Computer Science and Information Technologies
Vol. 6, No. 1, March 2025, pp. 80~90
ISSN: 2722-3221, DOI: 10.11591/csit.v6i1.pp80-90

Machine learning model approach in cyber attack threat detection in security operation center

Muhammad Ajran Saputra[1], Deris Stiawan[1], Rahmat Budiarto[2]
[1]Department of Computer Engineering, Faculty of Computer Science, University of Sriwijaya, Indralaya, Indonesia
[2]Department of Computer Science, College of Computing and Information, Al-Baha University, Alaqiq, Saudi Arabia

Article Info

Article history:

Received Oct 9, 2024
Revised Feb 7, 2025
Accepted Feb 18, 2025

Keywords:

Cyber attack
Detection
Hyperparameter
Naïve Bayes
Support vector machine

ABSTRACT

The evolution of technology roles attracted cyber security threats not only compromise stable technology but also cause significant financial loss for organizations and individuals. As a result, organizations must create and implement a comprehensive cybersecurity strategy to minimize further loss. The founding of a cybersecurity surveillance center is one of the optimal adopted strategies, known as security operation center (SOC). The strategy has become the forefront of digital systems protection. We propose strategy optimization to prevent or mitigate cyberattacks by analyzing and detecting log anomalies using machine learning models. This study employs two machine learning models: the naïve Bayes model with multinomial, Gaussian, and Bernoulli variants, and the support vector machine (SVM) model with radial basis function (RBF), linear, polynomial, and sigmoid kernel variants. The hyperparameters in both models are then optimized. The models with optimized hyperparameters are subsequently trained and tested. The experimental results indicate that the best performance is achieved by the RBF kernel SVM model, with an accuracy of 79.75%, precision of 80.8%, recall of 79.75%, and F1-score of 80.01%; and the Gaussian naïve Bayes model, with an accuracy of 70.0%, precision of 80.27%, recall of 70.0%, and F1-score of 70.66%. Overall, both models perform relatively well and are classified in the very good category (75%–89%).

Corresponding Author:

Deris Stiawan
Department of Computer Engineering, Faculty of Computer Science, University of Sriwijaya
Ogan Ilir-30662, Indralaya, Indonesia
Email: deris@unsri.ac.id

1. INTRODUCTION

In recent decades, the evolution of technology roles has impacted information improvement and human creativity. This evolution attracted cyber security threats such as denial of service (DoS) attack, zero-day attack, or social engineering, which have become so ubiquitous a small part of information security threats [1]. Cybersecurity attacks not only compromise stable technology but also cause significant financial loss for organizations and individuals. Cybercrime has an immense economic impact, with an estimated loss of 8 trillion in 2023 and continuously increasing to 10,5 trillion by 2025 [2]. The number of cyberattacks exceeds 800,000 and occurs almost every 39 seconds; this threat has evolved into a serious global risk [3]. As a result, organizations must create and implement a comprehensive cybersecurity strategy to minimize further loss. It is necessary to assess the solutions that impact comprehensive activities before building the security strategy. The founding of a cybersecurity surveillance center is one of the optimal adopted strategies, known as a security operation center (SOC). The strategy has become the forefront of digital systems protection.

Other researchers announced different ways to operate and designed a SOC. Various factors, such as regulations, company strategy, and expertise, influenced the design [4]. SOC was divided into several functions. The detached function was evaluated to determine the actual performance. These functions include monitoring and detection, analysis, response and reporting, intelligence, baseline and vulnerability, and policy and signature management [5]. The framework of another study proposed performance monitoring of each function using quantitative and qualitative metrics. However, the framework did not provide a concrete evaluation mechanism that organizations can use to implement the defined metrics in the framework [6].

Threats in unstable network traffic, known as traffic anomaly, become significant challenges [7]. Anomaly not only makes the network vulnerable to attack but also has the potential to branch off the system targeted by the intruder [8]. According to the National Cyber and Crypto Agency report, there were 27,476,788 traffic anomaly incidents in Indonesia, with more than 50% indicated as malware and trojan attacks on April 2023 [9].

Several ways are used to prevent network traffic anomalies, such as log anomaly analysis and detection [10]. Network traffic anomaly detection identifies unusual patterns or behaviors in network traffic. The detection helps determine potential security threats and allows for timely countermeasures [11], [12]. Network traffic anomaly detection is an important area of network security designed to improve network security [13]. Anomaly detection can be executed manually by the identified log, but this approach is impractical because of the complexity and large amount of data available [14]. Anomaly detection is crucial because the detected data can represent significant, critical, and actionable information [15]. Therefore, an automated process is needed to analyze log classification related to traffic anomalies [16].

Data science navigates the transformation, in which machine learning is a critical aspect of artificial intelligence (AI). Data science could take an important role in finding hidden patterns in data. The methods of data science generate a new scientific paradigm and machine learning, significantly impacting the cybersecurity landscape [11]. In the literature, Veena *et al.* [17] was conducted for attack detection with a comparison of support vector machine (SVM) and k-nearest neighbors (KNN) in detecting cybercrime. The research was conducted using the ECML-PKDD 2007 dataset that contained cybercrime data in the banking sector. As a result, it was found that the SVM had the highest accuracy than KNN, which was about 98.8% and 96.47%. Similarly, Vishwakarma and Kesswani [18] discussed the intrusion detection system (IDS) by comparing the naïve Bayes algorithm with the logistic regression, KNN, decision tree, random forest, linear discriminant analysis (LDA), quadratic discriminant analysis (QDA), AdaBoost, Gradient Boosting, and Extra Trees algorithms. The research was conducted using two types of datasets, NSL-KDD and UNSW_NB15. The results showed that naïve Bayes performed the best, with the 97.1% highest accuracy using the NSL-KDD dataset and 86.9% accuracy using the UNSW_NB15 dataset.

