Development of Detection and Mitigation of Advanced Persistent Threats Using Artificial Intelligence and Multi-Layer Security on Cloud Computing Infrastructure

Hartono [a,1,*], Ryan Aji Wijaya [b,2], Khusnul Khotimah [b,3]

[a] University of Muhammadiyah Kotabumi, Sindangsari, North Lampung and 34517, Indonesia
[1] hartono@umko.ac.id*; [2] ryan.aji.wijaya@umko.ac.id; [3] khusnul.khotimah@umko.ac.id
* corresponding author

ARTICLE INFO

Article history
Received
Revised
Accepted

Keywords
Advanced Persistent Threat
Multi-Layer Security
Artificial Intelegence
Machine Learning
Cyber Security

ABSTRACT

This research proposes a novel approach for detecting and mitigating Advanced Persistent Threats (APTs) in cloud computing infrastruc ture, offering more comprehensive protection compared to previous methods. By integrating detection and mitigation, this study addresses the shortcomings of prior research that focused solely on detection. Based on the conducted research, Artificial Intelligence (AI) detected Cross-Site Scripting (XSS) attacks with an accuracy of 0.9951, SQL Injection (SQLI) at 0.9964, and Remote Code Execution (RCE) at 0.9876. In trials against new attacks, the detection success rates reached 70% for XSS, 98% for SQLI, and 100% for RCE. During the deployment phase, the system successfully identified 23.040 out of 108.394 requests as XSS attacks, 2.684 out of 128.750 as SQLI attacks, and 1.135 out of 46.450 as RCE attacks. The detection and mitigation methods were directly tested on cloud server experiencing APT attacks. The daily attacks on the server reached 1.980, with 663.000 requests. Additionally, the number of attacks directed at authentication or sensitive pages reached 17.913.701. Attack mitigation was tested through seven layers of security, including DNS Protection, Config Server Firewall (CSF), OWASP ModSecurity, HTTP middleware, data filter or sanitizer, template engine, and manual mitigation successfully blocking million of persistent attacks. The DNS protection layer successfully mitigated 59,000 out of a total of 19 million requests. The CSF layer mitigated 173 sources IP of DDoS attacks. The ModSecurity layer mitigated 17,916,204 attacks. All attacks were successfully mitigated before reaching the HTTP Middleware stage or next layer. The use of NIST 2.0 standards helps manage security risks through identification, protection, detection, response, and recovery. Test results indicate that this multi-layered system is more efficient and effective in detecting and mitigating attacks compared to traditional methods. However, the complexity of implementation and maintenance poses challenges that must be addressed. This research significantly contributes to a more adaptive and sustainable cybersecurity strategy.

1. Introduction

From 2019 to 2023, there was a significant increase in the number of cyberattacks occurring in Indonesia [1-2]. This surge began in 2019, reaching 290 million attacks, a 25-fold increase from 2018, when the number of attacks was only 12 million. The upward trend continued, with an extraordinary spike in 2021, reaching 1.6 billion attacks—an increase of 500% from 2020 [3]. This rise in cyberattacks has had a substantial impact on the security of system infrastructure and data in Indonesia. The high number of attacks signals that the cybersecurity defenses within Indonesia's digital ecosystem remain notably weak. Figure 1 presents a summary of cyberattack data in Indonesia over the past five years.

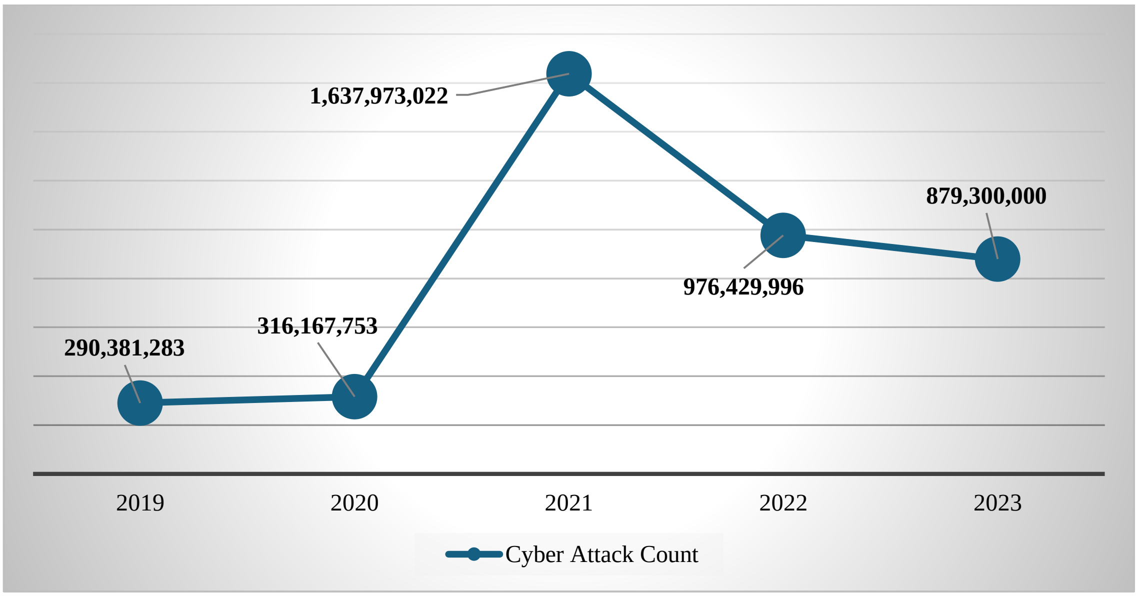

Fig. 1. Cyberattack Graph in Indonesia for 2019—2023 [3] [link]

From 2022 to 2023, the number of data breaches, particularly involving data theft or unauthorized access, remained alarmingly high. Data breaches occurred across various sectors, including government websites, private enterprises, industries, banking, e-commerce, and more. As a result, some internal or even confidential information was accessed and is suspected to have been sold by attackers. Even more concerning, these data breaches compromised millions of personal and critical data belonging to Indonesian citizens. The high number of attacks and data breaches indicates that Indonesia is in a state of cyberattack emergency. According to security OSINT (Open Source Intelligence), millions of Indonesian citizens' data have been sold by hackers on the dark web. Table 1 below presents a more detailed overview of the 10 most notable data breach cases throughout 2022–2023.

Table 1. Top-10 Data Breach Cases Throughout 2022–2023

	Data Breach	Estimated Number of Data Breaches
1	Bank Indonesia	228 GB including customer information and transaction
2	BPJS Kesehatan	720 GB of medical data and sensitive information
3	PLN	17 million PLN customer data
4	Telkom IndiHome	26 million browsing histories and search data
5	Jasa Marga	252 GB of corporate data
6	SIM Card	1.3 billion records related to SIM cards
7	Komisi Pemilihan Umum	105 million records of National Identity Numbers (NIK), family cards, etc.
8	My Pertamina	44 million user records
9	BPJS Ketenagakerjaan	19.56 million records
10	Bank Syariah Indonesia	15 million records, equivalent to 1.5 TB of data

Table References: [4], [5], [6]

Data breaches not only impact systems or infrastructure, but also affect privacy, businesses, and the financial value associated with the compromised data. The high number of cyberattacks and data theft indicates existing vulnerabilities or weaknesses in safeguarding data and digital spaces. The urgency or motives behind this research are as follows: (1) the significant increase in attacks over the past five years, reaching 1.56 billion in 2021; (2) over the last three years, more than 100 million data breaches have occurred, involving state and personal data of Indonesian citizens; (3) the emergence of sophisticated, automated, persistent attacks that are difficult to detect and mitigate, often utilizing advanced technology; (4) the high frequency of cyberattacks reveals weaknesses in the protection of state data, highlighting the need for research contributions in this area; (5) detecting an attack does not guarantee its resolution, thus necessitating the development of effective mitigation strategies.

