Semi-Supervised Learning for Early Warning: Zero-Day Intrusion Attack Detection and Classification

Latifah AlQuairi[a], Mnaouer Kachout[b]*

[a]Master Cybersecurity, Department of Computer and Information Sciences, University of Hail, Hail 55476, Saudi Arabia
[b]University of Hail, College of Computer Science and Engineering, Department of Computer Engineering*
E-mail: m.kachout@uoh.edu.sa

Abstract:

This study highlights the importance of cybersecurity in protecting individuals and organizations from security threats. Zero-day (ZD) attacks exploit previously unknown vulnerabilities to steal sensitive data, compromise systems, and damage trust and reputation. Previous research has not addressed this problem and has faced limitations and challenges in data processing, classification, detection, and early detection of attacks in machine learning (ML). Thus, this paper aims to solve this problem by building an early warning system (EWS) that has been evaluated using a semi-supervised learning (SSL) approach to detect and classify ZD attacks. For system training and testing, the CICIDS2017 and CSE-CIC-IDS2018 datasets were selected. After that, the training data were preprocessed, filtered, and balanced to identify important features, and for the testing data that was not previously processed. In addition, machine learning algorithms such as support vector machine (SVM) and decision tree (DT). were applied to detect and classify attacks, and the effectiveness of the system was evaluated using performance criteria, with SVM achieving 95% classification accuracy, 96% detection accuracy, and DT achieving 99%.

Keywords: *Cybersecurity, Early warning system, Semi-supervised learning, Machine learning, Zero-day attack*

1. INTRODUCTION

The research is based on a serious attack in this area, and mentions that among the challenges that many face is the detection of zero-day (ZD) attacks, where vulnerabilities that have not been identified are exploited. These attacks can cause significant damage to data confidentiality and system availability [1]. This paper discusses ZD attacks, their associated cybersecurity risks for the undiscovered vulnerabilities. Hence, an SSL approach is proposed to enhance the speed of response to cyber incidents and establish early warning systems (EWS) with the aim of improving the detection and classification of ZD attacks [2]. The researchers stated that SSL enhances the detection of attacks by integrating labeled and unlabeled data, which faces and addresses challenges such as misrepresentation of data and the rise of false positives, so that it leads to more reliable classification models than labeled data [3]. As for classification, support vector machine (SVM) and decision tree (DT) classify attacks effectively, enhance accuracy in intrusion detection systems, and reduce false alarms, especially in complex networks such as internet of things (IoT), but DT is better than SVM [4],[5]. Consequently, ZD attacks are identified using appropriate SSL datasets [6]. As a result, the accuracy of classification with unlabeled data and the detection of hidden threats has been improved by modifying multi-view learning graphs, which further complicates threats such as malware and DDoS attacks, and requires

proactive detection strategies for SVM and DT models in SSL [7],[8]. This paper aims to bridge the gap in previous research and to address the challenges mentioned by using an early warning system to effectively detect and classify ZD attacks to provide timely alerts to enhance security processes and performance evaluation.

2. RELATED WORK

This section highlights the importance of zero-day attack detection in the field of cybersecurity, specifically ZD attacks. Several researchers have developed a new model for detecting ZD attacks in intrusion detection, while others have studied anomaly-based systems to analyze network traffic metrics [9],[10]. The ZDGAN model, developed by the researchers, aims to accurately detect and block malicious IoT traffic, working in tandem with the IADM model. [11],[12]. Others used machine learning and deep learning models for the NIDS system [13]. In [14], the researchers focused on the Automated Design IDS, where they applied machine learning using algorithms. Notably, some researchers have used electronic data collection to identify attacks, while others have implemented a switch-based IDS system [15],[16]. According to the researchers, they have improved traditional intrusion detection methods by developing a CNN1D model that enhances performance and automates detection techniques [17]-[19]. Therefore, all studies aim to detect malicious attacks and selections, highlighting the importance of IDS in IoT networks. One study used the MFE-ELM algorithm to enhance threat detection, while another used a metahoristic algorithm to identify ZD attacks [20]–[22]. The researchers used the AdaBoost algorithm to develop a new method for detecting leaks in IoT networks, using group learning and subspace aggregation methods [23],[24]. In [25], the researchers presented a malware detection model based on SSL. In contrast, others have used SSL to classify DDoS attacks and have applied Benford's Law to detect abnormal network behavior and identify ZD attacks [26][27]. At the same time, the researchers proposed a method for detecting anomalies in smart power grids, combining the auto-encoder with the OCSVM classification for effective attack detection [28],[29]. See Table 1.

Table 1. Detection of Zero-Day Attacks Summary

Year	Method	Contributions and Limitations
2020	IDS [9]	ZD was discovered using machine learning; however, it faces false alarm problems and lacks adequate testing in IoT architecture.
2021	IDS [10]	Network traffic metrics were analyzed using machine learning, generating predictions from diverse datasets and identifying six new cyberattacks, but it requires human intervention to check for classification errors among similar attacks.
2023	ZDGAN [11]	The system was classified using synthetic attack data as it improved the accuracy of verification and reduced loss, but the nature of the single-source data required model optimization to address generation quality inconsistencies.
2023	ZDNID [22]	The ZD attack detection model was developed using EIF, BAT, and Nevergrad algorithms to reduce false alarms and improve security, but it has process problems in real-world networks and problems with training duration.
2022	DZDIA [27]	To identify unusual network activity, especially ZD attacks, the researchers used Benford's Law. However, it limits its effectiveness via diverse network properties and is limited by positive real numbers and binary data.

A semi-supervised approach solves the problem of intrusion detection system failure by treating threats as anomalies, using unaddressed traffic to detect high-resolution invisible patterns. Table 2 describes three of the detection systems related to the study.

Table 2. Classification of Intrusion Detection Approaches

Detection	Primary Objective	Current Work Positioning
Anomaly [9], [10], [28], [29]	Defines legitimate traffic boundaries by spotting behavioral deviations.	Used as a baseline to filter statistical deviations.
Novelty [25], [26], [27]	Identifies emerging patterns not seen during training using a semi-supervised approach.	Represents the core of our Deep SSL framework's discovery logic.
Zero-Day [11], [12], [21], [22]	Detects signature-less exploits and unseen malicious patterns.	The ultimate objective of the proposed model.

3. METHODOLOGY

This work details the research process, which includes data collection, the use of SSL for attack classification, the implementation of an EWS, and performance evaluation. As shown in Figure 1, system logs, network traffic, and attack patterns are collected and processed using an SSL-based approach to detect intrusions. DT and SVM models classify attacks, creating an EWS in real or near-real time.

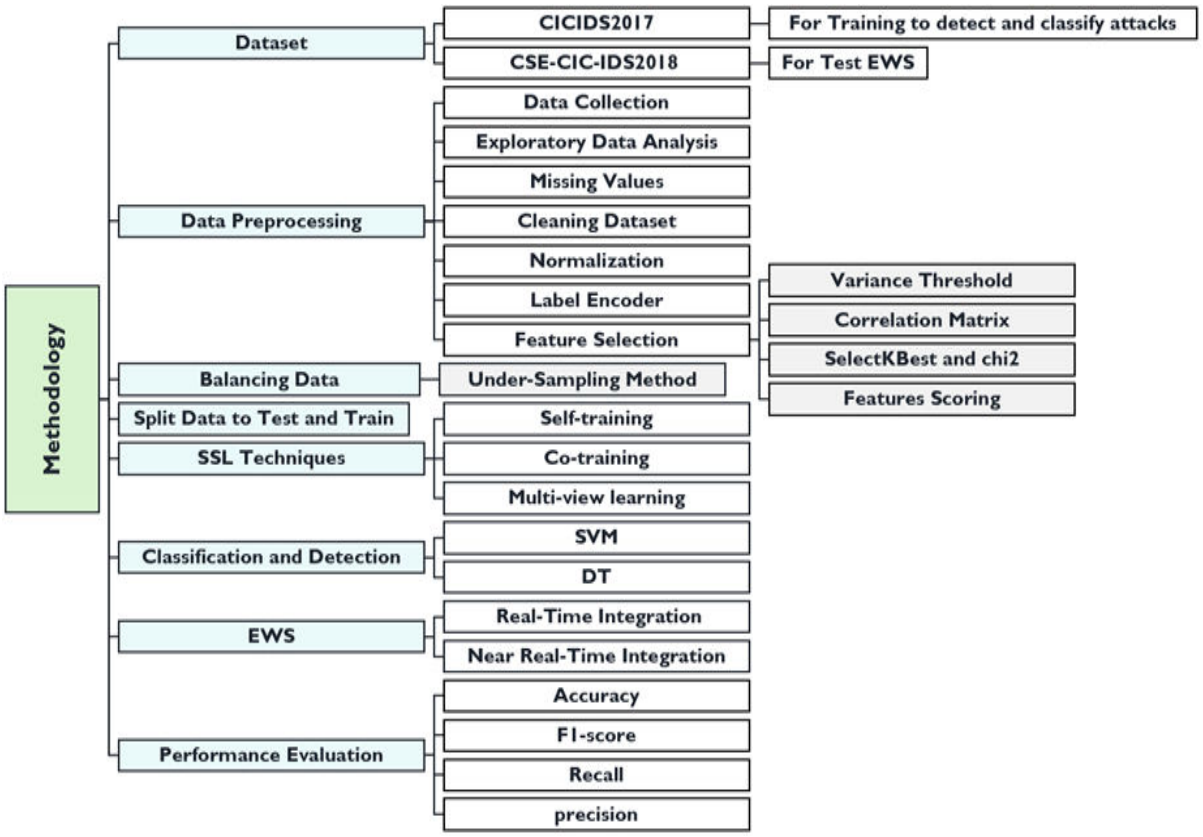

Figure 1. Framework of Research

3.1. Data Collection

The CICIDS2017 Intrusion Detection Evaluation Dataset, including 2,830,743 samples across 15 traffic types and 78 properties, organized into eight files with system logs, network traffic, and recognized attack patterns [30],[31]. Figure 2 shows the dataset of files, attacks, and samples.

