Intrusion Detection based on an Intelligent Security System using Machine Learning Methods

Svitlana Popereshnyak[1], Anastasiya Vecherkovskaya[2], and Viktoriia Zhebka[3]

[1] *National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute," 37 Prospect Beresteiskyi, Kyiv, 03056, Ukraine*
[2] *Taras Shevchenko National University of Kyiv, 24 Bohdana Gavrylyshyn str., Kyiv, 02000, Ukraine*
[3] *State University of Information and Communication Technologies, 7 Solomyanska str., Kyiv, 03110, Ukraine*

Abstract

Authors have analyzed the basic concepts of cybersecurity, and cybersecurity technologies, studied the peculiarities of artificial intelligence application in cybersecurity, analyzed applied machine learning methods, and presented the results of experimental studies of the application of machine learning methods in cybersecurity. Intrusion detection based on an intelligent security system using machine learning methods, which is designed to detect the latest malicious URLs and is extended for distributed denial of service (DDoS) attacks, has been studied. Experimental studies and performance evaluations of the investigated SIS-ID system have been carried out.

Keywords

Cybersecurity, attack, smart home, Internet of Things, machine learning, host, cybercrime, risk, threat, software.

1. Introduction

As the quantity of IoT-connected devices rises, encompassing smart thermostats, refrigerators, speakers, lamps, and locks, the concept of smart homes offers the allure of enhanced convenience and comfort [1]. Yet, the widespread integration of these smart technologies also amplifies the susceptibility to security vulnerabilities and compromises to household privacy [2]. To address these concerns, intrusion detection systems emerge as viable solutions, offering network-level safeguards for smart devices installed within residences [3–5].

Nowadays, the growth of network threats poses a challenging problem against the security mechanism in networks to protect sensitive data and its components [6]. Therefore, intelligent security systems will be advised to deploy them with machine learning techniques against the ever-increasing cyber threats. Machine learning has become a crucial cybersecurity technology for protecting computer networks and systems from cybercriminals [7].

The purpose of the study is to identify ways to improve the security mechanism using machine learning methods to detect cyberattacks.

2. Overview of Intelligent Intrusion Detection Security Systems

Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS) are regarded as crucial methodologies by cybersecurity researchers due to their efficacy in identifying and thwarting novel cyberattacks in live networks, encompassing a range of attacks and irregular activities [8].

Evaluation of input cyber data sets is essential to the effectiveness of any intrusion detection system approach. It allows evaluation of the proposed methods that are qualified to detect and prevent cyber-attacks. Preparing datasets for a network IDS can be difficult to obtain due to

CPITS-2024: Cybersecurity Providing in Information and Telecommunication Systems, February 28, 2024, Kyiv, Ukraine
EMAIL: spopereshnyak@gmail.com (S. Popereshnyak); vecherkovskaia90@gmail.com (A. Vecherkovskaya); viktoria_zhebka@ukr.net (V. Zhebka)
ORCID: 0000-0002-0531-9809 (S. Popereshnyak); 0000-0003-2054-2715 (A. Vecherkovskaya); 0000-0003-4051-1190 (V. Zhebka)

privacy issues. In addition, the biggest problem is the lack of public and accessible datasets to be used in the training phase based on machine learning techniques. In this work, the SIS-ID system was investigated by referencing datasets: DB-MALCURL using the ISCX-URL-2016 dataset (Table 1) [9] and DB-DDoS using the DDOS2019 dataset.

Table 1
Class distribution for DB-MALCURL

Type of attacks	Number of rows
Benign	7530
Spam	7735
Phishing	7945
Malware	6670
Defacement	6820

Preparing the DB-DDoS dataset. A Distributed Denial-of-Service (DDoS) attack stands out as one of the most formidable threats to network security, aiming to overwhelm networks with a flood of malicious traffic. Despite the existence of various tools designed to detect DDoS attacks, the challenge of low computational performance persists, hindering cybersecurity experts in developing intrusion detection systems that rely on machine learning techniques, which in turn depend heavily on dependable datasets. Hence, DB-DDoS is crafted utilizing a dataset to deploy SIS-ID models for detecting DDoS attacks. This dataset encompasses recent DDoS network flow attacks, detailed in Table 2. This framework encapsulates 13 classes of DDoS attacks as illustrated in the subsequent table.

Table 2
Class distribution for DB-DDoS

Type of attack	Quantity of rows
BENIGN	55665
DrDoS_DNS	55975
DrDoS_LDAP	55875
DrDoS_MSSQL	56050
DrDoS_NetBIOS	55635
DrDoS_NTP	56325
DrDoS_SNMP	56310
DrDoS_SSDP	56625
DrDoS_UDP	55930
Syn	55910
TFTP	55890
WebDDoS	55590

3. Analysis of Data and Function Realization

This subsection presents engineering methods of data and designed functions used in the implementation of the SIS-ID system. Indeed, several factors can influence the results of a machine-learning algorithm.

Therefore, at the implementation stage, data and functions are used to improve the performance of ML based on an intelligent model.

Data preprocessing. The data preprocessing method is a critical step in machine learning that aims to improve the quality of the data to enhance the extraction of useful information from the desired data. This phase prepares the samples by processing them in a proper form. Thus, in this part, the preprocessing phase has been addressed with several techniques related to converting the subsets into a readable, understandable, and clean format, as the data is collected from sources and may contain noisy, empty, or irrelevant data that can potentially reduce the performance of the SIS-ID system. The stages of data pre-processing are shown in Fig. 1 and are listed as follows:

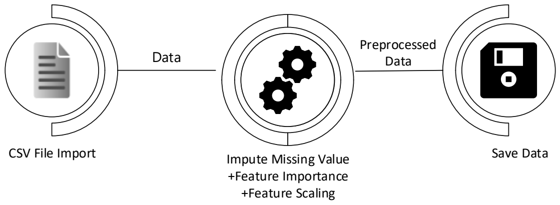


Figure 1: Workflow for data preprocessing

1. Conducting data cleansing to address missing values by replacing them with valid entries.
2. Achieving subset balance through techniques such as under-sampling and over-sampling.

Many datasets exhibit an imbalance in the number of instances across their classes. To address this issue, a sampling method has been employed utilizing the scikit-learn library [10], which encompasses two primary algorithms outlined as follows:

- Over-sampling: duplication of samples from under-represented classes by “imblearn.over_sampling import RandomOverSampler.”
- Under-sampling: removing samples from over-represented classes by “imblearn.under_sampling import RandomUnderSampler.”

3. Data transformation by coding categorical features and scaling them. As a rule, most IDS datasets include features with different scopes and ranges.

Indeed, the dataset includes features with multiple dimensions and ranges. Therefore, to overcome this problem, scaling methods should be applied to the data and they should be brought to the same scale of values [11].

Function technique. The function selection method used in the SIS-ID system to improve the efficiency of the results has been considered in this part. Therefore, a recursive function elimination based on function ranking is proposed, which uses the “feature_selection. RFECV” cross-validation method.