With the advancement of time, the methods, techniques, and technologies used for cyberattacks have also evolved. These attacks are not only becoming harder to identify, but also more difficult to mitigate. Moreover, successful detection does not always guarantee successful mitigation. Even when a system detects an attack, it may not necessarily withstand it or implement defensive actions to survive the attack. In other words, detection alone is insufficient; a strong strategy and robust infrastructure are required to effectively respond to and mitigate the ever-evolving cyberattacks. According to Kettani et al. [7], [8], innovations in cyberattack technologies have introduced new and significant threats. These threats are evolving more rapidly than many anticipated. Automated attack technologies supported by advanced hardware specifications are growing massively.

This reality indicates that an increasing number of cyberattack perpetrators are "sponsored" or have access to significant resources. With such resources, attacks can be carried out automatically, persistently, and continuously, known as Advanced Persistent Threats (APT) [9]. Common methods used in APT-based attacks [9], [10], [11] include bot-based automatic attacks, injection, AI agents of attackers, command and control, supply chain compromise, data exfiltration, evasion techniques, credential theft, and persistence. This research proposes a new approach to detecting and mitigating these evolving threats by (1) integrating detection and mitigation, (2) dynamically and separately distributing resource usage, and (3) utilizing the NIST 2.0 standard. Additionally, this research (4) applies a 7-layer security model and (5) conserves resources by leveraging external resources. Research that combines detection and mitigation is still limited, as most studies continue to separate the two processes.

Several studies have focused on cyberattack detection using machine learning (ML) and mitigation through multi-layer security. Research [12], [13], [14] ested attack detection using ML and CPU resources. However, the limitations of these studies include detection being limited to DDoS attacks and lacking specificity, less diverse evaluation methods, potential bias in using CPU data as an indicator of APT, and a lack of real-world testing, which can limit the generalizability of the findings. In contrast, this research proposes a more representative method than relying solely on CPU resources, as the use of AI/ML has been proven to enhance detection accuracy, and the multi-layered approach offers stronger resilience against APT attacks.

Detection trials in studies [15], [16], [17], [18] employed deep learning autoencoders and demonstrated accurate detection results. However, those studies had dependencies on normal data, potential overfitting, inaccuracies in detecting noisy datasets, and high development costs. Unlike these studies, this research not only focuses on detection but also on mitigation, so that once an attack is detected, mitigation actions can be taken as well. Regarding mitigation, research [19] used a 3-layer approach and behavior-based profile detection, while studies [20], [21] focused on DDoS attacks. These studies showed better results compared to individual detection, but their weakness lies in the complexity of implementation and maintenance, as well as large memory requirements. Unlike these 3-layer approaches, this research experiments with up to 7 layers of security.

2. Method

2.1. Research Flowchart

There are two developments in the research: (1) detection and (2) mitigation. The research flowchart is divided into two parts, as shown in Figures 2 and 3. Figure 2 illustrates the implementation stage of the attack detection method using AI. When the accuracy level is still low, the algorithm is retrained. Additionally, if accuracy remains low after conducting attack simulations, the attack log data is used as input to enhance the performance and accuracy of the model.

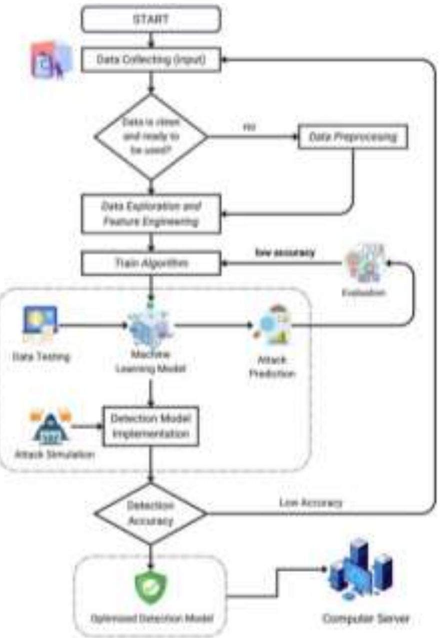

Fig. 2. Research Flowchart in Context Detection Model Development

In Figure 3, to measure the effectiveness of multi-layer security mitigation, each layer is implemented progressively to test the effectiveness of each layer and measure both successful and failed mitigated attacks. There are 7 layers used in the mitigation process: (1) DNS Protection; (2) Config Server Firewall (CSF); (3) OWASP ModSecurity (ModSec); (4) HTTP Middleware; (5) Sanitizer; (6) Template Engine; and (7) Administrator. After evaluating the effectiveness of all layers, the security layers are switched progressively to test performance and effectiveness when several layers are deactivated.

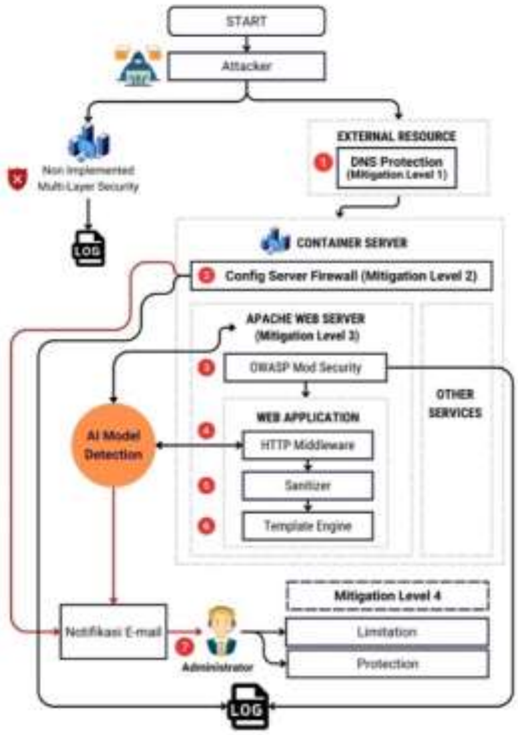

Fig. 3. Research Flowchart in the Context of Mitigation Methods

Based on the levels of mitigation, there are four mitigation levels tested: (1) DNS; (2) container server; (3) web server; and (4) admin, which is directly managed by the administrator. Mitigation at the first to third levels is carried out automatically, while the third level is performed by the administrator. If mitigation is successfully executed at the first level, the second level will not be activated as it has been eliminated, and so on. In other words, mitigation at the fourth level is likely to occur infrequently because attacks can potentially be eliminated by the first and second levels. The fourth level of mitigation relates to decision-making, which can be more complex. Figure 4 illustrates the default reactions or actions when the system detects an attack.

