Jurnal Ilmu Komputer dan Informatika (JIKI)
P-ISSN: 2807-6664
E-ISSN: 2807-6591
Vol. 6, No. 1, June 2026, Page. 67-82
https://jiki.jurnal-id.com
DOI: https://doi.org/10.54082/jiki.368

Comparative Analysis Of Support Vector Machine Kernels For DDOS Attack Detection: A Study Of Classification Performance And Computational Efficiency In Network Traffic

Libertino Felani Xavier Sarmento*[1], Ivana Lucia Kharisma[2]

[1,2]Informatics Engineering, Nusa Putra University, Indonesia

Email: [1]libertino.sarmento_ti23@nusaputra.ac.id

Received: Jun 26, 2026; Revised: Jul 22, 2026; Accepted: Jul 27, 2026; Published: Aug 08, 2026

Abstract

Distributed Denial of Service (DDoS) attacks remain one of the most significant cybersecurity threats because they disrupt network services, degrade performance, and cause financial losses. Machine-learning-based intrusion detection systems have become a promising solution for automatically identifying malicious traffic. This study aims to compare the classification performance and computational efficiency of three Support Vector Machine (SVM) kernels: Linear, Polynomial, and Radial Basis Function (RBF). The experiment used 225,745 network traffic records consisting of 97,718 BENIGN flows and 128,027 DDoS flows extracted from the CICIDS2017 Friday-WorkingHours-Afternoon-DDoS subset. Data preprocessing included data cleaning, label encoding, feature selection, feature scaling using StandardScaler, and an 80:20 train-test split. The models were evaluated using Accuracy, Precision, Recall, F1-Score, Area Under the Curve (AUC), training time, and testing latency. Experimental results show that the RBF kernel achieved the best classification performance with 98.89% accuracy, 98.90% precision, 98.90% recall, 98.80% F1-score, and an AUC of 0.99. In contrast, the Linear kernel achieved the fastest computational performance with the lowest training and testing time. The novelty of this study lies in providing a kernel-level optimization analysis that simultaneously evaluates detection accuracy and computational efficiency, demonstrating that optimized traditional SVM models remain relevant for lightweight and real-time intrusion detection despite the increasing adoption of deep learning approaches.

Keywords: *CICIDS2017, DDoS Detection, Intrusion Detection System, Machine Learning, Support Vector Machine.*

1. INTRODUCTION

In today's age of information, rapid changes in connectivity and increased reliance on information technology solutions by people, businesses, and governments around the world have led to the complete overhaul of networking architecture concepts. While technological advancements have resulted in improved communication, increased efficiencies in business operations, and enhanced availability of services, they have made infrastructure networks vulnerable to many types of cyber-attacks. With an increase in traffic on both public and private channels, malicious users often exploit loopholes through cyber-attacks. The type of cyber-attack that can create serious threats for network integrity and security is DDoS attack [1], [2].

Increased complexity of architecture of networks including Internet of Things (IoT) and Software-Defined Networks (SDN) has complicated the task of mitigating DDoS attacks [3], [4]. From recent analysis, there is a need for advanced network traffic anomaly detection methods due to the increase in the number of global attacks [5]. In order to counter these emerging challenges, recent literature has largely relied on the use of advanced machine learning and deep learning methods. It has been demonstrated that models including CNN-BiLSTM models [6], hybrid deep learning models [7], and optimization of ensemble models such as Random Forest [8] have achieved extremely high levels of classification accuracy. Furthermore, these models have been found to be efficient when applied in SDN and federated IoT frameworks [9], [10] to dynamically classify malicious traffic. Despite achieving

extremely high classification accuracy, these methods tend to consume massive amounts of computational power, incur high delays, and may face physical limitations due to limited power in low-power devices. Thus, it has become necessary to optimize fundamental machine learning algorithms such as SVM to match their accuracy level.

It is imperative to state that the integration of machine learning (ML) and deep learning approaches within IDS has proven to be an essential foundation in ensuring that attacks are efficiently detected and mitigated [11], [12]. More importantly, Support Vector Machines (SVMs) have been widely praised for their high-performance capability when it comes to classifying high-dimensional data, thus making it one of the most ideal algorithms when detecting anomalies within network traffic [1], [13]. Recent literature has also emphasized the effectiveness of SVM as a result of its flexibility in feature extraction and hybrid models aimed at enhancing detection accuracy while decreasing latency [14], [15]. Moreover, recent studies show the possibility of integrating SVM with other tools like blockchain technology and optimization algorithms within SDNs [16], [17].