Selected Functions. The recommended function technique assigns an importance score or weight to each selected function in the SIS-ID system. Thus, the functions with the lowest importance score and variance are excluded from the data, and the model is trained according to this technique:

1. DB-MALCURL dataset.
2. DB-DDoS dataset.

4. Training Methodology for the SIS-ID System

The SIS-ID system under investigation is an intelligent system developed using the Python language. It has been deployed via (data.ceh.vn) to detect recent cyberattacks using BDD-MALCURL and DB-DDoS, both of which are utilized during the training phase of the system outlined in this study. The training process is detailed in the subsequent section, wherein 80% of each data source is allocated for training, representing a significant portion necessary to effectively train and optimize the model parameters. However, the remaining 20% of the data is reserved for testing purposes to ensure an unbiased evaluation of the SIS-ID. This testing set is consistently excluded from the training instances to compare the results with the actual classes. Indeed, to achieve a robust detection rate, the proposed SIS-ID follows a methodology that incorporates distinct training and testing sets, as depicted in Fig. 2.

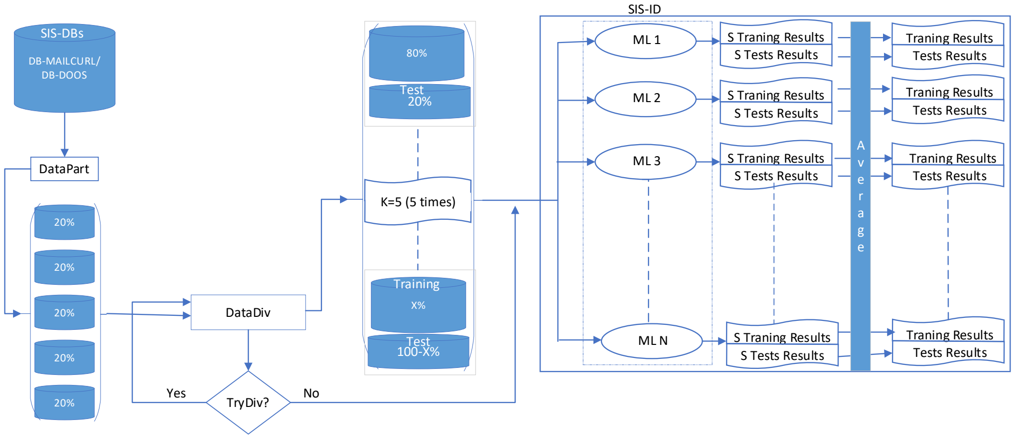


Figure 2: General architecture of the SIS-ID training methodology based on applied machine learning methods

Applied machine learning methods. Herein, we delineate the training methodologies employed in the SIS-ID proposal, incorporating various machine learning techniques validated on the aforementioned datasets, namely BDDMALCURL and DB-DDoS. Consequently, the machine learning models are chosen through controlled learning, including K-Nearest Neighbors (KNN), decision trees, and two multi-class approaches, namely OneVsRest and OneVsOne models. Additionally, ensemble methods have been devised utilizing five models: Voting, Stacking, Bagging, XGBoost, Random Forest, and Adaboost, following a predefined methodology aimed at optimizing the system across each dataset, as elaborated below.:

3. Controlled learning:
   - decision tree.
   - k-nearest neighbor algorithm.
   - multi-class techniques.
   - classifier vs. each other.
   - classifier one vs. the rest.
4. Ensemble techniques. This method amalgamates several weaker base models to construct a robust and effective model. For this experiment, six ET models were utilized, encompassing extreme gradient boosting (XGBoost), adaboost, batching, voting, random forest, and stacking.
5. Development of ensemble technique: bag classifier; voting classifier.
6. Learning without control.

The Local Outlier Factor (LOF) is considered to be one of the most widely used models in anomaly detection because of the importance of overcoming the problem of unknown traffic entering the network.

Therefore, the interest was to propose a proper SIS-ID configuration to avoid any attack that the server might face. Therefore, 60,000 records associated with benign instances were extracted from DB-DDoS to train the model on normal traffic. Hence, the system has been trained on legitimate traffic (inliers) and can be evaluated in real time for detecting any unusual activity (outliers) going forward. On the other hand, 40,000 records from multiple instances of DDoS were extracted to prove their effectiveness during the testing phase to detect any subsequent attack. As LOF operates as an unsupervised learning method, it evaluates the local density variation based on the data point about its neighboring points. Consequently, if a data point has a notably lower density compared to its neighbors, it will be classified as an outlier (indicating a potential attack), with LOF serving as the score for each anomalous sample. The model selected the location for the data point using the k-nearest neighbors method to calculate the distance and estimate the local intensity.

Training implementation. In the studied SIS-ID system, several machine learning classification models are applied in the training phase, as described in the above section. To overcome the long time required to verify each block, the SIS-ID training methodology has been applied.

Thus, the training implementation, training, and evaluation process for each model is analyzed using the cross-validation technique. It consists of two cycles for the training and evaluation phases [12].

At the beginning of the studied training system, both DB-MALCURL and DB-DDoS were divided into k convolutions ($k = 5$ in our study), which were approximately the same size. Then, in the outer loop for each iteration, we took 20% as one data convolution to be reserved for the testing phase.

Hence, the remaining convolutions ($k - 1$) are directed into the inner loop for hyperparameter tuning, functioning as an automated model that needs to be systematically chosen independently from the evaluation instances. Within each model, the inner loop entails a grid search for parameters and their values, necessitating estimation, with each parameter estimated via a cross-validation step ($k - 1$). Additionally, the hyperparameters that yield the highest average cross-validation performance are selected. Consequently, the SIS-ID is trained on the chosen data within the ($k - 1$) folds based on the optimal parameters achieved and subsequently evaluated for detection performance across a stable fold in the outer contour. Ultimately, this process is iterated k times ($k = 5$), ensuring that each outer loop data fold is utilized once, resulting in *k* performance evaluations of our system.

Optimization of the learning process was a critical aspect of developing the intelligent security system under study, given its reliance on machine learning techniques. A significant challenge we encountered was determining the most suitable method for optimizing the SIS-ID system during the training phase. Our priority was to identify a robust approach for model optimization. Consequently, we employed a hyperparameter optimization technique, which constitutes a meta-optimization task. Each trial involving a specific hyperparameter setting necessitated training the model, essentially constituting an internal optimization process. This approach enabled us to identify the optimal set of parameter combinations that significantly influenced the performance of our chosen algorithms on the validation set [13].

The hyperparameter plays an immediate role in shaping the learning process of the system. Therefore, we utilize GridSearchCV, which traverses a predefined dictionary of hyperparameters to fine-tune the model and determine the optimal parameter values. This also allows us to specify the number of cross-validation iterations for each set of hyperparameters [14, 15]. The selected parameters for the GridSearchCV object are listed as shown below:

- Estimator: selected model.
- Params_grid: parameters of the dictionary model that contains the hyperparameter.
- Evaluation: the evaluation metric.
- N_jobs: number of processes that will be executed in parallel.
- cv: The cross-validation technique's count, which in our system is set to 5.