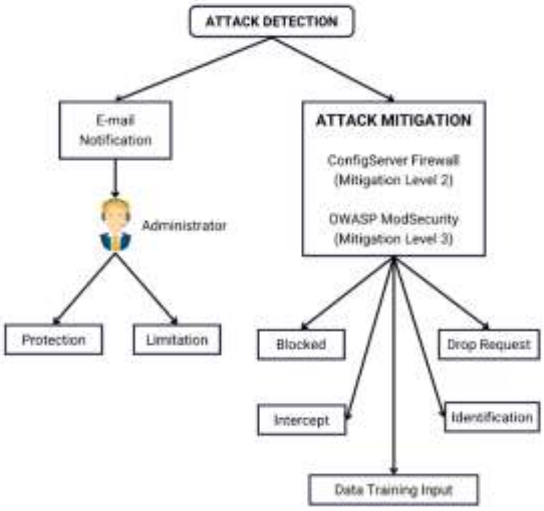

Fig. 4. Default Actions if Attack is Occuring

2.2. Research Approach

This research utilizes the NIST Cybersecurity Framework version 2.0 (NIST), released in February 2024. NIST is implemented based on standards, guidelines, and practices to assist in managing and mitigating cybersecurity risks. There are six activities (functions) in NIST: (1) govern: management and oversight of the cybersecurity program; (2) identify: identifying systems, assets, data, and cybersecurity risks; (3) protect: implementing protective measures for systems and data; (4) detect: monitoring systems to quickly detect cybersecurity incidents; (5) respond: responding to incidents swiftly and effectively; and (6) recover: restoring services and operations after an incident occurs. In this research, NIST is used more specifically in relation to the detection model and mitigation methods for attacks.

2.3. Data Sources, Tools, and Research Materials

This research utilizes 11 datasets consisting of 5 public datasets and 6 private datasets. For the private data, the researchers conducted mirroring or copying of the container server that has reached the production stage. The production stage was chosen because the security logs are more authentic and comprehensive compared to building from the development stage. The production server to be mirrored is one managed by an IT partner company and a higher education institution. Table 2 below is a detailed table of the research data.

Table 2. List of Datasets Used in the Research

	Category	Data References
1	Public Data	https://github.com/swisskyrepo/PayloadsAllTheThings https://www.kaggle.com/datasets/antonyj453/urldataset
		https://github.com/fmereani/Cross-Site-Scripting-XSS https://github.com/payloadbox/xss-payload-list
		https://github.com/payloadbox/sql-injection-payload-list
2	Private Data	Apache Log in /var/www/log
		OWASP Modsecurity in /var/log/apache2/modsec_audit.log
		Config Server Firewall log in /etc/firewalld/zones/, /etc/iptables/rules.v4, /etc/sysconfig/iptables, in /etc/firewalld/firewalld.conf, dan lain-lain

To conduct this research effectively and optimally, various research tools and materials were utilized, as detailed in table 3 below:

Table 3. Research Instrument and Material

	Categories	Instrument and Material
1	AI Development Tools	Scikit-learn, NLTK, PyTorch. Keras. Python
2	Cybersecurity Tools	Arachni dan Zed Attack Proxy
3	Multi Layer Security Technology	CloudFlare, Config Server Firewall, Mod Security
4	Cloud Computing Infrastructure	Microsoft Azure/Google Cloud Platform
5	Operating System and Hardware	Linux, Windows, Windows Subsystem for Linux

2.4. Data Analysis

Data were collected through downloads from public sources such as GitHub and Kaggle. For private data, techniques included monitoring Apache2 logs, ModSec, and CSF. Two types of data were analyzed: (1) direct attacks (XSS, RCE, and SQLI) and (2) security logs. Both were tested using five machine learning algorithms. Additionally, Natural Language Processing (NLP) techniques were employed due to the textual patterns in the attacks. The five algorithms were trained and tested using the scikit-learn library. The model used during deployment was the one with the highest accuracy. According to the documentation of scikit-learn, [22], [23], the formulas used for the five algorithms are as follows:

a) Support Vector Machine (SVM)

The SVM algorithm is commonly used for detecting cyber attacks, [24], [25], especially when the data has high dimensions and there is a clear boundary between attacks and normal activities. SVM is a data classification algorithm that separates samples by creating a hyperplane with a maximum margin. The formula for binary classification using SVM is as follows:

$$\mathrm{w} \,.\, \mathrm{x} - b = 0$$

Formula 1. Binary Classification SVM

- W is the weight vector
- XX is the feature vector
- bb is the bias

The bias is used to maximize the margin between two classes, which includes minimizing classification errors for data points that fall on the wrong side of the hyperplane. For the SVM kernel, the researcher utilizes the kernel function K(xi, xj) , such as linear, polynomial, or radial basis function (RBF) kernels:

$$\mathrm{K}(x_i, x_j) = \emptyset(x_i).\emptyset(x_j)$$

Formula 2. SVM Kernel Function

b) Gradient Boosting (GB)

GB is particularly effective in detecting attacks that exhibit complex and unclear patterns [26]. GB constructs the model incrementally by minimizing errors at each stage using weak models (typically decision trees) and employs boosting techniques to combine them. In each iteration, a new model is built to correct the residual errors from the previous model:

$$F_m(x) = F_{m-1}(x) + \eta * h_m(x)$$

Formula 3. Gradient Boosting Algorithm

- $F_m(x)$ = The prediction of the model at the m iteration for sample x.
- $h_m(x)$ = a new decision tree trained to correct the errors
- η = learning rate that controls the extent of the model's contribution to the final prediction.
- $h_m(x)$ = *weak learner*

c) Logistic Regression (LR)

LR is tested in this study due to its capability in binary classification between attacks and non-attacks [27]. LR utilizes logistic functions to classify data into two classes. The probability of predicting a class can be calculated using [28] with the following formula.

$$P\ (y = 1|x) = \frac{1}{1-e^{-(w.x+b)}}$$

Formula 4. Class Prediction Probability

- w = vector weight
- x = feature vector
- b = bias

d) Naive Bayes (NB)

NB is often applied in spam detection, phishing, and malware detection, primarily due to its simple yet effective assumption of independence among features in certain cases [29]. This algorithm is used as probabilistic classification algorithm based on Bayes' Theorem, with the assumption that each feature is independent of one another.

$$P(y|x) \propto P(y) \prod_{i=1}^{n} P(x_1|y)$$

Formula 5. Naïve Bayes

- P(y|x) = The posterior probability of class y given feature x.
- P(y) = prior probability of class y
- P (x_i|y) = *likelihood* from feature x_i given to class y.

Naive Bayes Gaussian uses a Gaussian distribution for continuous data.

$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma\frac{2}{y}}} \exp\left(-\frac{x_i - \mu_y}{2\sigma\frac{2}{y}}\right)$$

Formula 5. Naïve Bayes Distribution on Continuous Data

e) K-Nearest Neighbour (KNN)

KNN is used to detect attacks by comparing new activity patterns with known attack patterns. KNN does not have a parametric model; instead, it calculates the distance between the new data point X_{new} and the training data points using Euclidean distance, as follow:

$$d(x_i, x_j) = \sqrt{\sum_{k=3}^{n} \left(x_{i,k} - x_{j,k}\right)^{\ 2}}$$

Formula 6. K-Nearest Neighbour with Euclidean

- $d(x_i, x_j)$ = the distance between two data points x_i, x_j
- x_i, x_j = two feature vectors representing data points
- $\sum_{k=3}^{n}$ = an operator that sums the squared differences of feature vectors x_i and x_j

3. Result And Discussion

3.1 Cyber Attack Detection

Advanced Persistent Threats (APTs) employ a variety of techniques, yet they exhibit recognizable characteristics, notably the use of automated attacks. Attackers continuously utilize automated methods, leveraging scripts, bots, or software to streamline various stages of the attack. This enables attackers to execute large-scale attacks swiftly, persistently, and without interruption. Due to the employment of bots, the techniques utilized are limited to query or parameter-based attack methods, such as Cross-Site Scripting (XSS), SQL Injection (SQLI), and Remote Code Execution (RCE). Consequently, the detection of cyber attacks in this research focuses on XSS, SQLI, and RCE techniques. These three types of attacks were detected using machine learning models that were trained to achieve an accuracy rate of 0.9951 for XSS, 0.9964 for SQLI, and 0.9876 for RCE. To achieve these high accuracy rates, the researcher performed tuning and cross-validation on the detection model parameters, including feature margin, multi-algorithm testing, feature vector, letter case, and alphanumeric filter. The table 4 are the detection parameters utilized.