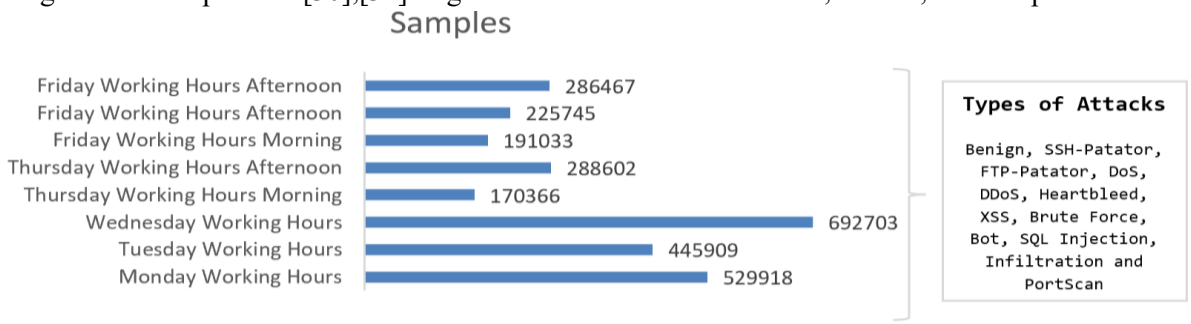

Figure 2. Data for Each File with Number of Data and Types of Attacks

3.1.1 Network Traffic Data

Malicious network traffic is difficult to detect because labeled data are limited; however, methods such as self-supervised edge embedding and graph-based deep learning can help address this challenge [32]. See Figure 3, where IP addresses are specified to prevent unauthorized data usage [32].

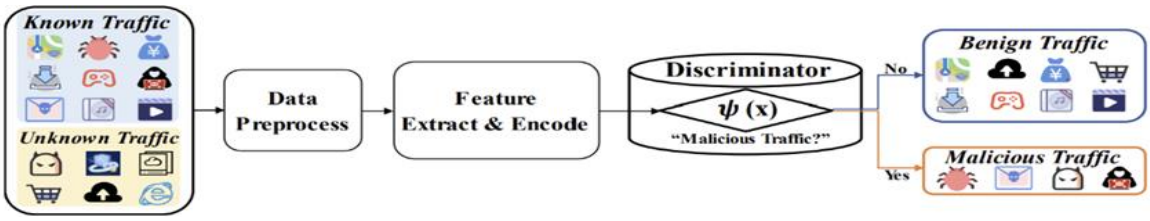

Figure 3. Network Traffic Data

3.1.2 System Logs

As mentioned in previous studies, system logs are classified into static, fixed, and variable sections, which are essential for monitoring performance and detecting anomalies, while their analysis helps identify gaps after the incident, relying on log events, templates, or keys [33]. Accordingly, SSL improves anomaly detection accuracy by leveraging large amounts of unprocessed data, reducing dataset requirements, and enhancing processing efficiency, especially in intrusion detection.

3.1.3 Known Attack Patterns

Attack patterns in network security refer to the analysis of the flow of attacks to detect malware attacks, which often hide their presence [34]. This characteristic supports early malware detection, especially when attackers use uncommon ports or protocols.

3.2. Data Preprocessing

At the beginning of the work, the dataset of eight files are combined to simplify processing, which includes exploratory analysis, dealing with missing values, performing cleanup, normalization, label encoding, and feature selection to highlight important attributes in Figure 4. In this paper, a dataset

representing ZD attacks based on previous studies was selected, as shown in Table 3, which lists the dataset used, technical details, and the type of research.

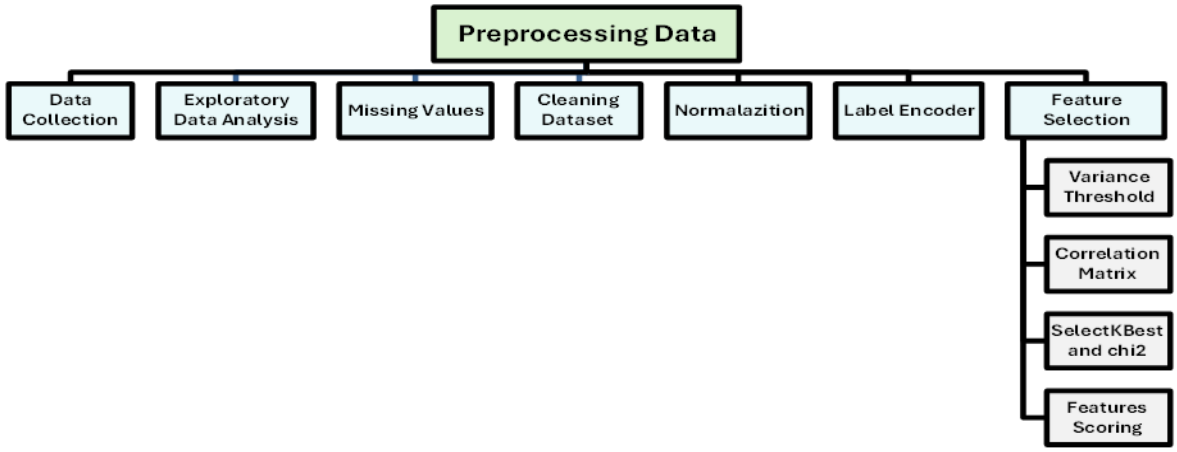

Figure 4. Preprocessing Data

Table 3. The summary outlines each study's dataset, technique, and learning type

Ref.	Dataset for ZD	Technique	Type
This study	CICIDS2017 for training CSE-CIC-IDS2018 for testing	Machine Learning	Semi-Supervised Learning
[35]	NSL-KDD and CIDD	Deep Learning	Transductive Transfer Learning
[36]	CIC-MalMem-2022	Machine Learning Deep Learning	Supervised Learning Semi-Supervised Learning

3.2.1 Exploratory Data Analysis (EDA)

This study used EDA to reveal hidden patterns and feature distributions in IoT traffic, helping ensure that the model architecture was aligned with the structure of ZD attack data [37],[38].

3.2.2 Missing Values

This stage is considered to be like gaps in the data, due to unavailable or unaggregated values, and has techniques to process them, such as mean, median, or more frequent values [39]. Columns with missing values are identified using .isnull().any(), and numerical columns are filled with the average, while categorical columns by the mode, thereby enhancing the integrity of the dataset and quality.

3.2.3 Data Cleaning

To improve the quality of the data, duplicates are removed in the data window using duplicated().sum(), which also processes the missing data. Therefore cleaned in a comprehensive, consistent, and coordinated way so that machine learning algorithms can understand it [39].

3.2.4 Normalization

To ensure metrics assign feature values between 0 and 1, subtract the minimum value and divide the score by the feature range [40]. Normalization via MinMaxScaler() was applied to prevent large-magnitude features from dominating the model's weight distribution [40]. For a balanced contribution of all attributes, this process stabilized the gradient in decline and improved the numerical structure of the dataset in classification and detection [40].

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (1)$$

$\boldsymbol{X'}$ is the normalized value after applying normalization.
$\boldsymbol{X}$ is the original value.
$\boldsymbol{X_{min}}$ is the minimum value.
$\boldsymbol{X_{max}}$ is the maximum value.
Steps of the Equation:
Subtract the minimum: $X - X_{min}$
Divide the result by the range of the feature: $(X_{max} - X_{min})$

3.2.5 Label Encoder

LabelEncoder() was used to convert categorical labels into numeric values for machine-learning-based processing [41].