Verbose: this is corrected to 1 to get the detailed output while we are fitting the data to the GridSearchCV object.

5. Experimental Research and Evaluation of SIS-ID Performance

Application of the SIS-ID system on DB-MALCURL. This section provides the outcomes yielded by the SIS-ID system when implemented on DB-MALCURL, employing controlled learning techniques alongside ensemble methods.

Controlled learning. The models were assessed employing various controlled learning algorithms, including KNN, and decision tree, as well as multi-class methodologies such as OneVsRest and OneVsOne models [16].

As depicted in Table 3 (let's denote Model OneVsRest—1, OneVsOne—2, KNN—3, Decision Tree—4), a compilation of performance metrics derived from each classification report of the applied models is provided. It was observed that the multi-class approaches attained the most superior performance, notably with OneVsRest achieving a precision of 98.29%, recall of 98.23%, an F1 score of 98.24%, and an accuracy rate of 98.18%.

The OneVsOne model demonstrated a precision of 98.16%, a recall of 98.06%, an F1 score of 98.07%, and an accuracy reaching 98.03%. Moreover, it was observed that the KNN model attained a precision of 96.38%, recall of 96.49%, an F1 score of 96.39%, and accuracy standing at 96.39%

Table 3
Outcomes of the controlled learning techniques implemented and evaluated with DB-MALCURL

Model	Macro Average			Accuracy
	Precision	Recall	F1-Score	
1	0.98296	0.98231	0.98246	0.98181
2	0.98167	0.98065	0.98073	0.98032
3	0.96382	0.96497	0.96395	0.96393
4	0.94812	0.94974	0.95535	0.96019

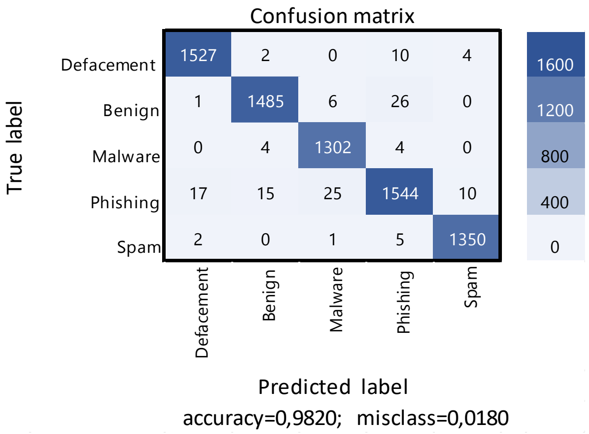
Nevertheless, the decision tree model yielded a precision of 94.81%, a recall of 94.97%, an F1 score of 95.53%, and achieved an accuracy rate of 96.01%. Furthermore, Table 4 provides a breakdown of class detection and measurement ratios using supervised learning models.

Table 4
Outcomes of identifying individual classes utilizing the measurement coefficient in conjunction with controlled learning models

Model	**Number Inst.**	**Coefficient**	**1**		**2**		**3**		**4**	
			R	**D**	**R**	**D**	**R**	**D**	**R**	**D**
Defacement	1547	Precision	98.96%	**1527**	98.96%	1526	95.50%	1527	95.50%	1502
		Recall	98.71%		98.71%		Recall		98.64%	
		F1-score	98.84%		97.08%		F1-score		98.80%	
Benign	1506	Precision	97.83%	1485	97.76%	**1487**	96.96%	1469	96.96%	1454
		Recall	98.61%		97.54%		Recall		98.74%	
		F1-score	98.21%		97.25%		F1-score		98.25%	
Malware	1334	Precision	99.39%	1302	99.23%	1297	96.13%	**1315**	96.13%	1292
		Recall	97.60%		98.58%		Recall		97.23%	
		F1-score	98.49%		97.34%		F1-score		98.22%	
Phishing	1589	Precision	95.84%	**1544**	95.53%	1540	96.27%	1421	96.27%	1431
		Recall	97.17%		89.43%		Recall		96.92%	
		F1-score	96.50%		92.72%		F1-score		96.22%	
Spam	1364	Precision	99.41%	**1350**	99.19%	1347	97.25%	1344	97.25%	1335
		Recall	98.97%		98.53%		Recall		98.75%	
		F1-score	99.19%		97.89%		F1-score		98.97%	

Let's denote Model OneVsRest—1, OneVsOne—2, KNN—3, Decision Tree—4 and R—Rate, D—Detect.

By analyzing Table 4, it has been found that one versus one achieved the most accurate model as well as the best classifier for detecting the classes of damage, phishing, and spam. The results were obtained from the confusion matrix shown in Fig. 3. Thus, among the 1547 records classified as malicious, we detected 1527 malicious, 1 benign, 0 malware, 17 phishing, and 2 spam with an accuracy rate of 98.96%, recall (98.71%), and f1 score (98.84%). In addition, among the 1506 entries classified as benign, we received 2 corruptions, 1485 benign, 4 malware, 15 phishing, and 0 spam with an accuracy rate of 97.83%, recall rate of 98.61%, and f1 score of 98.21%. Among the 1334 entries classified as malware, we found 0 corruptions, 6 benign programs, 1302 malware, 25 phishing programs, and 1 spam with an accuracy rate of 99.39%, recall rate of 97, 60% and f1 (98.49%), and among the 1589 entries we classified as phishing, we obtained 10 malware, 26 benign, 4 malware, 1544 phishing and 5 spam with a precision rate of 95.84%, recall (97.17%) and f1 (96.50%). Finally, among 1364 related to the spam attack, we achieved 4 corruptions, 0 benign, 0 malware, 10 phishing, and 1350 spams with a high ratio measurement; an accuracy rate of 99.41%, recall (98.97%), and f1 score (99.19%).



	Defacement	Benign	Malware	Phishing	Spam
Defacement	1527	2	0	10	4
Benign	1	1485	6	26	0
Malware	0	4	1302	4	0
Phishing	17	15	25	1544	10
Spam	2	0	1	5	1350

Figure 3: Confusion matrix for the OVR model on DB-MALCURL

Moreover, the one-versus-one approach has been identified as the top classifier for detecting the benign class. The findings have been derived from the confusion matrix depicted in Fig. 4. Thus, among the 1506 records classified as benign, we detected 1 corruption, 1487 benign, 4 malware, 13 phishing, and 1 spam with an accuracy of 97.76%, recall (98.74%), and f1 score (98.25%).

Furthermore, out of the 1547 entries categorized as malicious, 1526 were identified as malicious, 1 as harmless, 0 as malware, 18 as phishing, and 2 as spam.

Among the 1334 entries classified as malware, none were corrupted, 8 were benign, 1297 were identified as malware, 28 as phishing, and 1 was spam. However, among the 1589 entries classified as phishing, 11 were corrupted, 25 were benign, 6 were malware, 1540 were identified as phishing, and 7 as spam.