Therefore, this study conducted an analysis and classification of cyberattacks on AI-based system logs to reduce the workload of the SOC. The naïve Bayes and SVM algorithms are used as classifiers and their performances are compared. This study uses the naïve Bayes algorithm because it is suitable for classification tasks with the advantage of having high performance on large data sets and the ability to handle many features and can generalize information from previous observation [19]. The use of SVM in this study was chosen because SVM is a machine learning model that can be used for classification and regression problems [20].

Recent research presented an innovative machine learning method to detect anomalies in IoT devices. SVM and random forest methods generated an accuracy of 99.9% and 97.9%. The research estimated that SVM's detection process was an excellent supervised learning approach [21]. Last but not least, research proposed using naïve Bayes and SVM algorithms to identify anomalies. The study showed that the naïve Bayes algorithm could identify anomalies well [22]. Based on several researches that had been carried out by raising different case studies, this our research also contributes to the world of SOC. In summary, the main contributions are summarized as follows:

- We propose the analysis and classification of cyberattacks on log systems based on the machine learning approach that is useful in optimizing the workload in the SOC.
- We compare the machine learning approach, the naïve Bayes algorithm, with the multinomial, Gaussian, and bernoulli types, and the SVM algorithm with radial basis function (RBF), linear, polynomial, and sigmoid kernels in detecting the threat of cyberattacks.

2. METHOD

The research flow illustrated in Figure 1 outlines the stages involved in addressing the problem of detecting cyber-attack threats in the SOC.

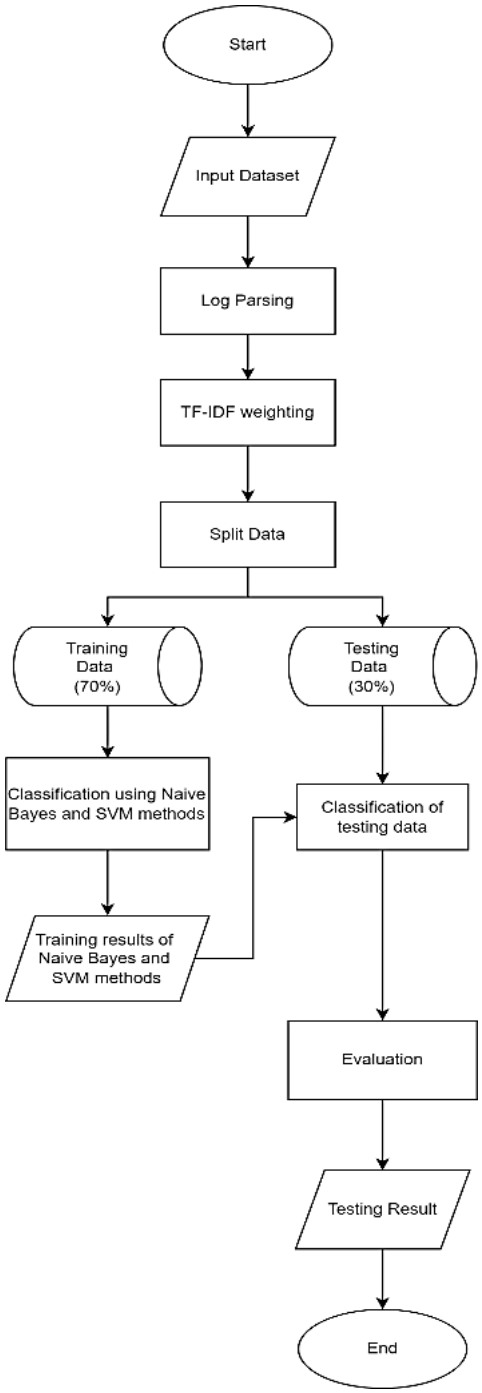

Figure 1. Overall research flow

2.1. Dataset

In this study, the data used came from Loghub, which maintains a collection of system logs using the Hadoop dataset. The dataset has four labels, namely machine_down, network_disconnection, disk_full, and normal [23]. Information about the dataset can be seen in Table 1, while samples of the dataset used can be seen in Tables 2 to 5 for each label/class.

Table 1. Dataset information

Description	Labeled	Time span	Lines	Raw size
Hadoop mapreduce job log	Yes	N.A.	394,308	48.61 MB

Table 2. Machine down class dataset sample

Date	Time	Level	Process	Component	Content	Template	Label
17/10/2015	09:21.5	INFO	main	org.apache.hadoop.mapreduce.v2.app.MRAppMaster:	Created MRAppMaster for application appattempt_1445076437777_0004_000001	Created MRAppMaster for application appattempt_<*>	machine_down
17/10/2015	09:21.9	INFO	main	org.apache.hadoop.mapreduce.v2.app.MRAppMaster:	Executing with tokens:	Executing with tokens:	machine_down
…..	…..	…..	…..	…..	…..	…..	…..
17/10/2015	09:23.4	INFO	main	org.apache.hadoop.mapreduce.v2.app.MRAppMaster:	OutputCommitter set in config null	OutputCommitter set in config null	machine_down