3.2.6 Feature Selection

A multi-stage feature selection pipeline has been implemented to ensure high predictive accuracy and model robustness. Initially, the variance threshold technique enhances prediction accuracy by removing features with a variance below 0.1, while SelectKBest helps identify weak correlations during correlation analysis [42]. Therefore, a statistical methodology for selecting significant features in a dataset is the chi-square (x^2) technique which rates a feature's independence from the aim variable [43]. In addition, the feature significance score technique improves model performance by eliminating noncontributing data features, which simplifies the model, reduces dimensionality, and speeds up training [44]. Furthermore, a correlation matrix analysis was performed to evaluate linear and non-linear relationships, with parameters ranging from -1 to 1 [45]. In this study, data pre-processing indicates setting a variance threshold, analyzing the correlation matrix, and selecting 15-22 top features with SelectKBest and Chi2 techniques. From 79 features, 39 were significant, and 22 were specified for model training, improving accuracy, mitigating training time, and reducing overfitting by dimensionality reduction, validated through score features. The final selection was instrumental in enhancing the stability of the approach and ensuring superior performance via all rating SSL strategies.

3.3. Data Balancing

Random under-sampling was applied to the training group to address the class imbalance and to achieve 50/50 distribution of the class, and the data were split into 80% training and 20% testing. Although undersampling may risk information loss, it was strategically employed to mitigate majority-class bias and enhance the model's sensitivity to rare, low-frequency network threats [46].

3.4. Semi-Supervised Learning

In the training phase, labeled and unlabeled data are collected, as well as pattern recognition, data analysis, and security systems, where the accuracy of the models can be enhanced using large unlabeled data volumes [47].

3.4.1 Self-Training

Is an SSL technique used when limited labeled datasets are available [48]. It involves training a model on labeled data and then retraining iteratively by labeling unlabeled data with its predictions. Its efficacy relies on finding local optima in the original labeled data [48].

3.4.2 Co-Training

It is an SSL technique that trains two classifiers using different data perspectives to improve each other's training by labeling unlabeled samples. It is particularly useful when labeled data is scarce and is effective under the conditional independence of data perspectives [49].

3.4.3 Multi-View Learning

It enhances feature extraction in different learning approaches by addressing appearance variations and exceeding single-view methods in material classification [50]. It merges multiple data representations to improve understanding and decision-making accuracy [51],[52].

3.5. Attack Classification

By discovering the connections between input attributes, such as network packet data, and output classes, like benign traffic or different types of attacks (like Denial of Service or Remote Access Trojans), ML is commonly used for identifying and categorizing network attacks [50]. Large data volumes may be handled by ML-based algorithms to categorize the attacks [53].

3.5.1 Support Vector Machine

Statistical learning classifiers use multidimensional hyperplanes to differentiate between classes, allowing for quick training with linear kernels [54]. They perform well with linear data, excel in high-dimensional spaces, and are especially effective in binary classification scenarios [54]. Selected for their high-precision classification in high-dimensional spaces. SVMs are essential for detecting complex, non-linear patterns in sophisticated systems and cyberattacks.

3.5.2 Decision Tree

A decision tree is a model of data processing that converts complex patterns into rules, providing higher resolution than SVM and flexibility when combined with other methods, so that branches represent options, nodes represent properties, as well as result sheets by iteratively dividing data based on different properties [55],[56]. Select for their computational efficiency and interpretability. DTs enable rapid, rule-based decision-making in real-time.

3.6. Early Warning System

In the system, sensor data is analyzed to predict risks, send alerts, and respond instantly to hazards, enhancing prediction accuracy and reliability in situations where data is scarce in real-time [57]. A range

of tools is therefore used to provide timely and accurate warnings, enabling individuals, groups, and organizations to plan and respond effectively to potential harm [58].

3.6.1 Real-Time Integration

The immediate processing of sensor data to analyze and generate alarms in areas prone to disasters encompasses data collection, filtering, examination of early warning characteristics, model training, and alarm system integration [58].

3.6.2 Near Real-Time Integration

In EWS, attacks are detected, and delays are minimized using machine learning models, including radio frequency, SVM, and DT, so that it can recognize real-time attacks in network traffic to demonstrate the processing power for rapid response and mitigation [59].

3.7. Performance Evaluation

Performance is evaluated to determine the model's ability to predict and disseminate new data in order to assess the effectiveness, efficiency, and reliability of the model, and is usually done using these metrics [60]. The method used in the study is to scale its K-fold from traditional K-folding by reducing Overfitting and promoting generalization in machine learning, thus making it particularly effective for unbalanced datasets in detecting rare events [60].

3.7.1 Comparison with Existing Intrusion Detection Systems

The researchers state that network traffic monitoring is done by IDS, as it is a security tool to identify unusual activities and identify potential breaches to enhance data security [61]. See Figure 5.

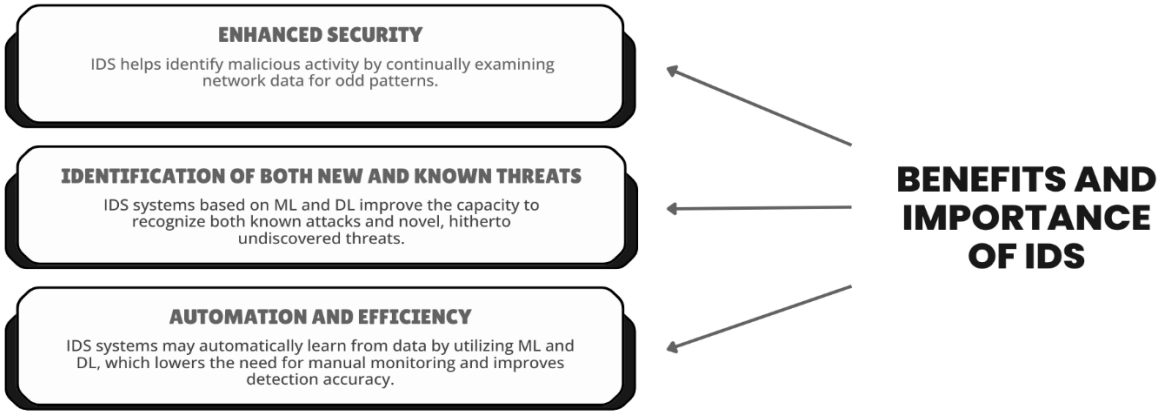

Figure 5. Benefits and Importance of IDS

Figure 6, the study illustrates how to collect and process network traffic data using ML/DL models to identify patterns, while these models were tested on test data to determine their effectiveness in detecting malicious or benign activity [61]. Figure 7, data classification techniques for models [61].

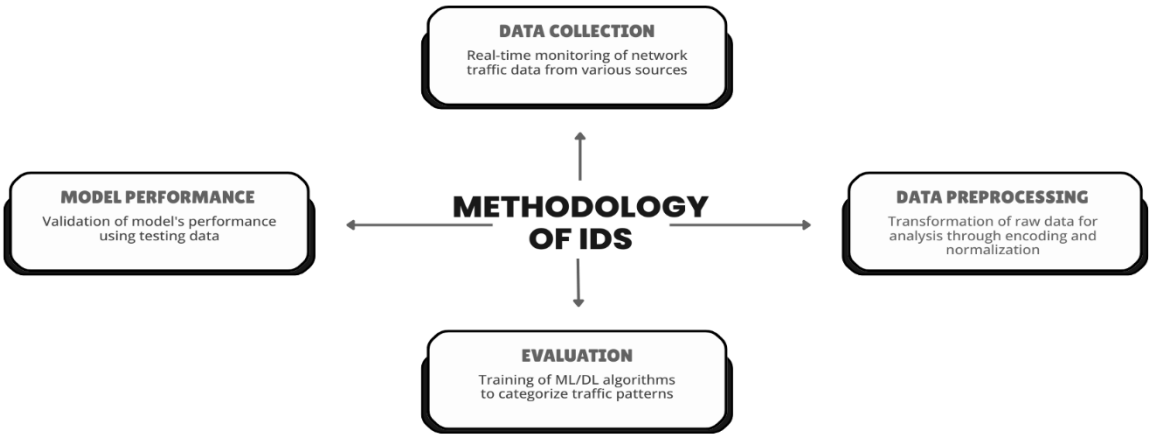

Figure 6. Methodology of IDS

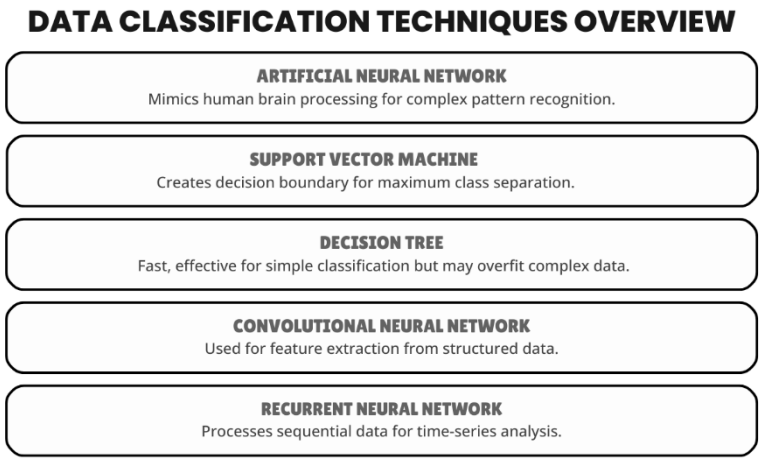

Figure 7. Data Classification Techniques

4. RESULTS AND DISCUSSION

In this experiment, Table 4 illustrates the methods used in terms of implementation, evaluation of performance metrics, discusses the results, and presents system alerts and categories.