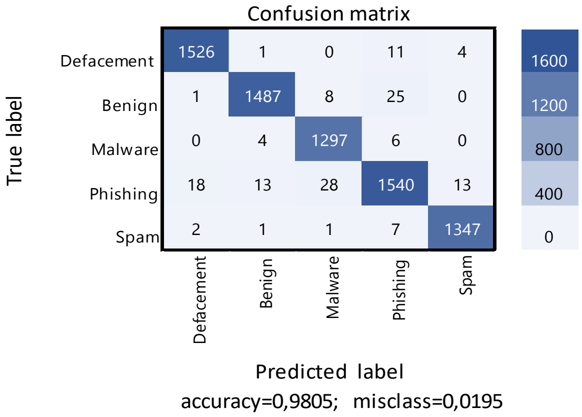


Figure 4: Confusion matrix for the OVO model on DB-MALCURL

Finally, out of the 1364 entries associated with the spam attack, the system registered 4 as corrupted, none as benign, none as malware, 13 as phishing, and 1347 as spam.

KNN emerged as the top classifier for detecting the malware class. The findings were extracted from the confusion matrix depicted in Fig. 5. Among the 1334 entries categorized as malware, we identified 0 as corrupted, 5 as benign, 1315 as malware, 13 as phishing, and 1 as spam, achieving an accuracy rate of 96.13%, recall of 98.58%, and an F1 score of 97.34%. Furthermore, out of the 1547 entries classified as malicious, we correctly identified 1527 as malicious, 1 as benign, 2 as malware, 12 as phishing, and 5 as spam. Among the 1506 entries classified as safe, we accurately classified 8 as corrupted, 1469 as benign programs, 11 as malware, 17 as phishing, and 1 as spam. Additionally, among the 1589 entries categorized as phishing, we successfully detected 58 as corrupted, 40 as benign, 39 as malware, 1421 as phishing, and 31 as spam. Lastly, out of the 1364 entries related to spam attacks, our system accurately identified 6 as corrupted, none as benign, 1 as malware, 13 as phishing, and 1344 as spam.

The outcomes from the decision tree classifier exhibited the least effectiveness in identifying all categories. Illustrated in the error matrix in Fig. 6, out of the 1547 entries classified as corruptions, we identified 1502 as corruptions, 3 as benign, 7 as malware, 28 as phishing, and 7 as spam.

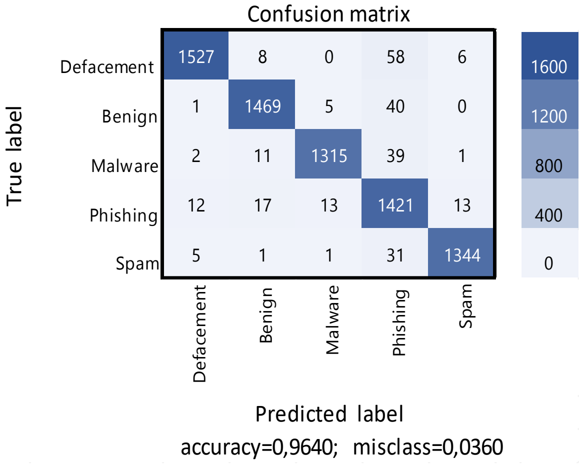


Figure 5: Confusion matrix illustrating the performance of the KNN model on DB-MALCURL

Additionally, among the 1506 entries categorized as benign, we recorded 9 as corruptions, 1454 as benign, 10 as malware, 32 as phishing, and 1 as spam.

In addition, among the 1334 entries classified as malware, we received 1 corruption, 4 benign, 1292 malware, 35 phishing, and 2 spam, while among the 1589 entries classified as phishing, we received 44 corruptions, 36 benign, 47 malware, 1431 phishing, and 31 spam. Finally, among the 1364 cases classified as spam, we received 7 malicious, 1 benign, 6 malware, 15 phishing, and 1335 spam.

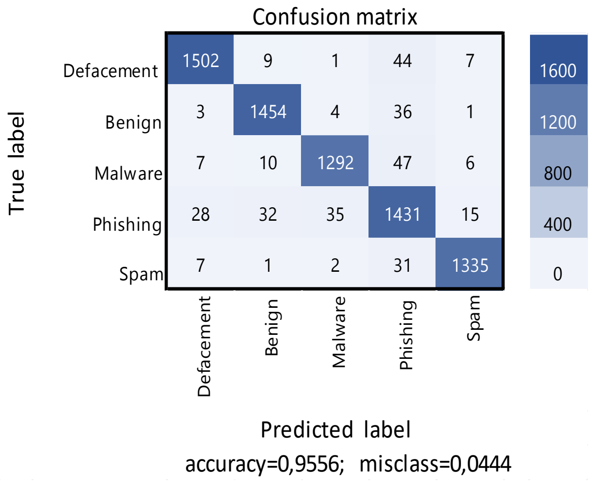


Figure 6: Error matrix for the decision tree model in DB-MALCURL

Application of the SIS-ID system on DB-DDoS. The following are the results obtained by the investigated SIS-ID system, which was used in DB-DDoS using controlled learning.

Controlled learning. The models have been tested using several controlled learning algorithms, including Decision Tree, KNN, and multi-class methods using OneVsRest and OneVsOne models. As depicted in Table 5, which provides a summary of the performance metrics acquired from the classification reports of these models. We found that the multiclass methods achieved the highest performance, with OneVSRest achieving 79.60% precision, recall (76.85%), F1 score (76.03%), and accuracy of 76.82%, while the OneVsOne models recorded precision (79.43%), recall (76.81%), F1 score (76.02%), and accuracy of 76.78%. In addition, we noticed that the Decision Tree model achieved precision (79.29%), recall (76.74%), F1 score (75.97%), and accuracy of 76.71%, and finally, the KNN model achieved precision (73.51%), recall (72.52%), F1 score (71.98%), and accuracy of 72.48%.

Table 5
Results of the applied controlled learning methods tested via DB-DDoS

Model	Macro Average			Accuracy
	Precision	Recall	F1-Score	
OneVsRest	0.7961	0.7685	0.7604	0.7682
OneVsOne	0.7944	0.7682	0.7603	0.7679
Decision Tree	0.793	0.7675	0.7597	0.7672
KNN	0.7352	0.7252	0.7199	0.7248

Let's denote Model OneVsRest—1, OneVsOne—2, Decision Tree—3, KNN—4, and R—Rate, D—Detect.

Table 6 illustrates the detection process for each class alongside the coefficient measurements employing controlled learning models. Upon analyzing Table 6, it becomes apparent that the one versus the rest method emerges as the most effective classifier for identifying the BENIGN, DrDoS_SNMP, DrDoS_SSDP, and TFTP classes.

The outcomes were derived from the confusion matrix illustrated in Figure 7. Consequently, out of the 11133 entries categorized as benign, we correctly identified 10998 as benign, while 135 cases were falsely classified as other attack classes, resulting in a precision rate of 99.54%, recall of 98.79%, and an F1 score of 99.16%.

Moreover, out of the 11262 entries categorized as DrDoS_SNMP, we correctly identified 11156 as DrDoS_SNMP, while 106 cases were falsely classified as other classes, resulting in an accuracy rate of 89.72%, recall of 99.06%, and an F1 score of 94.16%.