Table 3. Sample network disconnection class dataset

Date	Time	Level	Process	Component	Content	Template	Label
18/10/2015	20:33.8	INFO	main	org.apache.hadoop.mapreduce.v2.app.MRAppMaster:	Created MRAppMaster for application appattempt_1445144423722_0020_000002	Created MRAppMaster for application appattempt_<*>	network_disconnection
18/10/2015	20:34.2	INFO	main	org.apache.hadoop.mapreduce.v2.app.MRAppMaster:	Executing with tokens:	Executing with tokens:	network_disconnection
….	….	….	….	….	….	….	….
18/10/2015	20:34.9	INFO	main	org.apache.hadoop.mapreduce.v2.app.MRAppMaster:	OutputCommitter set in config null	OutputCommitter set in config null	network_disconnection

Table 4. Full disk class dataset sample

Date	Time	Level	Process	Component	Content	Template	Label
19/10/2015	21:32.9	INFO	main	org.apache.hadoop.mapreduce.v2.app.MRAppMaster:	Created MRAppMaster for application appattempt_1445182159119_0001_000001	Created MRAppMaster for application appattempt_<*>	disk_full
19/10/2015	21:33.7	INFO	main	org.apache.hadoop.mapreduce.v2.app.MRAppMaster:	Executing with tokens:	Executing with tokens:	disk_full
…..	…..	…..	…..	…..	…..	…..	…..
19/10/2015	21:34.9	INFO	main	org.apache.hadoop.mapreduce.v2.app.MRAppMaster:	OutputCommitter set in config null	OutputCommitter set in config null	disk_full

Table 5. Normal class dataset sample

Date	Time	Level	Process	Component	Content	Template	Label
19/10/2015	49:51.5	INFO	main	org.apache.hadoop.mapreduce.v2.app.MRAppMaster:	Created MRAppMaster for application appattempt_1445182159119_0012_000001	Created MRAppMaster for application appattempt_<*>	Normal
19/10/2015	49:51.8	INFO	main	org.apache.hadoop.mapreduce.v2.app.MRAppMaster:	Executing with tokens:	Executing with tokens:	Normal
…..	…..	…..	…..	…..	…..	…..	…..
19/10/2015	49:53.5	INFO	main	org.apache.hadoop.mapreduce.v2.app.MRAppMaster:	OutputCommitter set in config null	OutputCommitter set in config null	Normal

2.2. Log parsing

Log parsing is modeled as a clustering problem, where log messages describing the same system should be grouped into similar clusters [24]. The problem of log parsing is how to accurately and efficiently separate unstructured log messages into different groups by designing similarity metrics for log messages and new clustering approaches [24]. The implementation architecture of log parsing can be seen in Figure 2 [25].

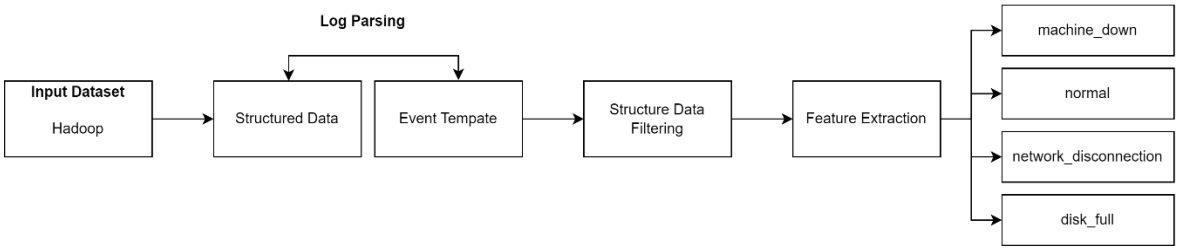

Figure 1. Log parsing architecture

2.3. Word weighting using TF-IDF method

Next, the term frequency-inverse document frequency (TF-IDF) stage is carried out because the log data is in the form of sentences (text) for word weighting [26]. The filtered log dataset is then further processed to extract discriminative and numeric features by adapting the TF-IDF algorithm [25]. TF-IDF was chosen because the method gives greater weight to terms that appear less frequently in a document and reduces the importance of terms that appear more frequently, resulting in compact numerical features [25]. The TF-IDF algorithm is used to numerically assess the relevance of words in documents. The frequency score assigned to a word with TF-IDF determines the importance of the word to the document based on the frequency of the word [27]. The TF-IDF method enables the identification of key terms that are unique to a particular document, which contributes significantly to tasks such as text analysis, information retrieval, and document classification [28].

First, a function is created to process the text given as input, which includes replacing the character "/" with a space. Next, the tokenization process is carried out on the X_train_token and X_test_token data. Then, a TfidfVectorizer object is created which is used to transform a collection of text documents into a TF-IDF matrix. TF-IDF is a technique commonly used in natural language processing to measure the importance of a word in a document relative to a collection of other documents.

After that, data transformation is performed to obtain feature names from TF-IDF data and calculate the frequency of occurrence of words in training and testing data. Thus, both representations (TF-IDF and frequency) can be used for further analysis, such as classification. The results of the TF-IDF process are stored in CSV file format, and the results of the TF-IDF process are snipped and displayed in Figure 3.