Table 4. Parameter Configuration of Detection Model

	Parameter Tuning	*Classifier Model*		
		XSS	SQLI	RCE
1	*Feature Margin*	6	1.5	15
2	*Algorithm*	SVM(posibility=yes)	SVM	SVM
3	*Vector Limit*	606	24	177
4	*Test Size*	0.2	0.2	0.2
5	*Total Featureset*	49232	30609	1620
6	*Lowercase*	True	True	True
7	*Alphanumeric*	False	False	False
8	*Remove Punctuation*	False	False	False

In addition to parameter tuning, this research has tested five algorithms: Naïve Bayes (NB), Logistic Regression (LR), Gradient Boosting (GB), K-Nearest Neighbor (KNN), and Support Vector Machine (SVM). The combination of parameter tuning and model testing has resulted in a total of 280 experiments, encompassing several key stages: (1) feature engineering; (2) parameter tuning; (3) model training; (4) model testing; and (5) model evaluation and optimization. These experiments were carefully designed to assess the performance of each algorithm under varying configurations and datasets. By systematically tuning the parameters, the research aimed to identify the most effective model for detecting XSS attacks. The evaluation metrics included accuracy, precision, recall, and F1-score, ensuring a comprehensive analysis of the models' effectiveness. Table 5 presents the data for five highest accuracy results based on attack technique and algorithm.

Table 5. Accuracy Levels of Algorithms Based on the Top 5 Accuracies

XSS		SQLI		RCE	
Model Name	*Accuracy Level*	*Model Rank*	*Accuracy Level*	*Model Rank*	*Accuracy Level*
SVM	0.9951	SVM	0.9964	SVM	0.9876
KNN	0.9944	KNN	0.9962	GB	0.9603
LR	0.9920	GB	0.9916	LR	0.9569
GB	0.9914	LR	0.9725	KNN	0.9525
SVM	0.9902	SVM	0.9625	NB	0.9520

In the model detection testing phase, a confusion matrix is utilized to help identify the number of correct predictions (TP/TN) as well as errors (FP/FN). The confusion matrix also provides comprehensive information about the model's performance in classifying types of attacks. Below are the confusion matrices for each type of cyber attack. The confusion matrix enables a clear comparison between predicted and actual outcomes, offering insights into the model's strengths and weaknesses in distinguishing between attack classes. Tables 6, 7, and 8 present the confusion matrices for the detection models based on their respective attack techniques.

Table 6. Confussion Matrix of XSS

	Non-payload	**Payload**
non-payload	<5596>	22
Payload	26	<4203>

	Label	**Precision**	**Recall**	**F-Measure**
0	non-payload	0.995375	0.996084	0.995730
1	payload	0.994793	0.993852	0.994322

Table 7. Confussion Matrix of SQLI

	Non-payload	**Payload**
non-payload	<3900>	2
payload	23	<2197>

	Label	**Precision**	**Recall**	**F-Measure**
0	non-payload	0.994137	0.999487	0.996805
1	payload	0.999090	0.989640	0.994343

Table 8. Confussion Matrix Serangan RCE

	Non-payload	**Payload**
non-payload	<208>	1
payload	3	<112>

	Label	**Precision**	**Recall**	**F-Measure**
0	non-payload	0.985782	0.995215	0.990476
1	payload	0.991150	0.973913	0.982456

To ensure the reliability of the model and prevent underfitting and overfitting, the model is tested on different datasets or datasets that have not been recognized by the model. This dataset contains queries and parameters for XSS, SQL Injection (SQLI), and Remote Code Execution (RCE) attacks. Evaluation and optimization also take execution time into account. The results of the attack detection performed by each detection model can be seen in Table 9.

Table 9. Results of Attack Detection by Each Model

	Detection Model	***Total Data***	***Valid Prediction***	***Invalid Prediction***	***Processed Time***	***Percentage***
1	XSS	10917	7705	3212	0:01:14.685927	70%
2	SQLI	33727	33387	340	0:00:06.087031	98%
3	RCE	323	323	0	0:00:00.535513	100%

The detection model that has been trained, tested, and optimized is deployed on a cloud server. This model operates by analyzing two types of data: (1) traffic requests and responses managed by the HTTP Middleware, and (2) access logs from the Apache2 web server or domlogs. For the first type of data, the detection model is integrated within the scope of a web-based application, utilizing the HTTP Middleware feature of the Django web framework. The second type operates within the web server environment, leveraging the access logs generated by the server. Initially, the number of access logs before data cleaning was 17,784,770. The detection model testing focuses on paths or requests, and after removing duplicate path data, the number of detected records decreased to 131,434. The performance of the detection model on incoming traffic requests is as follows.