Table 6
Results of detecting each class using controlled learning models tested via DB-DDoS

Attack	**Number Inst**	**Coefficient**	**1**		**2**		**3**		**4**	
			R	**D**	**R**	**D**	**R**	**D**	**R**	**D**
BENIGN	11133	Precision	99.54%	**10998**	99.56%	10995	99.49%	10995	93.65%	10337
		Recall	98.79%		98.67%		98.79%		98.67%	
		F1-score	99.16%		99.08%		99.16%		99.08%	
DrDoS_DNS	11195	Precision	81.99%	5493	81.85%	5500	79.97%	5479	47.37%	**8214**
		Recall	49.07%		48.94%		Recall		49.13%	
		F1-score	61.39%		60.72%		F1-score		61.40%	
DrDoS_LDAP	11175	Precision	52.02%	8355	52.02%	**8362**	51.94%	8285	50.86%	4763
		Recall	74.77%		74.14%		Recall		74.83%	
		F1-score	61.35%		61.09%		F1-score		61.37%	
DrDoS_MSSQL	11210	Precision	71.39%	4905	71.58%	4923	71.90%	4928	58.26%	**5011**
		Recall	43.76%		43.96%		Recall		43.92%	
		F1-score	54.26%		54.56%		F1-score		54.43%	
DrDoS_NetBIOS	11127	Precision	97.52%	10550	97.57%	10551	97.28%	**10549**	95.57%	10185
		Recall	94.81%		94.81%		Recall		94.82%	
		F1-score	96.15%		96.03%		F1-score		96.18%	
DrDoS_NTP	11265	Precision	79.04%	8076	78.86%	8071	79.15%	**8091**	76.50%	7745
		Recall	71.69%		71.82%		Recall		71.65%	
		F1-score	75.19%		75.31%		F1-score		75.08%	
DrDoS_SNMP	11262	Precision	89.72%	**11156**	89.73%	11155	89.68%	11151	89.67%	11114
		Recall	99.06%		99.01%		Recall		99.05%	
		F1-score	94.16%		94.12%		F1-score		94.16%	
DrDoS_SSDP	11325	Precision	61.65%	**9072**	62.55%	8759	61.76%	8828	68.44%	6435
		Recall	80.11%		77.95%		Recall		77.34%	
		F1-score	69.68%		68.92%		F1-score		69.16%	
DrDoS_UDP	11186	Precision	70.04%	10085	69.92%	10078	70.52%	**11186**	68.84%	9209
		Recall	90.16%		89.04%		Recall		90.09%	
		F1-score	78.84%		78.71%		F1-score		78.73%	
Syn	11182	Precision	96.05%	4714	95.83%	4709	95.52%	4729	79.44%	**5158**
		Recall	42.16%		42.29%		Recall		42.11%	
		F1-score	58.60%		58.63%		F1-score		58.51%	
TFTP	11178	Precision	61.98%	**10953**	61.95%	10942	61.99%	10928	61.90%	10076
		Recall	97.99%		97.76%		Recall		97.89%	
		F1-score	75.93%		75.87%		F1-score		75.88%	
UDP-lag	11244	Precision	77.13%	6400	74.31%	6657	74.63%	**6702**	68.96%	6255
		Recall	56.92%		59.61%		Recall		59.20%	
		F1-score	65.50%		66.28%		F1-score		65.90%	
WebDDoS	11118	Precision	96.85%	11099	96.95%	**11101**	97.04%	11088	96.28%	11036
		Recall	99.83%		99.73%		Recall		99.85%	
		F1-score	98.32%		98.37%		F1-score		98.38%	

Let's denote Model OneVsRest—1, OneVsOne—2, Decision Tree—3, KNN—4 and R—Rate, D—Detect.

Additionally, among the 11325 records labeled as DrDoS_SSDP, we accurately identified 9072 as DrDoS_SSDP, whereas 2253 cases were incorrectly classified as other classes, yielding a precision rate of 61.65%, recall of 80.11%, and an F1 score of 69.68%. Lastly, out of the 11178 entries classified as TFTP, we identified 10953 as such, while 225 cases were falsely classified as other classes, resulting in a precision rate of 61.98%, recall of 97.99%, and an F1 score of 75.93%.

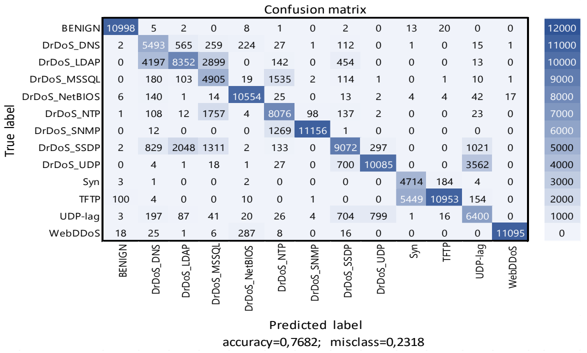


Figure 7: Error matrix for the OVR model on DB-DDoS

The one vs. one approach proves to be the most effective classifier in identifying the DrDoS_LDAP and WebDDoS classes. The results were derived from the confusion matrix depicted in Figure 8. Specifically, out of the 11175 entries classified as DrDoS_LDAP, we correctly identified 8362 as DrDoS_LDAP, while 2813 cases were falsely classified as other classes, resulting in an accuracy rate of 52.02%, recall of 74.83%, and an F1 score of 61.37%.

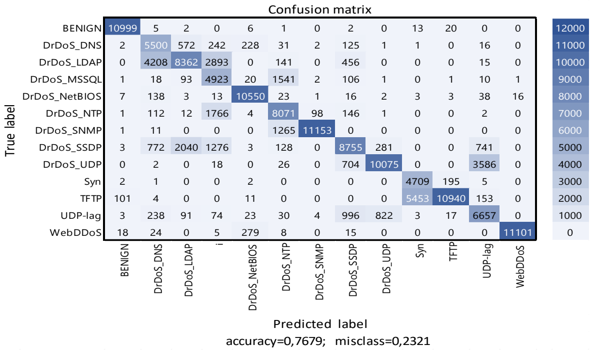


Figure 8: Error matrix for the OVO model in DB-DDoS

In addition, the decision tree model is the best for detecting the DrDoS_NetBIOS, DrDoS_NTP, DrDoS_UDP, and UDP-lag classes. The outcomes were derived from the confusion matrix depicted in Fig. 9. Thus, among the 11127 records classified as DrDoS_NetBIOS, we obtained 10549 DrDoS_NetBIOS and 578 cases for the other classes as false positives with an accuracy rate of 97.28%, recall (94.81%) and f1 score (96.03%). In addition, among the 11265 records classified as DrDoS_NTP, we obtained 8091 DrDoS_NTP and 3174 cases for the other classes as false positives with a 79.15% accuracy rate, 71.82% recall, and f1 score (75.31%).

Furthermore, out of the 11186 entries classified as DrDoS_UDP, we correctly identified 9960 as DrDoS_UDP, while 1226 cases were falsely classified as other classes, resulting in a precision level of 70.52%, recall of 89.04%, and an F1 score of 78.71%.