	0026013367	015954664	018885355	019536257	019864146	021491531	02377111	02735332	027681602	028498698	...	webapp	webapps	webproxy	will	with	writer	xml	yarn	yarn_am_rm_token	label
D1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	Normal
D2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	Normal
D3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	Normal
D4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	Normal
D5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	Normal
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
D3996	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.352933	0.0	0.0	0.0	0.0	network_disconnection
D3997	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	network_disconnection
D3998	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	network_disconnection
D3999	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	network_disconnection
D4000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	network_disconnection

4000 rows × 2691 columns

Figure 2. Result of TF-IDF process

The next stage is TF-IDF weighting, showing relevant words (with weights greater than 0) for the first five examples of the data used. Each word and its weight are presented in a table using a Pandas DataFrame, that is useful for further analysis of the features that contribute to the model in the context of classification or clustering. An example of TF-IDF weight on the 1st data from the test data can be seen in Figure 3.

2.4. Split data

At this stage, the data is divided into training data and testing data. The proportion of data division used in this study is 70% for training data and 30% for testing data. This division is based on previous research [25], which shows that using the proportion 70%:30% produces an accuracy of 100%.

	Word	Weight
0	10	0.153516
1	11	0.564395
2	after	0.247785
3	assignedmaps	0.223531
4	assignedreds	0.223531
5	completedmaps	0.223531
6	completedreds	0.223531
7	contalloc	0.223531
8	contrel	0.223531
9	hostlocal	0.223531
10	pendingreds	0.223531
11	racklocal	0.223531
12	scheduledmaps	0.223531
13	scheduledreds	0.223531
14	scheduling	0.216487

Figure 3. Example of weigh TF-IDF on the first data

2.5. Anomaly detection using naïve Bayes and support vector machine

Next, the model building stage is carried out to detect anomalies. At this stage, the model is built using naïve Bayes and SVM by utilizing 70% of the training data. Classification is carried out using the naïve Bayes and SVM models. The naïve Bayes methods used include naïve Bayes Gaussian, naïve Bayes Bernoulli, and naïve Bayes multinomial. In naïve Bayes Gaussian doesn't require complex parameter searching, which reduces the variance of the model and naïve Bayes Gaussian supports incremental learning [29]. While in naïve Bayes Bernoulli, model primarily focuses on searching for vector features that are binary [30]. Naïve Bayes Multinomial is a classification method with probability, which predicts future opportunities based on previous experience so it is known as Bayes Theorem [31].

While in the SVM model, various kernels are used, namely linear kernels, polynomial kernels, sigmoid kernels, and RBF kernels. The classification results of the two models will be compared. Linear kernel in SVM can guarantee global optimization for regression or classification problems in small-to-large datasets [32]. Unlike the linear kernel, the polynomial kernel does involve taking the inner product from a higher dimension space. Unlike the polynomial kernel, which looks at extra dimensions, RBF expands into and infinite number of dimensions [33]. Sigmoid kernel functions are commonly implemented, these functions are not positive semi-definite for certain values of these kernel parameters. Consequently, the parameters γ (gamma) and c must be chosen carefully to avoid errors in the results obtained [34].

In this study, hyperparameter implementation was carried out using GridSearchCV for the SVM model with RBF kernel. With C and gamma parameters, the model will be tested to find the best combination that provides the best performance on the training data. After finding the optimal parameters, the model will be used to predict the class on the test data. This is a common approach to improving the accuracy of machine learning models with hyperparameter optimization. The hyperparameters used can be seen in Table 6.

Table 6. Grid SVM parameters

Parameter	SVM			
	Sigmoid kernel	RBF kernel	Linear kernel	Polynomial kernel
C	0.1	0.1	0.1	0.1
C	1	1	1	1
C	10	10	10	10
C	100	100	100	100
gamma	0.001	0.001		scale
gamma	0.01	0.01		auto
gamma	0.1	0.1		
gamma	1	1		
degree				2
degree				3
degree				4

From the parameters used, the best parameters for the RBF kernel are obtained. The RBF kernel gets the best parameters, namely C =100, class_weight = 'balanced', gamma =0.1. The value of C =100 indicates that the model focuses more on reducing classification errors so that it is more sensitive to outliers. class_weight = 'balanced', helps handle class imbalance problems by giving more weight to underrepresented classes. A value of 0.1 indicates that the influence of one data is broader, helping the model recognize more complex patterns. From the parameters used, the best parameters for the linear kernel are obtained. In the

linear kernel, the best parameters are C =100, class_weight = 'balanced', kernel = 'linear'. The value of C =100 indicates that the model focuses more on reducing classification errors so that it is more sensitive to outliers. class_weight = 'balanced', helps handle class imbalance problems by giving more weight to underrepresented classes. kernel = 'linear' indicates that the model uses a linear separator function. From each parameter used, the best parameter for the polynomial kernel is obtained. In the polynomial kernel, the best parameter is C =10, class_weight = 'balanced', degree =2, kernel = 'poly'. The value of C =10 shows a little more tolerance for misclassification. This can help the model not overfit. class_weight = 'balanced', helps deal with the problem of class imbalance by giving more weight to underrepresented classes. degree =2 indicates that the model uses a polynomial function of degree 2, which allows the model to capture nonlinear interactions between features. kernel = 'poly' indicates the use of a polynomial kernel that is suitable for data with non-linear relationships.