Table 4. Experimental Setup

Requirements	Description
Operating System	Windows 11
Program	Google colab
programming language	Python, HTML
Dataset type	Benign and Attacks
Split data SSL	20% Labeled 80% Unlabeled
Name of the dataset	CICIDS2017 and CSE-CIC-IDS2018
Data Preprocessing	Collection, EDA, Missing, Cleaning, Normalization, Label Encoder, and Features
Classification and Detection	SVM & DT
Evaluation metrics	Recall, Precision, Accuracy, F1-score, ROC-AUC, PR-AUC, FAR, FNR, and MCC

4.1. Performance Results

The pre-processing phase was performed, resulting in an improved collection of 2,462,817 high-quality samples and 22 features. A 10-fold stratified cross-validation protocol was also adopted. The class nature of this approach suggests crucial in cybersecurity contexts, as it maintains the distribution of the original categories and prevents the model from developing a bias towards the majority groups. This study evaluates the performance of the early warning system using key measures before and after implementation. Figure 8 shows the results of cross-verification.

	accuracy	precision	recall	f1_score
0	0.8831	0.896440	0.8831	0.882069
1	0.8886	0.902783	0.8886	0.887570
2	0.8828	0.896828	0.8828	0.881712
3	0.8819	0.897194	0.8819	0.880707
4	0.8778	0.893217	0.8778	0.876543
5	0.8824	0.897231	0.8824	0.881248
6	0.8867	0.899152	0.8867	0.885771
7	0.8838	0.896070	0.8838	0.882853
8	0.8846	0.897899	0.8846	0.883587
9	0.8840	0.897459	0.8840	0.882968

	accuracy	precision	recall	f1_score
0	0.9987	0.998700	0.9987	0.9987
1	0.9987	0.998700	0.9987	0.9987
2	0.9985	0.998501	0.9985	0.9985
3	0.9984	0.998401	0.9984	0.9984
4	0.9994	0.999400	0.9994	0.9994
5	0.9984	0.998400	0.9984	0.9984
6	0.9988	0.998801	0.9988	0.9988
7	0.9989	0.998900	0.9989	0.9989
8	0.9982	0.998201	0.9982	0.9982
9	0.9989	0.998900	0.9989	0.9989

Figure 8. StratifiedKFold in Detection

4.1.1 Attack Detection

Figure 9 in the attack detection process shows that self-training improves predictions on unlabeled data using high-confidence results from labeled datasets, where it classifies the data into attack as 0 and non-attack as 1. On the other hand, it illustrates the same format for co-training, so that the two models were trained on different features, allowing for the exchange of high-confidence predictions. In addition to multi-view learning that enhances performance using two models trained on diverse data characteristics, each with its own expectations, as shown in Figure 10.

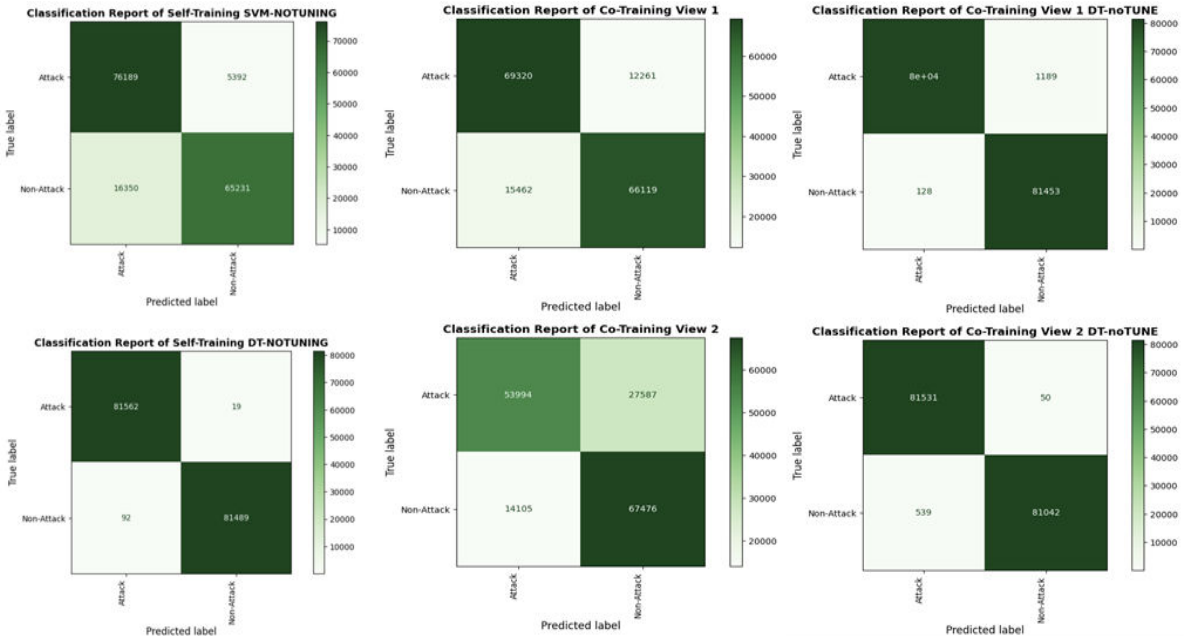

Figure 9. Self-Training and Co-Training in Detection for SVM and DT

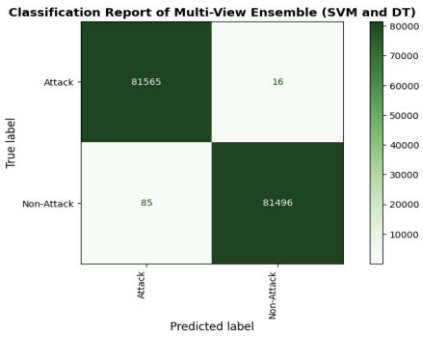

Figure 10. Multi-View Learning in Detection for SVM and DT

4.1.2 Attack Classification

Figure 11, the performance of the SVM and DT algorithms was evaluated using StratifiedKFold. Figure 12 shows the performance of the models for classifying attacks, where the SVM achieved remarkable accuracy. Comparison with the DT model, which has proven effective in co-training. Figure 13 shows that multi-view learning enhanced classification accuracy by integrating features from multiple scenes to train SVM and DT models.

	accuracy	precision	recall	f1_score
0	0.8502	0.892538	0.8502	0.858539
1	0.8422	0.900437	0.8422	0.857804
2	0.8498	0.897716	0.8498	0.861268
3	0.8427	0.898161	0.8427	0.856738
4	0.8443	0.896301	0.8443	0.856701
5	0.8521	0.899393	0.8521	0.862968
6	0.8298	0.901043	0.8298	0.851982
7	0.8513	0.896714	0.8513	0.861103
8	0.8484	0.902350	0.8484	0.861772
9	0.8283	0.901387	0.8283	0.850336

	accuracy	precision	recall	f1_score
0	0.9989	0.998901	0.9989	0.998900
1	0.9982	0.998203	0.9982	0.998200
2	0.9985	0.998511	0.9985	0.998503
3	0.9976	0.997604	0.9976	0.997600
4	0.9983	0.998304	0.9983	0.998300
5	0.9982	0.998201	0.9982	0.998200
6	0.9989	0.998900	0.9989	0.998899
7	0.9975	0.997504	0.9975	0.997501
8	0.9980	0.998001	0.9980	0.997999
9	0.9981	0.998100	0.9981	0.998099