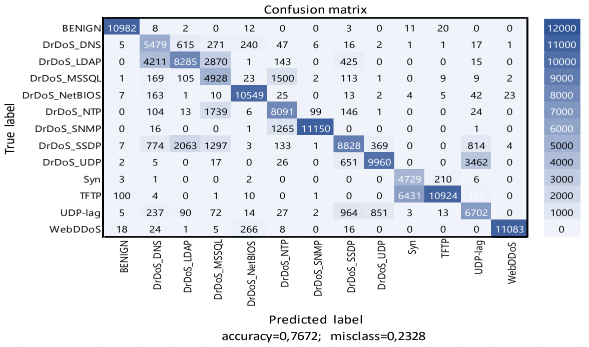


Figure 9: Error matrix for the decision tree model in DB-DDoS

Additionally, among the 11244 records labeled as UDP-lag, we identified 6702 as UDP-lag, whereas 4542 cases were falsely classified as other classes, yielding a precision rate of 74.63%, recall of 59.61%, and an F1 score of 66.28%.

Ultimately, the KNN model emerged as the top classifier for identifying the DrDoS_DNS, DrDoS_MSSQL, and Syn classes. The results were obtained from the confusion matrix depicted in Fig. 10.

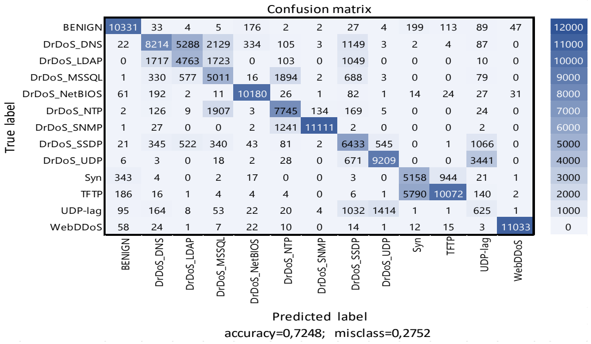


Figure 10: Error matrix for the KNN model on DB-DDoS

Additionally, among the 11210 entries categorized as DrDoS_MSSQL, we identified 5011 as DrDoS_MSSQL, while 6199 cases were falsely classified as other classes, leading to a precision rate of 58.26%, recall of 44.70%, and an F1 score of 50.59%. Lastly, out of the 11182 entries classified as Syn, we accurately identified 5158 as Syn, while 6024 cases were falsely classified as other classes, resulting in a precision rate of 79.44%, recall rate of 46.13%, and an F1 score of 58.36%.

Learning without a tutor. Next, we present the results of the local outlier model that was deployed in the tested hardware to detect unknown incoming traffic (novelty detection) (Fig. 11).

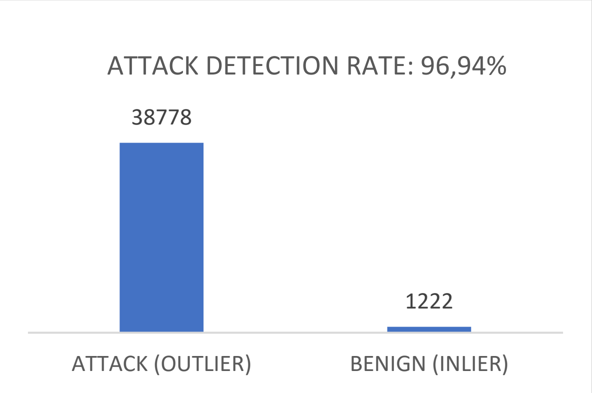


Figure 11: Evaluating the performance of the LOF model implemented in the proposed apparatus

As shown in Fig. 11, which represents the performance determination during the testing phase using DB-DDoS, the results show that among the 40,000 entries classified as subsequent attacks (outliers), we detected 38,778 as attacks and 1222 as benign (internal).

Furthermore, the model achieved an effective detection rate of 96.94%.

6. Comprehensive Examination and Assessment of the Results

The examined intelligent security system demonstrates significant success rates in forecasting both malicious URLs and DDoS attacks across ten machine-learning models. To substantiate our methodology and address any shortcomings in identifying malicious URL attacks, we present the efficacy of our system during the testing phase in Tables 7 and 8. These tables showcase the top-performing models for each attack type, along with their corresponding detection rates.

Table 7

Results of the models for identifying each attack using DB-MALCURL

Type of attack	Num. of Instance	Voting	Stacking	XG-Boost	OneVsRest	KNN
Defacement	1547	1535	1536	**1538**	1527	1527
Benign	1506	**1493**	1482	1489	1485	1469
Malware	1334	1312	1311	1313	1302	**1315**
Phishing	1589	1539	1541	1532	**1544**	1421
Spam	1364	1353	**1356**	1353	1350	1344

Therefore, upon thorough examination of these tables, we deduce that the XGBoost model demonstrated the most effective performance in identifying the defacement attack, achieving a detection rate of 99.41%.

Table 8

The detection rate achieved by the best models for each attack using DB-MALCURL

Type of attack	Model	Identification of instances	Detection speed
Defacement	XGBoost	1538	99.41%
Benign	Voting	1493	99.13%
Malware	KNN	1315	98.57%
Phishing	OneVsRest	1544	97.16%
Spam	Stacking	1356	99.41%

Similarly, the benignity voting method attained 99.13% accuracy, while KNN proved highly effective for malware detection, achieving a rate of 98.57%.

Furthermore, OneVsRest exhibited strong performance in phishing detection, with an accuracy of 97.16%, and the spam stack model showcased a detection rate of 99.41%. Furthermore, the system proved to be an improvement with decent results compared to the decision tree model, which recorded the lowest result using controlled learning, as shown in Table 9.

Table 9
Results of different ML methods tested with DB-MALCURL

Model	Macro Average			Accuracy
	Precision	Recall	F1-Score	
Voting	0.9857	0.9855	0.9856	0.9852
Stacking	0.9848	0.9847	0.9848	0.9844
XGBoost	0.9846	0.9846	0.9846	0.9843
OneVsRest	0.9828	0.9821	0.9824	0.9820
Random Forest	0.9826	0.9818	0.9821	0.9817
Adaboost	0.9818	0.9810	0.9813	0.9809
OneVsOne	0.9813	0.9805	0.9809	0.9805
Bagging	0.9799	0.9790	0.9794	0.9790
KNN	0.9642	0.9655	0.9645	0.9640
Decision Tree	0.9558	0.9568	0.9562	0.9555

As a result, we improved our system's performance by utilizing voting as the recommended ensemble model, leading to a 3% increase in precision, 2.86% in recall, 2.93% in F1 score, and 2.97% in accuracy. This was followed by stacking, which saw an increase of 2.90% in precision, 2.78% in recall, 2.85% in F1 score, and 2.88% in accuracy.

Moreover, upon juxtaposing the performance of malicious URL detection in our SIS-ID system with the findings presented by the CIC lab [17], outlined in Table 10, it becomes apparent that our method using the KNN model outperformed theirs.