From each parameter used, the best parameter for the polynomial kernel is obtained. In the polynomial kernel, the best parameter is C =10, class_weight = 'balanced', gamma =1, kernel = 'sigmoid'. The value of C =10 shows a little more tolerance for misclassification. This can help the model not overfit. class_weight = 'balanced', helps deal with the problem of class imbalance by giving more weight to underrepresented classes. Gamma =1 means that the influence of each data point is quite significant, and can produce non-linear patterns in the data. Kernel = 'sigmoid' indicates the use of the sigmoid activation function which can provide non-linear characteristics for class separation.

2.6. Evaluation

In the model evaluation using confusion matrix which produces accuracy, precision, recall, F1-score and false positive rate (FPR) values. Accuracy measure is calculated by taking all the true predictions and dividing them among all the predicted values, including the true predictions [35]. Precision is measuring the number of correctly predicted positive rate divided by the total predicted positive rates [36]. Recall or Sensitivity is the proportion of real positive cases that are correctly predicted positive [37]. F1-score is the weighted harmonic mean of the recall and precision values [38]. FPR is ratio between the incorrectly classified negative samples to the total number of negative samples [39]. Model testing consists of SVM models with RBF kernel, linear, polynomial and sigmoid as well as multinomial naïve Bayes, gaussian naïve Bayes, bernoulli naïve Bayes models.

3. RESULTS AND DISCUSSION

At this stage, the results of the model evaluation using the confusion matrix from the multinomial naïve Bayes, gaussian naïve Bayes, bernoulli naïve Bayes methods are explained. From several tests that have been conducted for the detection of cyberattack threats in SOC using the machine learning approach of the SVM method with RBF kernel, SVM with linear kernel, SVM with polynomial kernel, SVM with Sigmoid kernel and naïve Bayes from multinomial naïve Bayes, gaussian naïve Bayes and bernoulli naïve Bayes obtained varying model performance results. The overall model results can be seen in Table 7.

Table 7. Results of the overall model evaluation

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
SVM with RBF kernel	79.75	80.8	79.75	80.01
SVM with linear kernel	79.5	80.51	79.5	79.75
SVM with polynomial kernel	75.75	77.11	75.75	76.04
SVM with sigmoid kernel	78.75	80.12	78.75	79.05
Multinomial naïve Bayes	69.58	72.3	69.58	69.92
Gaussian naïve Bayes	70.0	80.27	70.0	70.66
Bernoulli naïve Bayes	63.75	68.6	63.75	64.2

Based on Figure 5, it can be explained that the SVM algorithm with the RBF kernel obtained prediction results on the machine_down label correctly classified as many as 230 data, while the prediction error with 42 data entered into the normal label, 1 data entered into the network_disconnection label, and 21 data entered into the disk_full label. In the normal label with a total of 255 data classified correctly, while the prediction error with 33 data entered into the machine_down label, 2 data entered into the network_disconnection label, and 25 data entered into the disk_full label. In the network_disconnection label with a total of 230 data classified correctly, while the prediction error with 17 data entered into the machine_down label, 15 data entered into the normal label, and 22 data entered into the disk_full label. In the disk_full label, 242 data were classified correctly, while prediction errors included 37 data that were included

in the machine_down label, 28 data that were included in the normal label, and 0 data that were included in the network_disconnection label. The results of the confusion matrix plot on the SVM method with the RBF kernel can be seen in Figure 4.

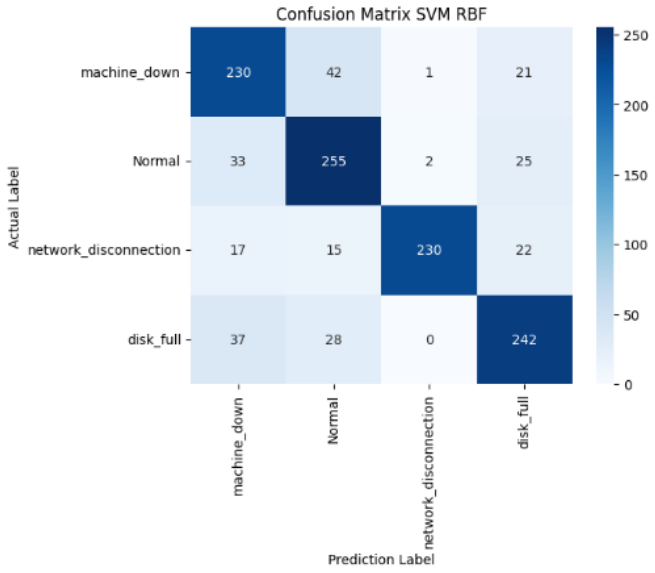

Figure 4. Confusion matrix SVM kernel RBF

The following explains the results of the FPR test of the method used. From Table 8 the results of the FPR test on the SVM with RBF kernel that the network_disconnection label has the lowest FPR of 0.33%, which means that the model is the best because the ground truth value of FPR is 0 [40]. From Table 8, the SVM method with the RBF kernel demonstrates the best performance in classifying cyber threats, achieving an accuracy of 79.75%, precision of 80.8%, recall of 79.75%, and an F1-score of 80.01%. Other SVM kernels, such as linear (79.5%), polynomial (75.75%), and sigmoid (78.75%), also yield good results. Meanwhile, Gaussian naïve Bayes achieves 70.0% accuracy, while multinomial (69.58%) and bernoulli naïve Bayes (63.75%) show lower performance. Overall, the SVM model falls within the "very good" category (75%-89%), whereas some naïve Bayes variants are categorized as "fair" or "poor." These findings indicate that the SVM with the RBF kernel is the most effective method, while naïve Bayes requires further development, such as hyperparameter tuning, to enhance its performance in detecting cyber threats.