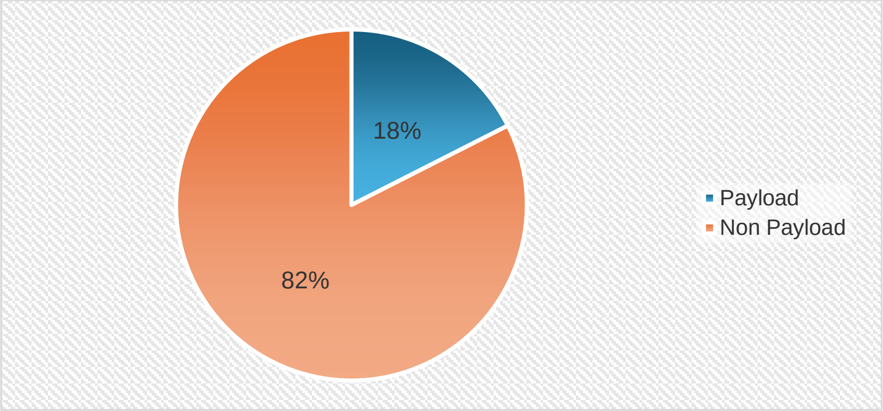

Fig. 5. Payload dan Non Payload XSS Percentation

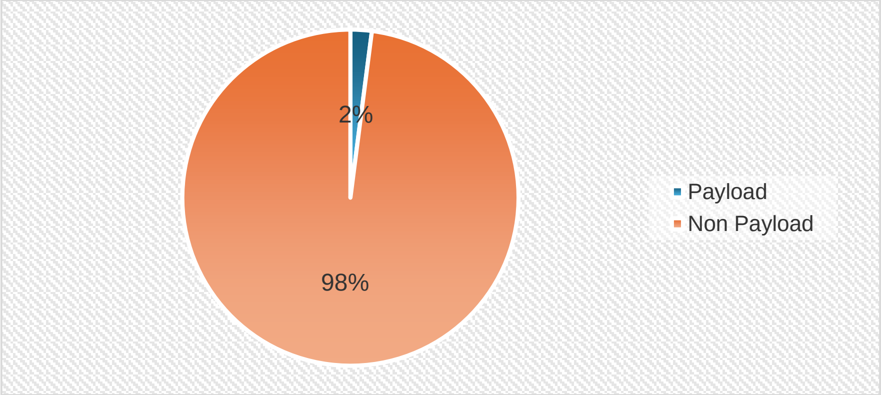

Fig. 6. Payload and Non Payload SQLI Percentation

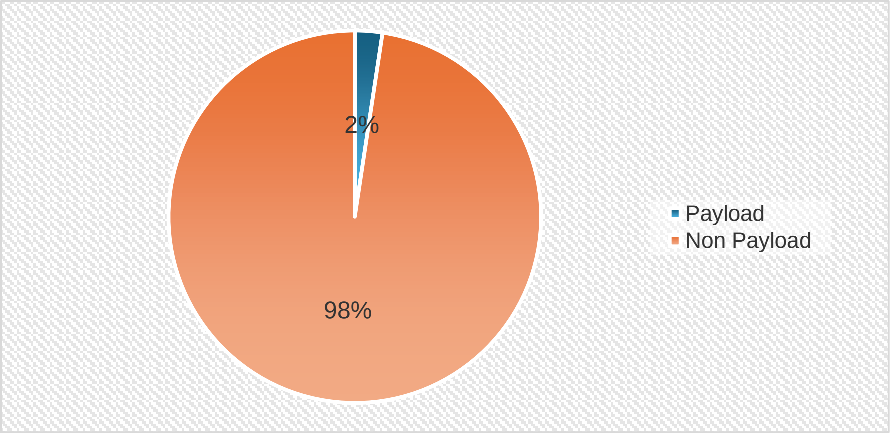

Fig. 7. Payload dan Non Payload RCE Percentation

The recap results for XSS, SQLI, and RCE attacks is presented in Table 10.

Table 10. The Detection Result of XSS, SQLI, and RCE

	Detection Model	*Detected as Non-Payload*	*Detected as Payload*
1	XSS	108.394 (82.47%)	23.040 (17.52%)
2	SQLI	128.750 (97.95%)	2.684 (2.04%)
3	RCE	46.450 (97.61)	1.135 (2.38%)

3.2 Cyber Attack Mitigation

To obtain data that is truly real and representative, a multi-layer security method was directly implemented on a cloud server that was experiencing APT attacks. In this case, the specifications of the server used are CentOS 7 Minimal with an 8-Core CPU, 8 GB RAM, and 160 GB Disk. Cyberattacks were highly persistent on this server. During the mitigation testing period, the server experienced several downtimes when certain layers were disabled. This can serve as an indicator that the implementation of layered security has a significant impact. The following is an explanation of each security layer.

DNS Protection

The DNS protection serves as the first layer. In this layer, the researcher utilized Cloudflare's DNS service. This layer validates each request sent, ensuring that the requests are not generated by bots or automated devices. Based on analytical data and logs, the number of incoming requests each month exceeds 19 million, with 59,000 detected as attacks successfully mitigated by the first layer. The number of attacks effectively handled by this layer is quite significant, as follows:

Table 11. Details of the Number of Attacks Detected and Mitigated by DNS Protection

	Attack Category	**The Number of Detected and Mitigated Attacks**
1	Average Daily Attacks (24 hours)	1.980 out of 663.000 total requests
2	Average Monthly Attacks	59,000 out of 19 million total requests
3	Country of Origin of Attacks	Indonesia, Singapore, United States, India, and Taiwan
4	Types of Mitigated Attacks	Bad IP (767/1.24%) and *Unclassified* (58.662/98.37%)
5	*Under Attack Mode*	*Active*

Config Server Firewall (CSF)

CSF is the second layer of security located at the server firewall level. This layer is designed to handle attacks that penetrate the first security layer. CSF blocks IPs that send an abnormal number of requests. Figure 8 provides a detailed breakdown of the attacks that were successfully mitigated.

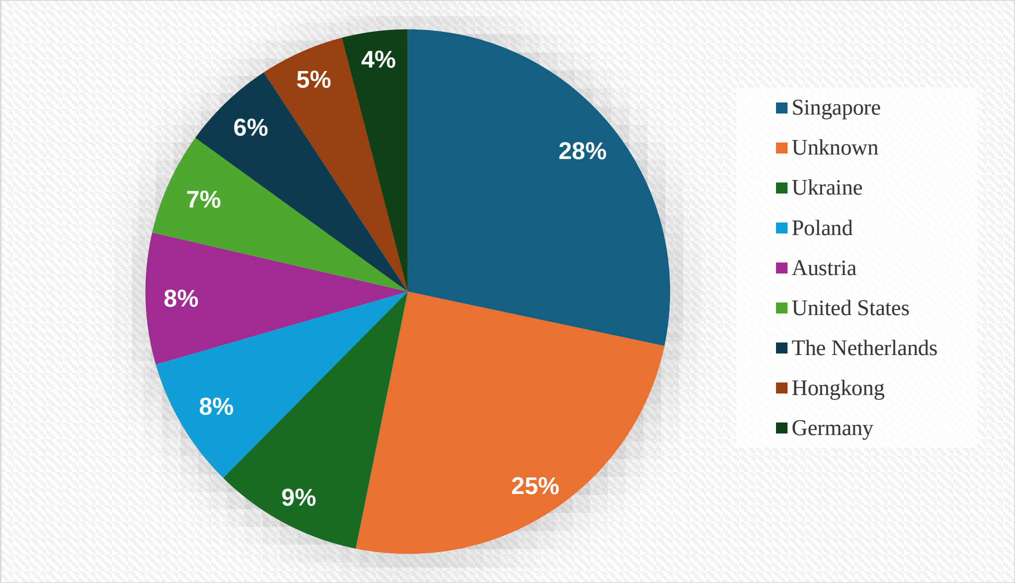

Fig. 8. Percentage of Attacks by Country of Origin Mitigated by CSF

ModSecurity

ModSecurity is the third layer that operates on the Apache web server side. The data analyzed is taken from May 30, 2023, to September 11, 2024, totaling 19,231,556 entries. After data cleaning, this number is reduced to 17,916,204 entries. Figures 9 and 10 below provide a summary of the mitigations implemented within the ModSecurity framework.

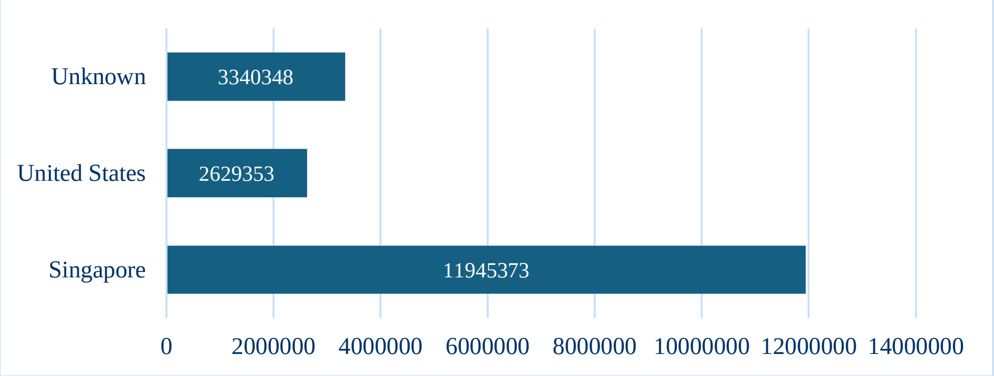

Fig. 9. Percentage of Attacks Mitigated by ModSecurity Based on Country of Origin.