We observed accuracy and recall improvement of over 1.4%, with precision and recall increasing by 2.42% and 2.55% respectively. Meanwhile, the decision tree model demonstrated enhancements of 0.55% in precision, 1.58% in recall, and 1.68% in recall. Ultimately, the random forest exhibited superior performance, showcasing a precision improvement of 3.17%, an accuracy boost of 1.26%, and an enhanced recall of 1.18%.

In contrast, we will delve into the evaluation of our system's performance utilizing DB-DDoS for detecting DDoS attacks.

Table 10
Comparative study of SIS-ID tested with DB-MALCURL

Model	CIC Laboratory			SIS-ID		
	Accuracy	Precision	Recall	Accuracy	Precision	Recall
Random Forest	>0.95	0.97	0.97	0.98174	0.9826	0.9818
Decision Tree	>0.95	0.94	0.94	0.9555	0.9558	0.9568
KNN	>0.95	0.94	0.94	0.9640	0.9642	0.9655

Upon examining Tables 11 and 12, which highlight the top-performing models for each class, we ascertain that the stacking model demonstrated the most effective performance in identifying the five designated classes, achieving the following accuracies respectively: DrDoS_LDAP: 75.45%, DrDoS_NetBIOS: 95.46%, DrDoS_NTP: 72.98%, DrDoS_SNMP: 99.18%, and DrDoS_UDP: 90.45%.

Thus, we validated our methodology by applying ensemble models.

Moreover, the OneVsRest model demonstrated superior performance in detecting BENIGN with an accuracy of 98.79% and DrDoS_SSDP with an accuracy of 80.11%, whereas the KNN model exhibited better accuracy in detecting DrDoS_DNS at 73.37% and Syn at 51.27%.

Additionally, the packetization model achieved an accuracy of 96.47% for TFTP and 98.97% for WebDDoS, while Adaboost attained 44.87% accuracy in detecting DrDoS_MSSQL. Finally, XG-Boost achieved a detection rate of 61.13% for the UDP delay attack.

Table 11
Results of the best models for detecting each attack using DB-DDoS

Model	Stacking	Bagging	XG- Boost	OneVsRest	Adaboost	KNN
BENIGN	10992	10997	10997	10998	10967	10337
DrDoS_DNS	5588	5469	5545	5493	5527	8214
DrDoS_LDAP	8431	8347	8357	8355	8328	4763
DrDoS_MSSQL	4847	4888	5015	4905	5030	5011
DrDoS_NetBIOS	10622	10536	10584	10550	10536	10185
DrDoS_NTP	8221	8180	8032	8076	7944	7745
DrDoS_SNMP	11170	11164	11162	11156	11150	11114
DrDoS_SSDP	8812	9042	8734	9072	8730	6435
DrDoS_UDP	10118	10095	10003	10085	10005	9209
Syn	4716	4676	4729	4714	4728	5158
TFTP	10912	10987	10924	10953	10963	10076
UDP-lag	6657	6424	6748	6400	6732	6255
WebDDoS	11094	11103	11095	11099	11093	11036

The system showcased an enhancement in performance metrics, yielding commendable results compared to KNN, which registered the lowest accuracy among supervised learning models, as illustrated in Table 13.

Hence, the system underwent evolution using the stacking model, resulting in enhancements as follows: 6.26% improvement in precision, 4.55% in recall, 4.29% in F1 score, and 4.56% in accuracy.

Table 12
Detection rates attained by best models for each DB-DDoS attack

Type of Attack	Model	Instances	Rate, %
Bening	OneVSRest	10998	98.79
DrDoS_DNS	KNN	8214	73.37
DrDoS_LDAP	Stacking	8431	75.45
DrDoS_MSSQL	Adaboost	5030	44.87
DrDoS_NetBIOS	Stacking	10622	95.46
DrDoS_NTP	Stacking	8221	72.98
DrDoS_SNMP	Stacking	11170	99.18
DrDoS_SSDP	OneVSRest	9072	80.11
DrDoS_UDP	Stacking	10118	91.35
Syn	KNN	5163	51.27
TFTP	Bagging	10987	96.47
UDP-lag	XG-Boost	6748	61.13
WebDDoS	Bagging	11103	98.97

Comparing the studied DDoS attack detection approaches with the performance of the IDS associated with the CIC lab [17], shown in Table 14, it is shown that the system under study showed better measurements using the random forest model; accuracy up to 2.64%, recall 11.87% and f1-Score (7.05%).

Table 13
Results of different ML methods tested with DB-DDoS

Model	Macro Average			Accuracy
	Precision	Recall	F1-Score	
Stacking	0.79775	0.77077	0.7628	0.77047
Voting	0.79639	0.76899	0.76091	0.76869
Bagging	0.79762	0.76898	0.76072	0.76869
XGBoost	0.79455	0.76894	0.76135	0.76863
Random Forest	0.79649	0.76865	0.76051	0.76837
OneVsRest	0.79609	0.76853	0.7604	0.76824
OneVsOne	0.79436	0.76819	0.76027	0.76788
Adaboost	0.79343	0.76771	0.7603	0.7674
Decision Tree	0.79298	0.76749	0.75975	0.76719
KNN	0.73518	0.72524	0.71987	0.72484

Table 14
Comparative study of SIS-ID tested through DB-DDoS

	CIC Laboratory			SIS-ID		
Model	Precision	Recall	F1-Score	Precision	Recall	F1-Score
Random Forest	0.77	0.65	0.69	0.796485	0.768653	0.760505
Decision Tree	0.78	0.65	0.69	0.792984	0.76749	0.759745

7. Real-Time Hardware Modelling

The following are the results of the verification phase of the SIS-ID system under study, which was tested using DB-DDoS based on the configured hardware in the real-time phase. Due to the intrusion prevention system's challenge of dealing with unknown future traffic that may deter cyberattacks in the network, the goal of implementing a cybersecurity mechanism to avoid any attack that may threaten the server is applied in this experiment, as shown in Fig. 12.

Validation. The attack simulation analysis and hardware verification process were conducted to avoid the denial of service attack in the real-time phase. LOIC software was taken to perform a DOS attack and then verified based on Raspberry Pi as intelligent security hardware using the local ejection model.

So, we chose a domain name as the victim. HTTP request is the attack method, there were five threads to simulate this attack.