Table 8. FPR model

Label	SVM				Multinomial naïve Bayes (%)	Gaussian naïve Bayes (%)	Bernoulli naïve Bayes (%)
	RBF kernel (%)	Linear kernel (%)	Polynomial kernel (%)	Sigmoid kernel (%)			
Machine_down	9.6	9.05	10.71	8.83	8.28	0.44	14.9
Normal	9.6	10.17	12.88	12.99	18.08	2.82	24.07
Network_disconnection	0.33	0.44	0.98	0.55	1.09	31.77	0.55
Disk_full	7.61	7.84	7.95	6.16	13.44	4.48	9.18

4. CONCLUSION

From the research that has been done on cyber-attack detection in SOC using the machine learning method approach, the following conclusions can be drawn: i) detection of cyber-attack threats in the SOC using AI-based system log data was successfully carried out using the naïve Bayes and SVM machine learning model approaches, ii) the machine learning model approach in detecting cyber-attack threats at the SOC using AI-based system log data is carried out with the naïve Bayes model with multinomial, gaussian, bernoulli types and SVM with RBF, linear, polynomial, sigmoid kernels, and iii) the results of the evaluation using confusion matrix obtained the best model performance in the SVM method with RBF kernel with an accuracy value of 79.75%, precision of 80.8%, recall of 79.75%, and F1-score of 80.01%. Meanwhile, in the naïve Bayes type there is a gaussian naïve Bayes with an accuracy of 70.0%, precision of 80.27%, recall of

70.0%, and F1-score of 70.66%. Therefore, overall, the model succeeds in classifying well and falls into the very good classification category (75%-89%), but the multinomial naïve Bayes method gets an accuracy of 69.58% in the fair category (65%-74%) and the bernoulli naïve Bayes method gets an accuracy of 63.75% in the poor category (50%-64%). Suggestions that can be given from this research, for further development, are comparisons made on various proportions of datasets used. In addition, other hyperparameter tuning can be done on SVM and naïve Bayes models so that the best parameter combination can be seen and improve model performance.

ACKNOWLEDGMENTS

The authors would like to thank the Computer Network, Enterprise and Information Security Research Group (COMNETS RG), Universitas Sriwijaya, Indonesia for providing full support for their research necessary.

FUNDING INFORMATION

Authors state no funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Muhammad Ajran Saputra	✓	✓	✓		✓	✓		✓	✓	✓	✓	✓	✓	✓
Deris Stiawan				✓		✓	✓		✓	✓				
Rahmat Budiarto						✓	✓	✓		✓				

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

Data availability is not applicable to this paper as no new data were created or analyzed in this study.

REFERENCES

[1] M. Ahsan, K. E. Nygard, R. Gomes, M. M. Chowdhury, N. Rifat, and J. F. Connolly, "Cybersecurity threats and their mitigation approaches using machine learning-a review," *Journal of Cybersecurity and Privacy*, vol. 2, no. 3, pp. 527–555, Jul. 2022, doi: 10.3390/jcp2030027.

[2] C. Brooks, "Cybersecurity trends & statistics for 2023; what you need to know," *Forbes*, 2023. Accessed: Oct. 16, 2024. [Online]. Available: https://www.forbes.com/sites/chuckbrooks/2023/03/05/cybersecurity-trends--statistics-for-2023-more-treachery-and-risk-ahead-as-attack-surface-and-hacker-capabilities-grow/?sh=6bb5ea0319db

[3] S. Jain, "160 cybersecurity statistics 2024," *Getastra*, 2024. Accessed: Dec. 12, 2024. [Online]. Available: https://www.getastra.com/blog/security-audit/cyber-security-statistics/

[4] M. Vielberth, F. Bohm, I. Fichtinger, and G. Pernul, "Security operations center: a systematic study and open challenges," *IEEE Access*, vol. 8, pp. 227756–227779, 2020, doi: 10.1109/ACCESS.2020.3045514.

[5] E. Agyepong, Y. Cherdantseva, P. Reinecke, and P. Burnap, "A systematic method for measuring the performance of a cyber security operations centre analyst," *Computers & Security*, vol. 124, Jan. 2023, doi: 10.1016/j.cose.2022.102959.

[6] E. Ahlm, "How to build and operate a modern security operations center," *Gartner Inc*, 2021. Accessed: Dec. 12, 2024. [Online]. Available: https://www.gartner.com/en/documents/4002259

[7] M. Zolanvari, M. A. Teixeira, L. Gupta, K. M. Khan, and R. Jain, "Machine learning-based network vulnerability analysis of industrial internet of things," *IEEE Internet of Things Journal*, vol. 6, no. 4, pp. 6822–6834, Aug. 2019, doi: 10.1109/JIOT.2019.2912022.