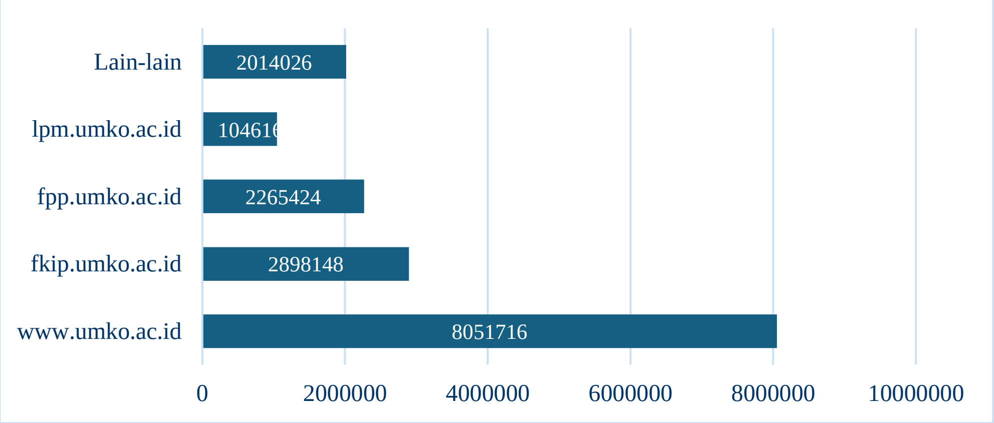

Fig. 10. Number of Domains Attacked and Mitigated by ModSecurity

One of the indicators of APT attacks is the massive, persistent, and large-scale submission of requests to search for vulnerabilities and inject subsequent attack codes. This phenomenon is illustrated in Figure 11, which shows abnormal activity, as the number of POST methods is approximately 19 times higher than that of GET methods.

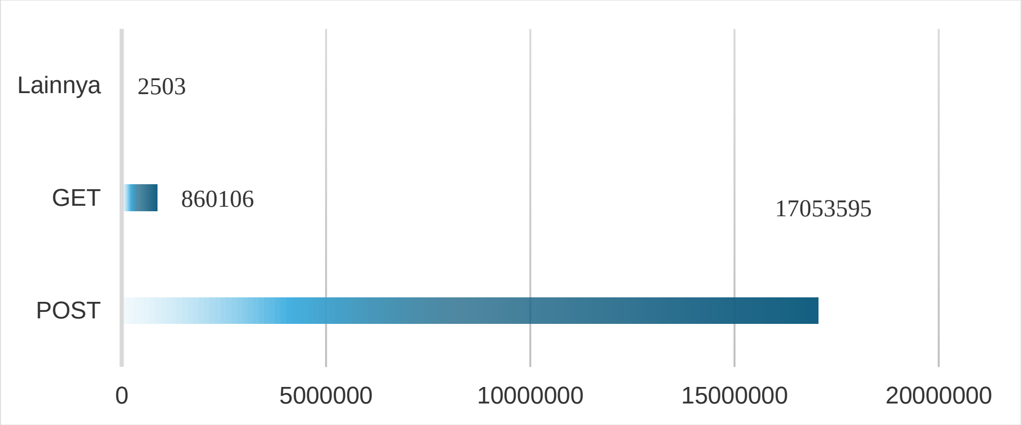

Fig. 11. Number of HTTP Methods Mitigated by ModSec

In addition to the high volume of POST methods, APT attacks are also indicated by the dominance of access paths that are abnormally directed towards sensitive pages such as /wp-login.php and /xmlrpc.php. These path are used by Wordpress.

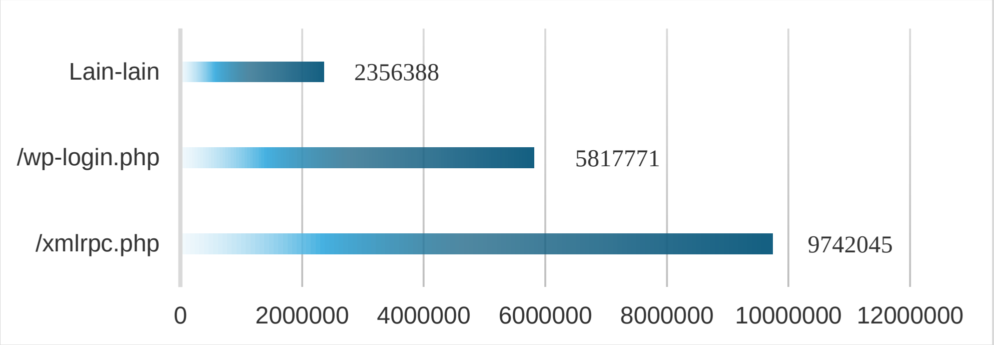

Fig. 12. Number of Illegitimately Accessed Paths Mitigated by ModSec

Validation of Mitigated Attack Techniques

In addition to being used for mitigation, ModSecurity logs are also utilized to validate the techniques or methods employed by attackers. The detection models for XSS, SQLI, and RCE are used to classify the techniques and number of incoming attacks based on the type of attack. Before classifying the attack techniques, the ModSecurity log data is cleaned by removing duplicate columns or paths, as the detection model classifies based on these paths. After the cleaning process, the total classified data amounted to 131,434 from the original 17,916,204 entries. Not only ModSecurity logs, but also ModEvasive logs are employed to classify DDoS attacks that occur. Based on the tests conducted, the following are the results of the classification of attack techniques.

Table 12. **Results of Successfully Detected Attacks by the Model on ModSec Logs**

	Attack Technique	Detected as Payload	Detected as Non-Payload
1	XSS	23.040 (17.52%)	108.394 (82.47%)
2	SQLI	2.684 (2.04%)	128.750 (97.95%)
3	RCE	23.040 (17.52%)	108.394 (82.47%)

Meanwhile, based on the analysis of ModEvasive log data, there were DDoS attacks with a total of 2.175.989 requests. The details of the DDoS attacks by country of origin are as follows.

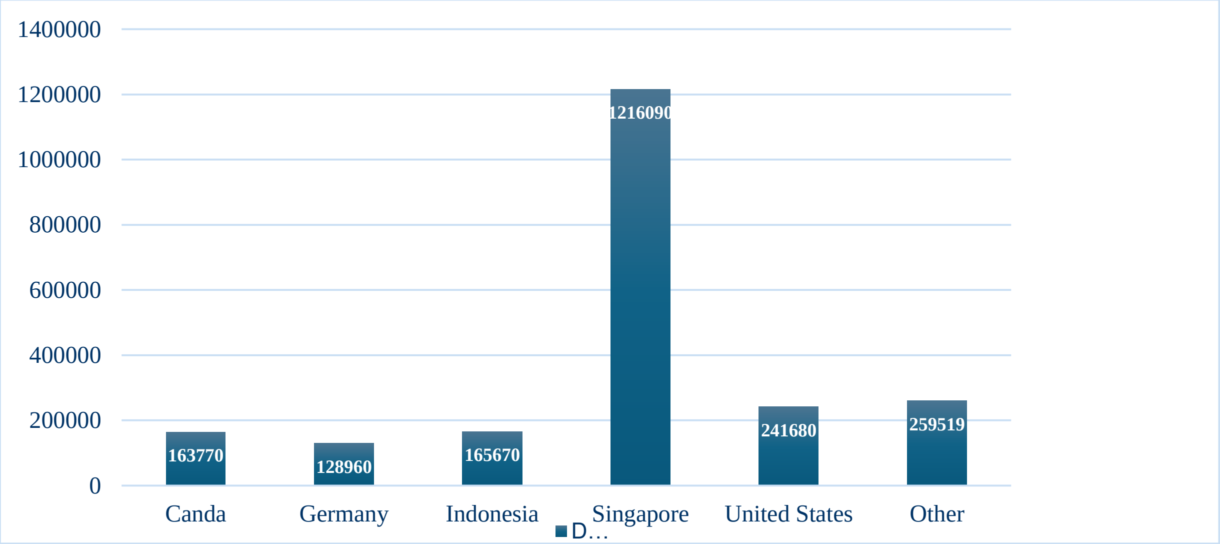

Fig. 13. Percentage of Attacks by Country of Origin from ModEvasive Logs Mitigated by ModSec