Figure 11. StratifiedKFold in Classification

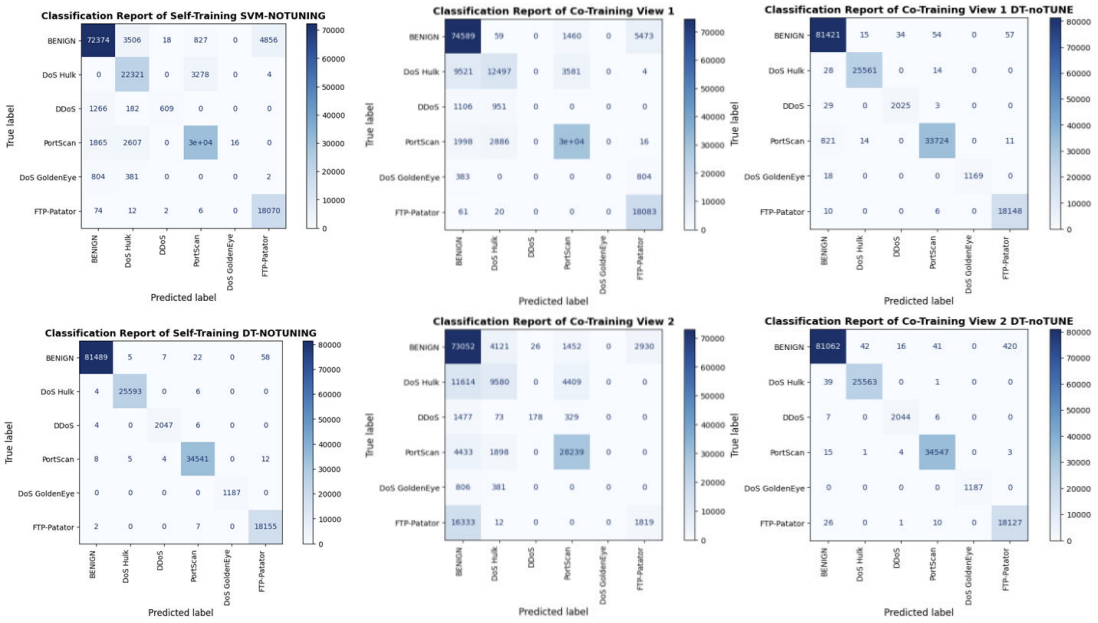

Figure 12. Self-Training and Co-Training in Classification for SVM and DT

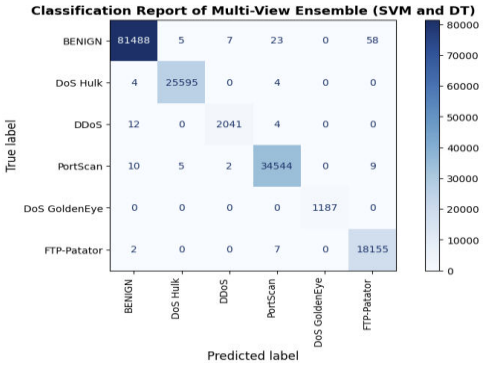

Figure 13. Multi-View Learning in Classification for SVM and DT

4.2. Discussion on Findings

Tables 5 and 6 indicate that DT results significantly outperform SVM, while maintaining an accuracy rate of 99% from self-training, co-training, and multi-view learning structures. 10-fold stratified cross-validation was used to ensure these results were not due to proper data segmentation. This confirmed that the high accuracy and Matthews correlation coefficient (MCC) scores are stable indicators of the model's durability. This consistency highlights the regime's ability to generalize to unseen attack patterns. This capability is essential for early identification of ZD threats in complex and rapidly evolving network environments.

Table 5. Evaluation of Classification Models

Method	ACC	Precision	Recall	F1-Score	ROC-AUC	PR-AUC	FAR	FNR	MCC
DT	99.93	99.93	99.93	99.93	99.98	99.97	0.07	0.07	99.86
DT V1	99.19	99.20	99.19	99.19	99.85	99.72	0.81	0.81	98.38
DT V2	99.64	99.64	99.64	99.64	99.92	99.88	0.36	0.36	99.28
SVM	86.67	87.35	86.67	86.61	91.50	89.20	13.33	13.33	73.96
SVM V1	83.01	83.06	83.01	83.00	88.20	85.10	16.99	16.99	66.07
SVM V2	74.45	75.13	74.45	74.27	79.40	76.50	25.55	25.55	49.40
DT+SVM	99.94	99.94	99.94	99.94	99.99	99.98	0.06	0.06	99.88

Table 6. Evaluation of Detection Models

Method	ACC	Precision	Recall	F1-Score	ROC-AUC	PR-AUC	FAR	FNR	MCC
DT	99.91	99.91	99.91	99.91	99.97	99.96	0.09	0.09	99.82
DT V1	99.32	99.32	99.32	99.32	99.88	99.80	0.68	0.68	98.64
DT V2	99.61	99.61	99.62	99.61	99.91	99.85	0.39	0.38	99.22
SVM	87.92	88.88	87.92	87.54	92.10	90.40	12.08	12.08	75.84
SVM V1	82.64	80.86	82.64	81.02	86.80	82.10	17.36	17.36	63.50
SVM V2	69.18	66.01	69.18	65.12	74.20	68.90	30.82	30.82	35.19
DT+SVM	99.91	99.91	99.91	99.91	99.98	99.97	0.09	0.09	99.82

Supervised learning is used to accurately classify anomalies and cyberattacks in Internet traffic, while accurately evaluating detection performance using evaluation metrics to ensure system robustness and security [63]. The same methodology from the dataset, number of features, and algorithms was used to demonstrate that the semi-supervised analysis technique outperformed supervised. See Tables 7 and 8 to present the results.

Table 7. Evaluation of Detection and Classification for Supervised Learning

Method	ACC	Precision	Recall	F1-Score	ROC-AUC	PR-AUC	FAR	FNR	MCC
DT	99.62	99.68	99.56	99.62	99.86	99.78	0.32	0.44	99.23
SVM	92.13	89.17	95.91	92.42	96.88	96.59	11.66	04.09	84.51

Table 8. Evaluation of Detection and Classification for Supervised Learning

Method	ACC	Precision	Recall	F1-Score	ROC-AUC	PR-AUC	FAR	FNR	MCC
DT	97.31	97.38	97.31	97.31	99.22	97.84	6.48	1.26	96.65
SVM	93.32	93.68	93.32	93.24	99.22	98.20	18.26	1.86	91.78

Receiver operator characteristic (ROC) curves reveal the extraordinary discriminating power of SSL, which appears to have a near-perfect Area under the curve (AUC) of 99.99% in detection and 99.98% in classification. Superior statistical accuracy suggests that convergence proves to isolate threats from core activities. Supplementing these results, Precision-recall (PR) curves assert the robustness of SSL under unbalanced distributions. See Figure 14.

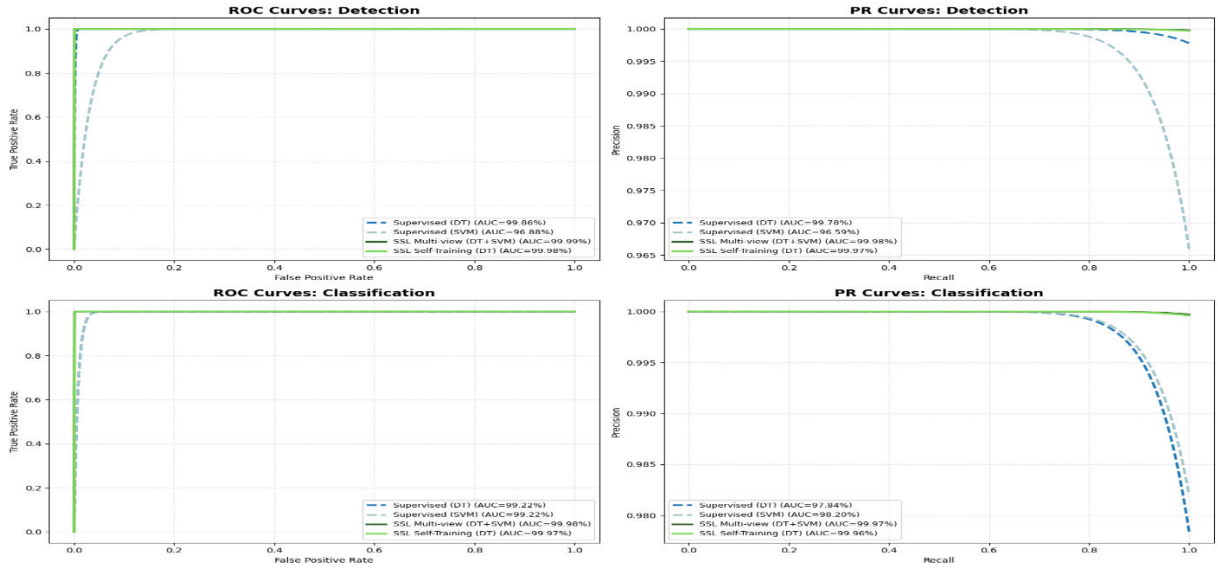

Figure 14. ROC and PR for Semi-Supervised vs. Supervised Learning

Table 9 shows more details of SSL in this model and the previous study, with DT+SVM outperforming XGBoost with 99.94% accuracy with 100,000 samples. In addition to the detection time for inference within 0.26 seconds.