[8] T. Jafarian, M. Masdari, A. Ghaffari, and K. Majidzadeh, "A survey and classification of the security anomaly detection mechanisms in software defined networks," *Cluster Computing*, vol. 24, no. 2, pp. 1235–1253, Jun. 2021, doi: 10.1007/s10586-020-03184-1.
[9] CNN, "BSSN detects 44 million malware activities until May 2024 (in Indonesian: *BSSN deteksi 44 juta aktivitas malware hingga Mei 2024*)," *CNN Indonesia*, 2024. Accessed: Oct. 16, 2024. [Online]. Available: https://www.cnnindonesia.com/teknologi/20240516184354-185-1098626/bssn-deteksi-44-juta-aktivitas-malware-hingga-mei-2024
[10] A. H. Shah, D. Pasha, E. H. Zadeh, and S. Konur, "Automated log analysis and anomaly detection using machine learning," *Frontiers in Artificial Intelligence and Applications*, pp. 137-147, 2022, doi: 10.3233/FAIA220378.
[11] H. Han, Z. Yan, X. Jing, and W. Pedrycz, "Applications of sketches in network traffic measurement: a survey," *Information Fusion*, vol. 82, pp. 58–85, Jun. 2022, doi: 10.1016/j.inffus.2021.12.007.
[12] A. Diro, S. Kaisar, A. V. Vasilakos, A. Anwar, A. Nasirian, and G. Olani, "Anomaly detection for space information networks: a survey of challenges, techniques, and future directions," *Computers & Security*, vol. 139, Apr. 2024, doi: 10.1016/j.cose.2024.103705.
[13] Z. Zhao, H. Guo, and Y. Wang, "A multi-information fusion anomaly detection model based on convolutional neural networks and AutoEncoder," *Scientific Reports*, vol. 14, no. 1, Jul. 2024, doi: 10.1038/s41598-024-66760-0.
[14] A. B. Nassif, M. A. Talib, Q. Nasir, and F. M. Dakalbab, "Machine learning for anomaly detection: a systematic review," *IEEE Access*, vol. 9, pp. 78658–78700, 2021, doi: 10.1109/ACCESS.2021.3083060.
[15] W. A. Ali, K. N. Manasa, M. Bendechache, M. F. Aljunid, and P. Sandhya, "A review of current machine learning approaches for anomaly detection in network traffic," *Journal of Telecommunications and the Digital Economy*, vol. 8, no. 4, pp. 64–95, Dec. 2020, doi: 10.18080/jtde.v8n4.307.
[16] N. A. Azeez, T. O. Odeyemi, C. C. Isiekwene, and A. P. Abidoye, "Cyber attack detection in a global network using machine learning approach," *FUOYE Journal of Engineering and Technology*, vol. 8, no. 4, pp. 448-455, Dec. 2023, doi: 10.46792/fuoyejet.v8i4.1113.
[17] K. Veena, K. Meena, Y. Teekaraman, R. Kuppusamy, and A. Radhakrishnan, "C SVM classification and KNN techniques for cyber crime detection," *Wireless Communications and Mobile Computing*, vol. 2022, pp. 1–9, Jan. 2022, doi: 10.1155/2022/3640017.
[18] M. Vishwakarma and N. Kesswani, "A new two-phase intrusion detection system with Naïve Bayes machine learning for data classification and elliptic envelop method for anomaly detection," *Decision Analytics Journal*, vol. 7, Jun. 2023, doi: 10.1016/j.dajour.2023.100233.
[19] V. Nakhipova *et al.*, "Use of the naive Bayes classifier algorithm in machine learning for student performance prediction," *International Journal of Information and Education Technology*, vol. 14, no. 1, 2024, doi: 10.18178/ijiet.2024.14.1.2028.
[20] B. Mahesh, "Machine learning algorithms-a review," *International Journal of Science and Research (IJSR)*, vol. 9, no. 1, pp. 381–386, Jan. 2020, doi: 10.21275/ART20203995.
[21] M. D. Nath and T. Bhattasali, "Anomaly detection using machine learning approaches," *Azerbaijan Journal of High Performance Computing*, vol. 3, no. 2, pp. 196–206, Dec. 2020, doi: 10.32010/26166127.2020.3.2.196.206.
[22] A. Al Obaidli, D. Mansour, S. M. Abdulhamid, N. B. Halima, and A. Al-Ghushami, "Machine learning approach to anomaly detection attacks classification in IoT devices," in *2023 1st International Conference on Advanced Innovations in Smart Cities (ICAISC)*, Jan. 2023, pp. 1–6, doi: 10.1109/ICAISC56366.2023.10085349.
[23] P. Chhajer, M. Shah, and A. Kshirsagar, "The applications of artificial neural networks, support vector machines, and long-short term memory for stock market prediction," *Decision Analytics Journal*, vol. 2, Mar. 2022, doi: 10.1016/j.dajour.2021.100015.
[24] J. Zhu, S. He, P. He, J. Liu, and M. R. Lyu, "Loghub: a large collection of system log datasets for ai-driven log analytics," in *2023 IEEE 34th International Symposium on Software Reliability Engineering (ISSRE)*, Oct. 2023, pp. 355–366. doi: 10.1109/ISSRE59848.2023.00071.
[25] M. Akanle *et al.*, "Experimentations with openstack system logs and support vector machine for an anomaly detection model in a private cloud infrastructure," in *2020 International Conference on Artificial Intelligence, Big Data, Computing and Data Communication Systems (icABCD)*, Aug. 2020, pp. 1–7, doi: 10.1109/icABCD49160.2020.9183878.
[26] L. Xiang, "Application of an improved TF-IDF method in literary text classification," *Advances in Multimedia*, vol. 2022, pp. 1–10, May 2022, doi: 10.1155/2022/9285324.
[27] A. Addiga and S. Bagui, "Sentiment analysis on twitter data using term frequency-inverse document frequency," *Journal of Computer and Communications*, vol. 10, no. 08, pp. 117–128, 2022, doi: 10.4236/jcc.2022.108008.
[28] R. Chavan, G. Patil, V. Madle, and R. Joshi, "Curating stopwords in marathi: a TF-IDF approach for improved text analysis and information retrieval," *2024 IEEE 9th International Conference for Convergence in Technology (I2CT)*, Pune, India, 2024, pp. 1-6, doi: 10.1109/I2CT61223.2024.10544359.
[29] M. V. Anand, B. KiranBala, S. R. Srividhya, C. Kavitha, M. Younus, and M. H. Rahman, "Gaussian naïve Bayes algorithm: a reliable technique involved in the assortment of the segregation in cancer," *Mobile Information Systems*, vol. 2022, pp. 1–7, Jun. 2022, doi: 10.1155/2022/2436946.
[30] M. Ismail, N. Hassan, and S. S. Bafjaish, "Comparative analysis of Naive Bayesian techniques in health-related for classification task," *Journal of Soft Computing and Data Mining*, vol. 1, no. 2, pp. 1–10, 2020, doi: 10.30880/jscdm.2020.01.02.001.
[31] W. B. Zulfikar, A. R. Atmadja, and S. F. Pratama, "Sentiment analysis on social media against public policy using multinomial Naive Bayes," *Scientific Journal of Informatics*, vol. 10, no. 1, pp. 25–34, Jan. 2023, doi: 10.15294/sji.v10i1.39952.
[32] N. Naicker, T. Adeliyi, and J. Wing, "Linear support vector machines for prediction of student performance in school-based education," *Mathematical Problems in Engineering*, vol. 2020, pp. 1–7, Oct. 2020, doi: 10.1155/2020/4761468.
[33] M. Alida and M. Mustikasari, "Rupiah exchange prediction of US dollar using linear, polynomial, and radial basis function kernel in support vector regression," *Jurnal Online Informatika*, vol. 5, no. 1, pp. 53-60, 2020, doi: 10.15575/join.v5i1.537.
[34] I. S. Al-Mejibli, J. K. Alwan, and D. H. Abd, "The effect of gamma value on support vector machine performance with different kernels," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 10, no. 5, pp. 5497-5506, Oct. 2020, doi: 10.11591/ijece.v10i5.pp5497-5506.
[35] S. Uddin, I. Haque, H. Lu, M. A. Moni, and E. Gide, "Comparative performance analysis of k-nearest neighbour (KNN) algorithm and its different variants for disease prediction," *Scientific Reports*, vol. 12, no. 1, Apr. 2022, doi: 10.1038/s41598-022-10358-x.
[36] C. Kavitha, V. Mani, S. R. Srividhya, O. I. Khalaf, and C. A. T. Romero, "Early-stage alzheimer's disease prediction using machine learning models," *Frontiers in Public Health*, vol. 10, Mar. 2022, doi: 10.3389/fpubh.2022.853294.
[37] D. M. W. Powers, "Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation," *arXiv-Computer Science*, pp. 1-27, 2020.

