. Later for enhancing the transparency of the system, explainable AI methods like LIME and SHAP are implemented on the xgboost model for explaining the contribution of features for predicting the output.

The paper provides the description of the proposed intrusion detection system which includes the combination of machine learning and explainable AI techniques in order to detect the unknown web attacks. The future scope includes loading more diversified data, implementing advanced deep learning and explainable AI techniques and considering the idea of multi-class classification for enhanced functionality and usability.

Dispute of Interest: The authors state that there is no potential conflict of interest.

REFERENCES

1. Tang, R., Yang, Z., Li, Z., Meng, W., Wang, H., Li, Q., Sun, Y., Pei, D., Wei, T., Xu, Y., & Liu, Y. (2020). ZeroWall: Detecting zero-day web attacks through encoder-decoder recurrent neural network. In Proceedings of the IEEE International Conference on Communications. IEEE.
2. Bhatnagar, M., Rozinaj, G., & Yadav, P. K. (2022). Web intrusion classification system using machine learning approaches. In Proceedings of the IEEE International Conference on Cyber Security and Protection of Digital Services. IEEE.
3. Mearaj, N., & Wani, M. A. (2023). Zero day attack detection with machine learning and deep learning. In Proceedings of the IEEE International Conference on Intelligent Systems and Security. IEEE.
4. Neha Sharma, Mayank Swarnkar, and Biltu Mondal (2024). WebWall: Zero-Day Attack Detection in Web Traffic Using Spatial Graph Neural Network. In Proceedings of the IEEE International Conference on Advanced Networks and Telecommunications Systems IEEE. DOI: 10.1109/ANTS63515.2024.10898287
5. Dr. C.V. Suresh babu, Surendar V, and Subhash S. (2024). Web Based Deep Learning Model for Zero Day Vulnerability Detection using FastAPI. In Proceedings of the IEEE nternational Conference on Advances in Data Engineering and Intelligent Computing Systems. IEEE. DOI: 10.1109/ADICS58448.2024.10533540
6. Vahid Babaey and Hamid Reza Faragardi (2025). Detecting Zero Day Web Attacks with an Ensemble of LSTM, GRU and Stacked Autoencoders. MDPI. DOI: https://doi.org/10.3390/computers14060205
7. Deepa Krishnan, Swapnil Singh, and Vijayan Sugumaran (2025). Explainable AI for Zero-Day Attack Detection in IoT Networks using Attention Fusion Model. Springer Nature. DOI https://doi.org/10.1007/s43926-025 00184-8
8. Khorshed Alam, Md Fahad Monir, Md Junayed Hossain, Mohammad Shorif Uddin, and Md. Tarek Habib (2025). Adaptive Defense: Zero-Day Attack Detection in NIDS with Deep Reinforcement Learning. In Proceedings of IEEE. DOI:10.1109/ACCESS.2025.3585445
9. Mujeeb Ur Rehman, Margaret Zital, Muhammad Abrar, Muhammad Kazim, and Sohail Khalid (2025). Zero Day Attack Detection System using Autoencoders and Isolation Forest: An unsupervised Machine Learning Approach. Springer Nature.
10. Subrat Jena, Ashutosh Soni, Jayanti Rout, Surendra Kumar Nanda (2024). Zero-Day Exploits in Web Applications: Detection, Response, and Prevention. In Proceedings of the IEEE 3rd International Conference for Advancement in Technology IEEE. DOI:10.1109/ICONAT61936.2024.10774961

11. Nyashadzashe Tamuka, Khulumani Sibanda (2024) A Novel Real-time Web Based Intrusion Detection System (IDS) (2024). In Proceedings of the IEEE 4th International Multidisciplinary Information Technology and Engineering Conference IEEE. DOI: 10.1109/IMITEC60221.2024.10850945
12. Jonas Herskind Sejr, Arthur Zimek, Peter Schneider-Kamp (2020) Explainable Detection of Zero Day Web Attacks. In Proceedings of the IEEE 3rd International Conference on Data Intelligence and Security. DOI: 10.1109/ICDIS50059.2020.00016
13. Temitope Damilola Elijah, Nafeeul Alam Walee, Atef Mohamed (Shalan) (2025) AI-Based Detection of Zero-Day Exploits: A Framework. In Proceedings of the IEEE Twelfth International Conference on Intelligent Computing and Information Systems. DOI: 10.1109/ICICIS66182.2025.11313144
14. Sruthi C.K. , Aswathy Ravikumar, Harini Sriraman (2024) Enhancing Zero-Day Attack Detection with XAI-Driven ML Models and SMOTE Analysis . In Proceedings of the IEEE 3rd IEEE International Conference on Artificial Intelligence for Internet of Things. | DOI: 10.1109/AIIoT58432.2024.10574566
15. Reham Eltomy and Wassila Lalouani (2025) Explainable Intrusion Detection in Industrial Control Systems. In Proceedings of the IEEE 3rd IEEE 7th International Conference on Industrial Cyber-Physical Systems IEEE. DOI: 10.1109/ICPS59941.2024.10640024