Table 9. Comparative Performance of Proposed Models vs. Literature

Method	ACC	Precision	Recall	F1-Score	Dataset	Samples	Train/ Test	Real-Time
DT	99.93	99.93	99.93	99.93	CICIDS2017	100000	80/20	0.25 sec
SVM	87.92	88.88	87.92	87.54	CICIDS2017	50000	80/20	0.20 sec
DT+SVM	99.94	99.94	99.94	99.94	CICIDS2017	100000	80/20	0.26 sec
XGBoost [25]	98.96	98.95	98.96	98.95	VirusShare	10974	80/20	N/A

4.3. System Alerts and Categories

In the testing phase of the system for detection and classification, the CSE-CIC-IDS2018 dataset was used, and 500 samples were selected to ensure that the models worked as required [64].

4.3.1 Attack Detection and Classification in Early Warning System

Self-training improved the performance of SVM models, in Figure 15, but did not significantly affect the performance of DT. Shows Figure 16 external approach, thus effectively identified attack and non-attack in view 1, while the SVM model showed improved performance in view 2.

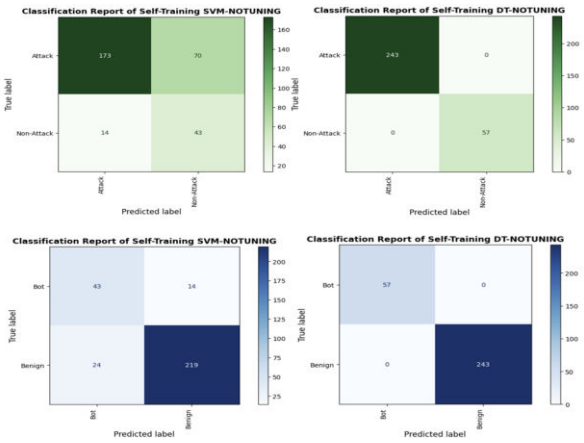

Figure 15. Self-Training in Detection and Classification for SVM and DT

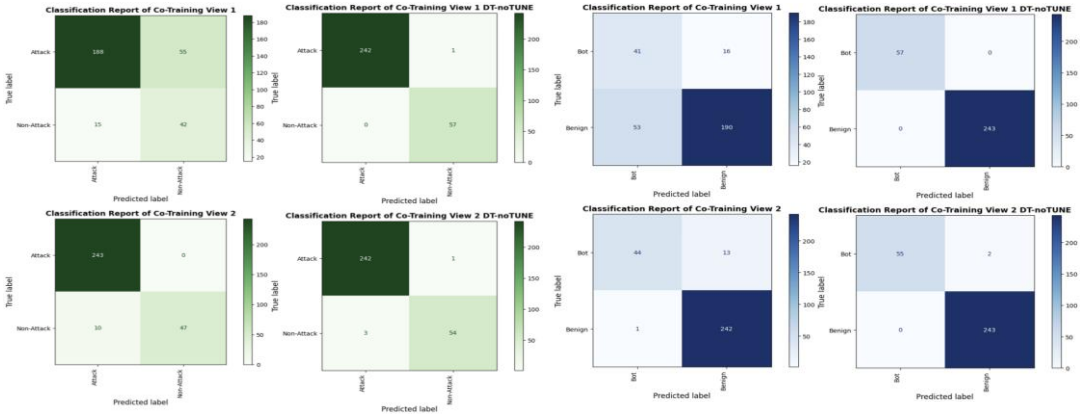

Figure 16. Co-Training in Detection and Classification for SVM and DT

Figure 17, the model detects and classifies attacks with high accuracy and enhances the performance of the system, providing excellent attack detection.

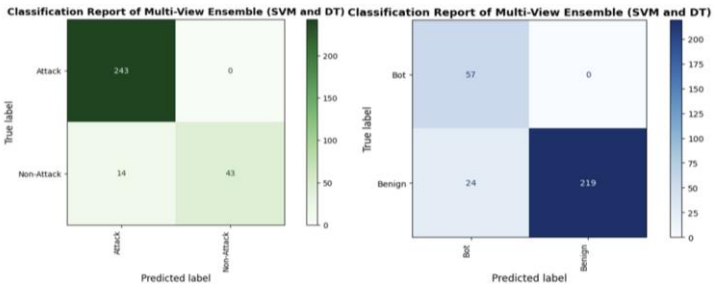

Figure 17. Multi-View Learning in Detection and Classification for SVM and DT

4.4. Discussion on Findings

4.4.1 Evaluation for Classification and Detection in Early Warning System

This paper compared three SSL techniques and found that multi-view learning and self-training were better than co-training in EWS, while the SVM models in Table 10 had better evaluation metrics in system testing as well as DT still excelled in the training and testing phases. DT showed better performance, achieving 99.19% accuracy. The co-Training method was recommended for maximum accuracy and performance in attack detection. The performance of this system in detection is 95.12% accuracy in Table 10; it is 92.48% in classification, as shown in Table 11, to enhance multi-view learning (SVM + DT) by balancing detection and classification accuracy.

Table 10. Evaluation of Detection Models in EWS

Method	ACC	Precision	Recall	F1-Score	ROC-AUC	PR-AUC	FAR	FNR	MCC
SVM	72.40	82.15	72.40	74.22	72.50	82.10	23.85	27.60	46.55
DT	100.00	100.00	100.00	100.00	100.00	100.00	0.00	0.00	100.00
SVM V1	76.32	83.40	76.32	78.10	76.45	83.25	21.75	23.68	52.40
SVM V2	96.28	96.35	96.28	96.31	96.50	96.42	3.82	3.72	92.65
DT V1	99.19	99.20	99.19	99.19	99.85	99.80	0.85	0.81	98.38
DT V2	98.07	98.15	98.07	98.11	98.40	98.25	1.95	1.93	96.20
DT+SVM	95.12	95.25	95.12	95.18	95.40	95.30	4.92	4.88	90.35

Table 11. Evaluation of Classification Models in EWS

Method	ACC	Precision	Recall	F1-Score	ROC-AUC	PR-AUC	FAR	FNR	MCC
SVM	87.12	88.05	87.12	87.08	87.45	88.20	12.88	12.88	74.25
DT	100.00	100.00	100.00	100.00	100.00	100.00	0.00	0.00	100.00
SVM V1	77.45	83.12	77.45	77.40	77.60	83.35	22.55	22.55	54.18
SVM V2	95.32	95.40	95.32	95.31	95.50	95.45	4.68	4.68	90.62

DT V1	100.00	100.00	100.00	100.00	100.00	100.00	0.00	0.50	99.45
DT V2	99.15	99.20	99.15	99.15	99.30	99.25	0.85	01.05	98.32
DT+SVM	92.48	94.15	92.48	92.45	92.60	94.30	7.52	7.52	84.85

4.5. Early Warning System Application

A Python-based application was developed to experimentally detect attacks using an HTML web interface and self-training approaches. When the system is opened, security is shown in green; when an attack is discovered, security is shown in red. By issuing alerts or putting extra security measures in place, administrators may act right away. The program interacts with larger systems, such as monitoring and data analysis, by providing an API for applications that you may use to obtain a security status by identifying system threats, or issuing an alert to put in place additional protections, so that this function enables security verification to make quick decisions in production settings. See Figures 18 and 19. In Figure 20, the diagram illustrates EWS from operations in terms of operation to alert status, including attack detection, reporting, safety assurance, and streamlining operations by identifying different situations and actions.

Figure 18. Safe System

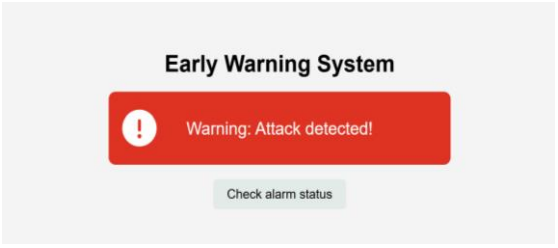

Figure 19. Warning System

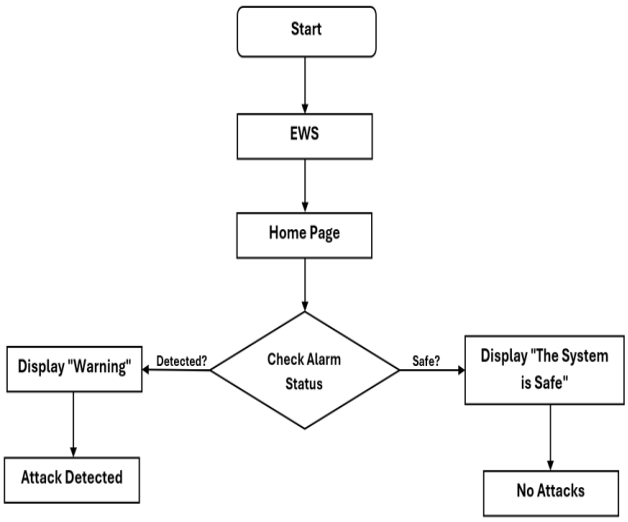

Figure 20. Early Warning System Work

5. CONCLUSION AND DIRECTIONS FOR FUTURE RESEARCH

This study presents an SSL-based early warning system for detecting and classifying ZD attacks by leveraging both labeled and unlabeled data, in addition to the already proposed framework that overcame the limitations of traditional intrusion detection methods. Experimental results on the work show that the decision tree classifier outperforms the support vector machine in several scenarios, achieving higher accuracy and stability within EWS.