[38] N. Yuvaraj *et al.*, "Nature-inspired-based approach for automated cyberbullying classification on multimedia social networking," *Mathematical Problems in Engineering*, vol. 2021, pp. 1–12, Feb. 2021, doi: 10.1155/2021/6644652.
[39] A. Tharwat, "Classification assessment methods," *Applied Computing and Informatics*, vol. 17, no. 1, pp. 168–192, Jan. 2021, doi: 10.1016/j.aci.2018.08.003.
[40] B. J. Erickson and F. Kitamura, "Magician's corner: 9. Performance metrics for machine learning models," *Radiology: Artificial Intelligence*, vol. 3, no. 3, May 2021, doi: 10.1148/ryai.2021200126.

BIOGRAPHIES OF AUTHORS

Muhammad Ajran Sapura currently a master's student in Universitas Sriwijaya. He received her undergraduate degree in the same university, majoring in computer science. He areas of interest include cyber security, SIEM, machine learning, and cyber security strategy. He can be contacted at email: akhiajran@outlook.co.id.

Deris Stiawan received the Ph.D. degree in computer engineering from Universiti Teknologi Malaysia, Malaysia. He is currently a Professor at Department of Computer Engineering, Faculty of Computer Science, Universitas Sriwijaya. His research interests include computer network, intrusion detection/prevention system, and heterogeneous network. He can be contacted at email: deris@unsri.ac.id.

Rahmat Budiarto received B.Sc. degree from Bandung Institute of Technology in 1986, M.Eng. and Dr.Eng. in computer science from Nagoya Institute of Technology in 1995 and 1998 respectively. Currently, he is a Full Professor at College of Computing and and Information, Albaha University, Saudi Arabia. He was the chairman of Network Security Working Group at Asia Pacific Advanced Networks (APAN) from January 2006 to August 2008, and the chairman of Fellowship Committee at Asia Pacific Advanced Networks (APAN) from January 2007 to August 2008. He was the deputy director and co-founder of National Advanced IPv6 (NAv6) Center, under the Ministry of Energy, Water, and Communication, Malaysia. His research interests include IPv6, network security, wireless sensor networks and intelligent systems. He can be contacted at email: rahmat@bu.edu.sa.