Comparison with Previous Research

Previous research on cyberattack detection has demonstrated various approaches and their limitations. Studies [12], [13], [14] for instance, employed a multi-layer protection approach in detecting Advanced Persistent Threats (APT). Although this method offers layered protection, it is limited to detecting DDoS attacks and uses CPU resources as a detection indicator, which may lead to bias and lack of generalization in real-world environments. Furthermore, studies [15], [16], [17], [18] implemented deep autoencoders for anomaly detection in networks. While these studies demonstrated accurate results in detecting anomalies, they faced potential issues of overfitting and difficulties in detecting anomalies in noisy datasets. A significant drawback of the autoencoder approach, widely used in these studies, is its dependence on normal data for training. If the data used to train the autoencoder does not represent the full range of possible attack variations, the model may fail to detect attacks that deviate from the normal pattern. In contrast, this research utilizes AI/ML to detect attacks with very high accuracy rates: 99.51% for Cross Site Scripting (XSS) attacks, 99.64% for SQL Injection (SQLI), and 98.76% for Remote Code Execution (RCE). These results highlight the system's effectiveness in identifying attacks and its adaptability to new threats. During trials involving previously unseen attacks, the detection success rates were 70% for XSS, 98% for SQLI, and 100% for RCE.

Research Limitations

This research proposes an innovative approach for detecting and mitigating APT attacks, but several limitations need to be addressed. First, the complexity of implementing and maintaining a multi-layer security system presents its own challenges. Deploying up to seven layers of security requires significant resources and precise configurations to ensure that each layer operates optimally. Second, although AI/ML-based detection systems have proven effective in identifying XSS, SQLI, and RCE attacks, these models occasionally rely on normal data for training. This dependence increases the risk of overfitting, particularly when faced with unrepresentative or noisy datasets, which can reduce detection accuracy. Additionally, while utilizing external resources can save computational power, this reliance introduces potential risks related to data security and privacy. If not properly managed, the use of external resources may become a vulnerability that attackers could exploit.

Research Implication

This study addresses several existing weaknesses by not only focusing on attack detection but also on mitigating attacks through a multi-layer security approach. This method enables a more dynamic and resilient response to APT attacks, involving the use of more efficient resources and layered protection. As a result, this research is expected to contribute more comprehensively to tackling the challenges of cyberattack detection and mitigation in the future. Moreover, the findings highlight the importance of integrating advanced technologies, such as AI and machine learning, into cybersecurity frameworks. This integration can enhance the adaptability of security systems, allowing them to evolve in response to emerging threats. Additionally, the study emphasizes the necessity for continuous monitoring and updating of security protocols to maintain effectiveness against sophisticated attack vectors. Overall, this research sets a foundation for future exploration of innovative defense strategies in cybersecurity.

4. Conclusion

This research proposes a novel approach to detecting and mitigating APT attacks on cloud computing infrastructure, offering more comprehensive protection compared to previous methods. By integrating detection and mitigation within a single system, this study overcomes the limitations of earlier research, which generally focused only on detection without addressing mitigation. The approach employs AI for detection with accuracy rates of 0.9951 for XSS, 0.9964 for SQLI, and 0.9876 for RCE. After being evaluated and tested on new attack data, the detection success rates reached 70% for XSS, 98% for SQLI, and 100% for RCE. During deployment, the model successfully detected 23,040 out of 108,394 requests as XSS attacks, 2,684 out of 128,750 requests as SQLI attacks, and 1,135 out of 46,450 requests as RCE attacks. On the mitigation side, the study tested up to seven security layers, allowing for more effective responses to automated and persistent attacks. The DNS

Protection layer successfully mitigated 767 Bad IP (1.24%) and 58,662 Unclassified attacks (98.37%). The CSF/firewall layer blocked 173 IPs from various countries. The ModSec layer detected and mitigated 17,916,204 requests and responses, which meant that subsequent layers were not required to act. The DNS protection layer successfully mitigated 59,000 out of a total of 19 million requests. The CSF layer mitigated 173 sources IP of DDoS attacks. The ModSecurity layer mitigated 17,916,204 attacks. All attacks were successfully mitigated before reaching the HTTP Middleware stage or next layer. The use of the NIST 2.0 standard helps manage cybersecurity risks through identification, protection, detection, response, and recovery. Furthermore, a key advantage of this research is the efficient use of resources via external resources and the stronger layered protection capabilities. Trials showed that the multi-layer system was able to detect and mitigate attacks with a higher success rate compared to traditional detection methods. However, the complexity of implementing and maintaining these security layers remains a challenge that must be addressed. Overall, this research offers more resilient solution to APT in cloud computing environments, making a significant contribution to the development of adaptive and sustainable cybersecurity strategies.

Acknowledgment

We would like to express our sincere gratitude to all those who have supported and contributed to the completion of this research. Special thanks to our academic supervisors for their insightful guidance, constructive feedback, and unwavering encouragement throughout the research process. We also extend our appreciation to University of Muhammadiyah Kotabumi, Ministry of Education, Culture, Research, and Technology, and collaborators who provided valuable input and technical assistance. Lastly, we are grateful for the resources and facilities made available by our institution, which were critical to the successful execution of this study. Without the support of these individuals and organizations, this research would not have been possible.

References

[1] BSSN, ‘Lanskap Keamanan Siber Indonesia 2023’, Badan Siber dan Sandi Negara Republik Indonesia, 2023, 2023.

[2] A. Yusuf, *Laporan Tahunan 2020 Honeynet Project BSSN - IHP*. Badan Siber dan Sandi Negara, 2022.

[3] H. Hartono, K. Khotimah, and A. Wibowo, ‘DETEKSI SERANGAN REMOTE CODE EXECUTION DAN CROSS SITE SCRIPTING MENGGUNAKAN MACHINE LEARNING’, *J. Inform.*, vol. 23, no. 2, pp. 229–242, Dec. 2023, doi: 10.30873/ji.v23i2.3931.