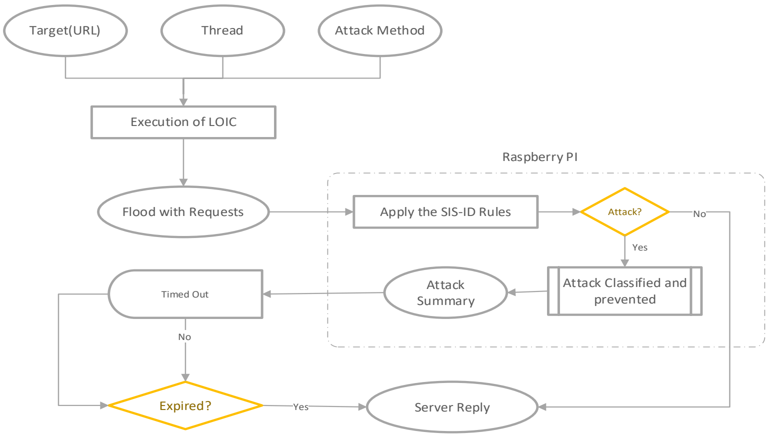


Figure 12: General architecture of the SIS-ID real-time hardware simulation

After that, a small DOS attack was carried out. On the other hand, the hardware was configured to capture the incoming packets using CICFLOWMETER to get the relevant functions related to the flows. Thus, these packets were generated and matched with our SIS-ID training system according to its rules. Furthermore, as shown in Fig. 13, the hardware proved effective in detecting the following attack as well as preventing it. The observed attack lasted 60 seconds from 12:40:08 to 12:41:08 and bombarded the victim with several unusual requests.

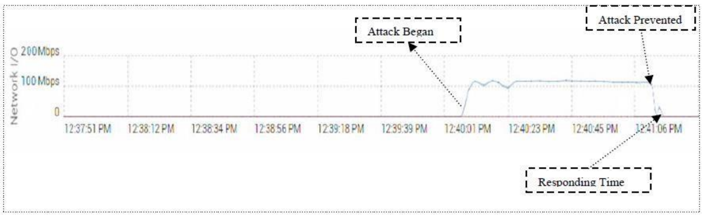


Figure 13: Effectiveness of detecting the next attack in real-time

Prevention has been achieved by utilizing the customized firewall in our equipment to avoid the detected attacks. Hence, our implemented system successfully detected all data streams containing numerous abnormal packets, enabling the identification of the IP addresses linked to the source of the threat. Thus, in this experiment, the intelligent security hardware prevented five consequence flows coming from the IP address "10.3.141.106" to the web server, which confirmed dynamic rule-based protection. Ultimately, the attack was identified as blocked, and delayed for 5 seconds as the proposed request expired to indicate that the server had not received any unusual request within the specified period.

8. Conclusions

The peculiarities of artificial intelligence applications in cybersecurity and analyses of applied machine learning methods have been studied.

Consideration has been given to the operation of a Host-Based Intrusion Detection System (HIDS) employing text mining techniques. The experimental findings regarding the application of machine learning methods in cybersecurity are presented. Four distinct machine learning techniques (KNN, SVM, Decision Tree, and MLP) were employed in the analysis to identify the most effective classification model. HIDS demonstrated the capability to detect SQLi, XSS, and directory traversal attacks. As a result, MLP achieved the highest accuracy of 90.67%. Following this, KNN attained the second-highest accuracy rate at 88.17%, succeeded by a decision tree with 86.08%. Lastly, SVM exhibited the lowest accuracy rate of 82.67%.

The exploration of intrusion detection through an intelligent security system employing machine learning techniques has been undertaken. This system is specifically designed to identify recent malicious URLs and has been expanded to encompass DDoS attacks. Experimental investigations and performance assessments of the examined SIS-ID system have been conducted.

In addition, several measurements on the achieved detection rate for each model in terms of each considered SIS-ID attack have been discussed. The studied SIS-ID system has been verified as intrusion prevention hardware with the effectiveness of preventing DOS attacks in the real-time phase.

References

[1] V. Sokolov, et al., Method for Increasing the Various Sources Data Consistency for IoT Sensors, in: IEEE 9th International Conference on Problems of Infocommunications, Science and Technology (PICST) (2023) 522–526. doi: 10.1109/PICST57299.2022.10238518.

[2] O. Bahatskyi, V. Bahatskyi, V. Sokolov, Smart Home Subsystem for Calculating the Quality of Public Utilities, in: Workshop on Cybersecurity Providing in Information and Telecommunication Systems, vol. 3421 (2023) 168–173.

[3] P. Anakhov, et al., Protecting Objects of Critical Information Infrastructure from Wartime Cyber Attacks by Decentralizing the Telecommunications Network, in: Cybersecurity Providing in Information and Telecommunication Systems Vol. 3550 (2023) 240–245.

[4] S. Popereshnyak, One Way of Testing of Lightweight Pseudorandom Number Generator for Securing the Internet of Things, IEEE 15th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (2020) 580–585. doi: 10.1109/TCSET49122.2020.235499

[5] S. Popereshniak The Testing of Pseudorandom Sequence of Small Length as a Component of the Internet of Things Security, IEEE International Conference on Advanced Trends in Information Theory (ATIT) (2019) 277–281. doi: 10.1109/ATIT49449.2019.9030471.

[6] H. Hulak, et al., Dynamic Model of Guarantee Capacity and Cyber Security Management in the Critical Automated System, in: 2nd International Conference on Conflict Management in Global Information Networks, vol. 3530 (2023) 102–111.

[7] V. Zhebka, et al., Optimization of Machine Learning Method to Improve the Management Efficiency of Heterogeneous Telecommunication Network, in: Workshop on Cybersecurity Providing in Information and Telecommunication Systems, vol. 3288 (2022) 149–155.

[8] V. Buriachok, et al., Invasion Detection Model using Two-Stage Criterion of Detection of Network Anomalies, in: Workshop on Cybersecurity Providing in Information and Telecommunication Systems, vol. 2746 (2020) 23–32.

[9] M. Mamun, et al., Detecting Malicious URLs Using Lexical Analysis, Netw. Syst. Secur. LNSC 9955 (2016) 467–482. doi: 10.1007/978-3-319-46298-1_30.

[10] F. Pedregosa, et al., Scikit-learn: Machine Learning in Python, J. Mach. Learn. Res. 12 (2011) 2825–2830.

[11] R. Mohammed, J. Rawashdeh, M. Abdullah, Machine Learning with

Oversampling and Undersampling Techniques: Overview Study and Experimental Results, 11th International Conference on Information and Communication Systems (2020) 243–248. doi: 10.1109/ ICICS49469.2020. 239556.

[12] S. Mani, et al., Medical Decision Support Using Machine Learning for Early Detection of Late-Onset Neonatal Sepsis, J. Am. Med. Inform. Assoc. 21(2) (2017) 326–36. doi: 10.1136/amiajnl-2013-001854.

[13] A. Lashkari, et al., Characterization of Tor Traffic Using Time Based Features, 3rd International Conference on Information System Security and Privacy 1 (2017) 253–262. doi: 10.5220/0006 105602530262.

[14] R. Toshniwal, K. Dastidar, A. Nath, Big Data Security Issues and Challenges, Int. J. Innov. Res. Adv. Eng. (2020).

[15] A. Saravanan, S. Bama, A Review on Cyber Security and the Fifth Generation Cyberattacks, Oriental J. Comput. Sci. Technol. (2019) 50–56.

[16] A. Salah, M. Shouman, H. Faheem, Surviving Cyber Warfare with A Hybrid Multiagent-Base Intrusion Prevention System, IEEE Potentials 29(1) (2020) 32–40.

[17] S. Singh, S Silakari, An Ensemble Approach for Cyber Attack Detection System: A Generic Framework, 14th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (2018) 79–84.