The results emerged after the implementation of the near-instantaneous EWS for the timely detection of malicious activities as well as support for rapid response actions, in order to minimize the potential impact of attacks. Thus, SSL results provide an effective and practical solution for proactive cybersecurity defense. Future work will focus on the goal of integrating deep learning models,

expanding attack coverage, automating response mechanisms, and validating systems in real-world network environments.

Acknowledgment

The authors extend their sincere thanks and gratitude to the University of Hail in Saudi Arabia for their generous support and effective contribution to the completion of this work.

References

[1] R. M. Zaki and I. S. Naser, "Hybrid classifier for detecting zero-day attacks on IoT networks," *Mesopotamian Journal of CyberSecurity*, vol. 4, no. 3, pp. 59–74, Nov. 2024, doi: 10.58496/MJCS/2024/016.

[2] D. Agnew, A. Rice-Bladykas, and J. McNair, "Detection of zero-day attacks in a software-defined LEO constellation network using enhanced network metric predictions," *IEEE Open Journal of Communications Society*, vol. 5, pp. 6611–6626, Oct. 2024, doi: 10.1109/OJCOMS.2024.3481965.

[3] M. Nozad Bonab, J. Tanha, and M. Masdari, "A semi-supervised learning approach to quality-based web service classification," *IEEE Access*, vol. 12, pp. 50489–50502, Apr. 2024, doi: 10.1109/ACCESS.2024.3385341.

[4] B. I. Hairab, M. S. Elsayed, A. D. Jurcut, and M. A. Azer, "Anomaly detection based on CNN and regularization techniques against zero-day attacks in IoT networks," *IEEE Access*, vol. 10, pp. 98427–98440, Sep. 2022, doi: 10.1109/ACCESS.2022.3206367.

[5] L. A. C. Ahakonye, C. I. Nwakanma, J.-M. Lee, and D.-S. Kim, "Efficient classification of enciphered SCADA network traffic in smart factory using decision tree algorithm," *IEEE Access*, vol. 9, pp. 154891–154904, Nov. 2021, doi: 10.1109/ACCESS.2021.3127560.

[6] C. A. S. Barreto, A. C. Gorgônio, J. C. Xavier-Junior, and A. M. P. Canuto, "Applying efficient selection techniques of unlabeled instances for wrapper-based semi-supervised methods," *IEEE Access*, vol. 10, pp. 43535–43555, 2022, doi: 10.1109/ACCESS.2022.3169498.

[7] J. Ma and C. Yuan, "Adaptive safe semi-supervised extreme machine learning," *IEEE Access*, vol. 7, pp. 76176–76188, 2019, doi: 10.1109/ACCESS.2019.2922385.

[8] O. I. Falowo, M. Ozer, C. Li, and J. Bou Abdo, "Evolving malware and DDoS attacks: Decadal longitudinal study," *IEEE Access*, vol. 12, pp. 39221–39232, 2024, doi: 10.1109/ACCESS.2024.3376682.

[9] H. Hindy et al., "Utilising deep learning techniques for effective zero-day attack detection," *Electronics*, vol. 9, no. 10, p. 1684, Oct. 2020, doi: 10.3390/electronics9101684.

[10] B. M. Serinelli, A. Collen, and N. A. Nijdam, "On the analysis of open source datasets: Validating IDS implementation for well-known and zero day attack detection," *Procedia Computer Science*, vol. 191, pp. 192–199, Jul. 2021, doi: 10.1016/j.procs.2021.07.024.

[11] N. Peppes, T. Alexakis, E. Adamopoulou, and K. Demestichas, "The effectiveness of zero-day attacks data samples generated via GANs on deep learning classifiers," *Sensors*, vol. 23, no. 2, p. 900, Jan. 2023, doi: 10.3390/s23020900.

[12] N. S. Alotaibi, H. I. Ahmed, and S. O. M. Kamel, "Dynamic adaptation attack detection model for a distributed multi-access edge computing smart city," *Sensors*, vol. 23, no. 16, p. 7135, 2023, doi: 10.3390/s23167135.

[13] K. A. Dhanya et al., "Detection of network attacks using machine learning and deep learning models," *Procedia Computer Science*, vol. 218, pp. 57–66, 2023, doi: 10.1016/j.procs.2022.12.401.

[14] K. S. Kiran et al., "Building an intrusion detection system for IoT environment using machine learning techniques," *Procedia Computer Science*, vol. 171, pp. 2372–2379, 2020, doi: 10.1016/j.procs.2020.04.256.

[15] V. Kanimozhi and T. P. Jacob, "Artificial intelligence outflanks all other machine learning classifiers in network intrusion detection system on the realistic cyber dataset CSE-CIC-IDS2018 using cloud computing," *ICT Express*, vol. 7, no. 3, pp. 366–370, 2021, doi: 10.1016/j.icte.2020.12.004.

[16] Z. Wu, H. Zhang, P. Wang, and Z. Sun, "RTIDS: A robust transformer-based approach for intrusion detection system," *IEEE Access*, vol. 10, pp. 64375–64387, 2022, doi: 10.1109/ACCESS.2022.3182333.

[17] A. T. Assy, Y. Mostafa, A. Abd El-khaleq, and M. Mashaly, "Anomaly-based intrusion detection system using one-dimensional convolutional neural network," *Procedia Computer Science*, vol. 220, pp. 78–85, 2023, doi: 10.1016/j.procs.2023.03.013.

[18] A. Paya, S. Arroni, V. García-Díaz, and A. Gómez, "Apollon: A robust defense system against adversarial machine learning attacks in intrusion detection systems," *Computers & Security*, vol. 136, p. 103546, 2024, doi: 10.1016/j.cose.2023.103546.

[19] F. Nabi and X. Zhou, "Enhancing intrusion detection systems through dimensionality reduction: A comparative study of machine learning techniques for cyber security," *Cyber Security and Applications*, vol. 2, p. 100033, 2024, doi: 10.1016/j.csa.2023.100033.

[20] M. Sarhan et al., "Feature extraction for machine learning-based intrusion detection in IoT networks," *Digital Communications and Networks*, 2022, doi: 10.1016/j.dcan.2022.01.001.

[21] H. Lin, Q. Xue, J. Feng, and D. Bai, "Internet of things intrusion detection model and algorithm based on cloud computing and multi-feature extraction extreme learning machine," *Digital Communications and Networks*, vol. 9, no. 1, pp. 111–124, 2023, doi: 10.1016/j.dcan.2022.04.018.

[22] N. Alam and M. Ahmed, "Zero-day network intrusion detection using machine learning approach," *IJRITCC*, vol. 11, no. 8s, pp. 194–201, Aug. 2023, doi: 10.17762/ijritcc.v11i8s.7183.

[23] Z. Sun, G. An, Y. Yang, and Y. Liu, "Optimized machine learning enabled intrusion detection system for internet of medical things," *Franklin Open*, vol. 6, p. 100056, 2024, doi: 10.1016/j.fraope.2023.100056.

[24] J. Zhu and X. Liu, "An integrated intrusion detection framework based on subspace clustering and ensemble learning," *Computers and Electrical Engineering*, vol. 115, p. 109113, 2024, doi: 10.1016/j.compeleceng.2024.109113.

[25] X. Gao et al., "Malware classification for the cloud via semi-supervised transfer learning," *Journal of Information Security and Applications*, vol. 55, p. 102661, 2020, doi: 10.1016/j.jisa.2020.102661.

[26] M. Aamir and S. M. A. Zaidi, "Clustering based semi-supervised machine learning for DDoS attack classification," *Journal of King Saud University - Computer and Information Sciences*, vol. 33, no. 4, pp. 436–446, 2021, doi: 10.1016/j.jksuci.2019.02.009.

[27] I. Mbona and J. H. P. Eloff, "Detecting zero-day intrusion attacks using semi-supervised machine learning approaches," *IEEE Access*, vol. 10, pp. 69822–69838, 2022, doi: 10.1109/ACCESS.2022.3187116.