[4] L. Cloudeka, ‘10 Kasus Kebocoran Data di Indonesia dan di Dunia, Apa Saja?’, Lintasarta. Accessed: Mar. 19, 2024. [Online]. Available: https://www.cloudeka.id/id/berita/web-sec/kasus-kebocoran-data/

[5] ‘Kasus Kebocoran Data di Indonesia Melonjak 143% pada Kuartal II 2022 | Databoks’. Accessed: Apr. 09, 2023. [Online]. Available: https://databoks.katadata.co.id/datapublish/2022/08/09/kasus-kebocoran-data-di-indonesia-melonjak-143-pada-kuartal-ii-2022

[6] BeritaSatu.com, ‘Deretan Kasus Kebocoran Data yang Pernah Terjadi di Indonesia Selama 2023’, beritasatu.com. Accessed: Mar. 19, 2024. [Online]. Available: https://www.beritasatu.com/ototekno/2784168/deretan-kasus-kebocoran-data-yang-pernah-terjadi-di-indonesia-selama-2023

[7] H. Kettani and P. Wainwright, ‘On the Top Threats to Cyber Systems’, in *2019 IEEE 2nd International Conference on Information and Computer Technologies (ICICT)*, Mar. 2019, pp. 175–179. doi: 10.1109/INFOCT.2019.8711324.

[8] J. Chen *et al.*, ‘A Multi-Layer Security Scheme for Mitigating Smart Grid Vulnerability against Faults and Cyber-Attacks’, *Appl. Sci.*, vol. 11, no. 21, p. 9972, Oct. 2021, doi: 10.3390/app11219972.

[9] A. Alshamrani, S. Myneni, A. Chowdhary, and D. Huang, ‘A Survey on Advanced Persistent Threats: Techniques, Solutions, Challenges, and Research Opportunities’, *IEEE Commun. Surv. Tutor.*, vol. 21, no. 2, pp. 1851–1877, 2019, doi: 10.1109/COMST.2019.2891891.

[10] A. F. Doss, *Cyber privacy: who has your data and why you should care*. Dallas, TX: BenBella Books, Inc, 2020.

[11] Y. Maleh, M. Alazab, L. Tawalbeh, and I. Romdhani, *Big Data Analytics and Intelligent Systems for Cyber Threat Intelligence*. CRC Press, 2023.

[12] N. Mohamed, E. Alam, and G. L. Stubbs, ‘Multi-Layer Protection Approach MLPA for the Detection of Advanced Persistent Threat’, *J. Posit. Sch. Psychol.*, pp. 4496–4518, Jun. 2022.

[13] S. Ahmed *et al.*, ‘Effective and Efficient DDoS Attack Detection Using Deep Learning Algorithm, Multi-Layer Perceptron’, *Future Internet*, vol. 15, no. 2, p. 76, Feb. 2023, doi: 10.3390/fi15020076.

[14] A. A. Alahmadi *et al.*, ‘DDoS Attack Detection in IoT-Based Networks Using Machine Learning Models: A Survey and Research Directions’, *Electronics*, vol. 12, no. 14, p. 3103, Jul. 2023, doi: 10.3390/electronics12143103.

[15] F. J. Abdullayeva, ‘Advanced Persistent Threat attack detection method in cloud computing based on autoencoder and softmax regression algorithm’, *Array*, vol. 10, p. 100067, Jul. 2021, doi: 10.1016/j.array.2021.100067.

[16] K. A. Alaghbari, H.-S. Lim, M. H. M. Saad, and Y. S. Yong, ‘Deep Autoencoder-Based Integrated Model for Anomaly Detection and Efficient Feature Extraction in IoT Networks’, *IoT*, vol. 4, no. 3, pp. 345–365, Aug. 2023, doi: 10.3390/iot4030016.

[17] T. Tabassum, O. Toker, and M. R. Khalghani, ‘Cyber–physical anomaly detection for inverter-based microgrid using autoencoder neural network’, *Appl. Energy*, vol. 355, p. 122283, Feb. 2024, doi: 10.1016/j.apenergy.2023.122283.

[18] H. Torabi, S. L. Mirtaheri, and S. Greco, ‘Practical autoencoder based anomaly detection by using vector reconstruction error’, *Cybersecurity*, vol. 6, no. 1, p. 1, Jan. 2023, doi: 10.1186/s42400-022-00134-9.

[19] C. D. Xuan, D. Duong, and H. X. Dau, ‘A multi-layer approach for advanced persistent threat detection using machine learning based on network traffic’, *J. Intell. Fuzzy Syst.*, vol. 40, no. 6, pp. 11311–11329, Jan. 2021, doi: 10.3233/JIFS-202465.

[20] A. Mishra, N. Gupta, and B. B. Gupta, ‘Defensive mechanism against DDoS attack based on feature selection and multi-classifier algorithms’, *Telecommun. Syst.*, vol. 82, no. 2, pp. 229–244, Feb. 2023, doi: 10.1007/s11235-022-00981-4.

[21] T. Cai, T. Jia, S. Adepu, Y. Li, and Z. Yang, ‘ADAM: An Adaptive DDoS Attack Mitigation Scheme in Software-Defined Cyber-Physical System’, *IEEE Trans. Ind. Inform.*, vol. 19, no. 6, pp. 7802–7813, Jun. 2023, doi: 10.1109/TII.2023.3240586.

[22] S. K. Rajamani and R. S. Iyer, ‘Machine Learning-Based Mobile Applications Using Python and Scikit-Learn’, in *Designing and Developing Innovative Mobile Applications*, IGI Global, 2023, pp. 282–306. doi: 10.4018/978-1-6684-8582-8.ch016.

[23] F. Nelli, ‘Machine Learning with scikit-learn’, in *Python Data Analytics: With Pandas, NumPy, and Matplotlib*, F. Nelli, Ed., Berkeley, CA: Apress, 2023, pp. 259–287. doi: 10.1007/978-1-4842-9532-8_8.

[24] M. A. Selvan, ‘Svm-Enhanced Intrusion Detection System for Effective Cyber Attack Identification and Mitigation (1st edition)’, *J. Sci. Technol. Res. JSTAR*, vol. 5, no. 1, pp. 397–403, 2024.

[25] I. Avci and M. Koca, ‘Cybersecurity Attack Detection Model, Using Machine Learning Techniques’, *Acta Polytech. Hung.*, vol. 20, no. 7, pp. 29–44, 2023, doi: 10.12700/APH.20.7.2023.7.2.

[26] M. Douiba, S. Benkirane, A. Guezzaz, and M. Azrour, ‘An improved anomaly detection model for IoT security using decision tree and gradient boosting’, *J. Supercomput.*, vol. 79, no. 3, pp. 3392–3411, Feb. 2023, doi: 10.1007/s11227-022-04783-y.

[27] S. Srivastava and S. Raj, ‘Cyber Security Assessment and Awareness: A Statistical Modelling Approach’, in *2024 IEEE 1st Karachi Section Humanitarian Technology Conference (KHI-HTC)*, Jan. 2024, pp. 1–6. doi: 10.1109/KHI-HTC60760.2024.10482035.

[28] F. Handayani, ‘Komparasi Support Vector Machine, Logistic Regression Dan Artificial Neural Network Dalam Prediksi Penyakit Jantung’, *J. Edukasi Dan Penelit. Inform. JEPIN*, vol. 7, no. 3, p. 329, Dec. 2021, doi: 10.26418/jp.v7i3.48053.

[29] M. Elbes, S. Hendawi, S. AlZu’bi, T. Kanan, and A. Mughaid, ‘Unleashing the Full Potential of Artificial Intelligence and Machine Learning in Cybersecurity Vulnerability Management’,

in *2023 International Conference on Information Technology (ICIT)*, Aug. 2023, pp. 276–283. doi: 10.1109/ICIT58056.2023.10225910.