[28] E. Shchetinin and T. Velieva, "Detection of cyber-attacks on the power smart grids using semi-supervised deep learning models," *Discrete and Continuous Models and Applied Computational Science*, vol. 30, no. 3, pp. 258–268, 2022, doi: 10.22363/2658-4670-2022-30-3-258-268.

[29] C. Kim et al., "Automated, reliable zero-day malware detection based on autoencoding architecture," *IEEE Transactions on Network and Service Management*, vol. 20, no. 3, pp. 3900–3914, Sept. 2023, doi: 10.1109/TNSM.2023.3251282.

[30] J. Zhang et al., "A network intrusion detection model based on BiLSTM with multi-head attention mechanism," *Electronics*, vol. 12, no. 19, p. 4170, Oct. 2023, doi: 10.3390/electronics12194170.

[31] I. Sharafaldin et al., "CIC-IDS2017 [Data set]," Kaggle, 2022. [Online]. Available: https://doi.org/10.34740/KAGGLE/DSV/4059877

[32] M. Liu et al., "Semi-supervised encrypted malicious traffic detection based on multimodal traffic characteristics," *Sensors*, vol. 24, no. 20, p. 6507, Oct. 2024, doi: 10.3390/s24206507.
[33] S. Ranga and N. G. Mohankumar, "Investigating optimal features in log files for anomaly detection using optimization approach," *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 13, no. 1, pp. 287–295, Mar. 2024, doi: 10.11591/ijai.v13.i1.pp287-295.
[34] P. Fernandes et al., "Unveiling malicious network flows using Benford's law," *Mathematics*, vol. 12, no. 15, p. 2299, Jul. 2024, doi: 10.3390/math12152299.
[35] N. Sameera and M. Shashi, "Deep transductive transfer learning framework for zero-day attack detection," *ICT Express*, vol. 6, no. 4, pp. 361–367, 2020, doi: 10.1016/j.icte.2020.03.003.
[36] Z. Dai et al., "An intrusion detection model to detect zero-day attacks in unseen data using machine learning," *PLoS ONE*, vol. 19, no. 9, p. e0308469, 2024, doi: 10.1371/journal.pone.0308469.
[37] M. A. Furqon et al., "Critical exploratory data analysis on stroke prediction dataset," *Jurnal Komputer Terapan*, vol. 10, no. 1, pp. 67–77, Jun. 2024, doi: 10.35143/jkt.v10i1.6307.
[38] K. Eckelt et al., "Loops: Leveraging provenance and visualization to support exploratory data analysis in notebooks," *IEEE Transactions on Visualization and Computer Graphics*, 2024, doi: 10.1109/TVCG.2024.3456186.
[39] M. R. Buiya et al., "Detecting IoT cyberattacks: Advanced machine learning models for enhanced security in network traffic," *Journal of Computer Science and Technology Studies*, vol. 6, no. 4, pp. 142–152, Oct. 2024, doi: 10.32996/jcsts.2024.6.4.16.
[40] M. H. Alsulami, "Residual dense optimization-based multi-attention transformer to detect network intrusion against cyber attacks," *Applied Sciences*, vol. 14, no. 17, p. 7763, Sept. 2024, doi: 10.3390/app14177763.
[41] M. Gamal et al., "Improving intrusion detection using LSTM-RNN to protect drones' networks," *Egyptian Informatics Journal*, vol. 27, p. 100501, 2024, doi: 10.1016/j.eij.2024.100501.
[42] A. Chatterjee, "Enhancing digital sovereignty in automated stroke prediction and recommendation system with explanations and semantic reasoning," Sept. 2024, doi: 10.13140/RG.2.2.18712.40968.
[43] M. A. H. Alias et al., "Student performance classification: A comparison of feature selection methods based on online learning activities," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 14, no. 4, pp. 4675–4685, Aug. 2024, doi: 10.11591/ijece.v14i4.pp4675-4685.
[44] A. Pathak et al., "Machine learning approach to detect android malware using feature-selection based on feature importance score," *Journal of Engineering Research*, Apr. 2024, doi: 10.1016/j.jer.2024.04.008.
[45] R. Prasad and A. K. Saxena, "A machine learning approach for an early prediction of Parkinson's disease," *Indian Journal of Science and Technology*, vol. 17, no. 33, pp. 3410–3418, 2024, doi: 10.17485/IJST/v17i33.2091.
[46] Y.-S. Jeon and D.-J. Lim, "PSU: Particle stacking undersampling method for highly imbalanced big data," *IEEE Access*, vol. 8, pp. 131920–131926, Jul. 2020, doi: 10.1109/ACCESS.2020.3009753.
[47] J. R. Stomps et al., "SNM radiation signature classification using different semi-supervised machine learning models," *Journal of Nuclear Engineering*, vol. 4, no. 3, pp. 448–466, July 2023, doi: 10.3390/jne4030032.
[48] S. Baek et al., "Abnormal vibration detection in the bearing-shaft system via semi-supervised classification of accelerometer signal patterns," *Procedia Manufacturing*, vol. 51, pp. 316–323, Jan. 2020, doi: 10.1016/j.promfg.2020.10.045.
[49] Y. Zhao et al., "Co-training semi-supervised learning for fine-grained air quality analysis," *Atmosphere*, vol. 14, no. 1, p. 143, Jan. 2023, doi: 10.3390/atmos14010143.
[50] J. Bi and F. Dornaika, "Sample-weighted fused graph-based semi-supervised learning on multi-view data," *Information Fusion*, vol. 104, p. 102175, 2024, doi: 10.1016/j.inffus.2023.102175.

[51] B. U. Sumon et al., "Multi-view learning for material classification," *Journal of Imaging*, vol. 8, no. 7, p. 186, 2022, doi: 10.3390/jimaging8070186.

[52] Y. Liu et al., "Dynamic evidence decoupling for trusted multi-view learning," *arXiv preprint arXiv:2410.03796*, 2024.

[53] N. Rust-Nguyen et al., "Darknet traffic classification and adversarial attacks using machine learning," *Computers & Security*, vol. 127, p. 103098, 2023, doi: 10.1016/j.cose.2023.103098.

[54] K. A. Dhanya et al., "Detection of network attacks using machine learning and deep learning models," *Procedia Computer Science*, vol. 218, pp. 57–66, 2023, doi: 10.1016/j.procs.2022.12.401.

[55] A. Sinha et al., "Exploring sentiments in the Russia-Ukraine conflict: A comparative analysis of KNN, decision tree and logistic regression machine learning classifiers," *Procedia Computer Science*, vol. 235, pp. 1068–1076, 2024, doi: 10.1016/j.procs.2024.04.101.

[56] R. Bemthuis et al., "Business rule extraction using decision tree machine learning techniques: A case study into smart returnable transport items," *Procedia Computer Science*, vol. 220, pp. 446–455, Apr. 2023, doi: 10.1016/j.procs.2023.03.057.

[57] H. Zhou et al., "Dynamic real-time infrastructure planning and deployment for disaster early warning systems," in *Computational Science – ICCS 2018*, pp. 644–654, June 2018, doi: 10.1007/978-3-319-93701-4_51.

[58] Y. Wu et al., "A fast deploying monitoring and real-time early warning system for the baige landslide in Tibet, China," *Sensors*, vol. 20, no. 22, p. 6619, Nov. 2020, doi: 10.3390/s20226619.

[59] V. Morfino and S. Rampone, "Towards near-real-time intrusion detection for IoT devices using supervised learning and Apache Spark," *Electronics*, vol. 9, no. 3, p. 444, Mar. 2020, doi: 10.3390/electronics9030444.

[60] J. Kaliappan et al., "Impact of cross-validation on machine learning models for early detection of intrauterine fetal demise," *Diagnostics*, vol. 13, no. 10, p. 1692, May 2023, doi: 10.3390/diagnostics13101692.

[61] Z. Ahmad et al., "Network intrusion detection system: A systematic study of machine learning and deep learning approaches," *Transactions on Emerging Telecommunications Technologies*, vol. 32, no. 3, p. e4150, Sep. 2020, doi: 10.1002/ett.4150.

[62] N. Shamshad et al., "Enhancing brain tumor classification by a comprehensive study on transfer learning techniques and model efficiency using MRI datasets," *IEEE Access*, vol. 12, pp. 100407–100418, 2024, doi: 10.1109/ACCESS.2024.3430109.

[63] B. Olanrewaju-George and B. Pranggono, "Federated learning-based intrusion detection system for the internet of things using unsupervised and supervised deep learning models," *Cyber Security and Applications*, vol. 3, p. 100068, 2025, doi: 10.1016/j.csa.2024.100068.

[64] I. Sharafaldin et al., "CSE-CIC-IDS2018 [Data set]," Kaggle, 2022. [Online]. Available: https://doi.org/10.34740/KAGGLE/DSV/4059899