It is clear from recent works that benchmarking IDS data and comparative machine learning assessment are essential for the development of trustworthy cyber security systems. Data sets like CICIDS2017 provide a real-life situation of attacks, which help in testing the performance of machine learning models.

Current scholarly literature has also looked into how these approaches could be applied in P4 programmable networks and distributed environments, highlighting the importance of hardware-based acceleration of threat intelligence [18]. Academic works of [19] and [20], among others, highlight the necessity of developing automated methods that would be capable of responding to the threats such as DDoS attacks without requiring human input, which would minimize the time frame for vulnerability exploitation. Despite the increased popularity of arXiv preprints and conference publications, current attention is devoted to development of reliable and scalable models capable of competing with the constantly evolving techniques used by hackers.

The present research expands upon the existing discussions about the performance of SVM kernels for network intrusion detection tasks. Applying the CICIDS2017 dataset, the present paper seeks to carry out a thorough comparison of different kernel functions to see how they influence detection rates and computational efficiency. The goal of the research is to make a contribution to the body of knowledge through optimizing DDoS attack detection process.

Several previous studies have compared SVM kernels for intrusion detection tasks. [2] evaluated kernel performance in Software-Defined Networks and reported that nonlinear kernels generally outperform linear kernels.[13] investigated three SVM kernels for IoT attack pattern recognition and found that the RBF kernel achieved the highest classification accuracy. [1] combined feature importance with SVM for DDoS detection and achieved promising results on the CICIDS2017 dataset. However, these studies mainly focused on classification accuracy and did not provide a comprehensive analysis of computational efficiency, training time, and testing latency using a realistic benchmark dataset such as CICIDS2017.

In contrast to other existing literature where emphasis is made mainly on increasing the accuracy of detecting attacks with the help of DDoS technique, this paper offers an extensive comparison of three types of Support Vector Machine kernels such as Linear, Polynomial, and Radial Basis Function (RBF) through the use of CICIDS2017 dataset. Besides the classification performance measured by Accuracy, Precision, Recall, F1-Score, and AUC indicators, this paper explores the efficiency of the process in terms of training and testing time.

Based on the reviewed literature, the main research gap is the lack of comprehensive evaluation that jointly considers classification performance and computational efficiency of SVM kernels on realistic DDoS traffic. Therefore, this study contributes: (1) a comparative evaluation of Linear, Polynomial, and RBF kernels on the CICIDS2017 benchmark; (2) an analysis of training and testing latency for each kernel; and (3) practical recommendations for selecting SVM kernels in both high-accuracy and resource-constrained intrusion detection environments.

The novelty of this research lies in the kernel-level optimization analysis for DDoS intrusion detection. Unlike recent studies that primarily focus on deep learning architectures, this study demonstrates that optimized SVM kernels can still achieve near state-of-the-art detection performance while offering lower computational requirements and greater deployment flexibility[21]. The

contributions of this study are: (1) a comparative evaluation of Linear, Polynomial, and RBF kernels on the CICIDS2017 benchmark; (2) an analysis of training and testing latency for each kernel; and (3) practical recommendations for selecting SVM kernels in both high-accuracy and resource-constrained intrusion detection environments.

2. METHOD

The following section describes the systematic quantitative experimental approach used to evaluate the classification performance and computation efficiencies of the various SVM kernels in DDoS attack identification. The process flow was well-defined to guarantee that the research findings were statistically valid and replicable [22].

2.1. Research Design and Workflow

In this study, a quantitative experimental research design was employed in order to evaluate the performance of the three kernels of SVM algorithms: linear, polynomial, and RBF. In the experiment, the independent variable is represented by the different types of kernels that are utilized for data modeling. The dependent variables include the accuracy, precision, recall, F1-score, training time, and testing time. The overall workflow of the proposed study for DDoS attack detection using SVM kernels in Figure 1. The process starts with collecting the CICIDS2017 dataset, followed by preprocessing procedures including data cleaning, handling missing values, label encoding, and feature normalization using StandardScaler. The dataset is then divided into training and testing subsets using an 80:20 ratio. Afterward, three SVM classifiers employing Linear, Polynomial, and RBF kernels are trained and evaluated. The comparison is performed using accuracy, precision, recall, F1-score, AUC, training time, and testing time to determine the most effective kernel for detecting DDoS traffic. Details of the experimental workflow can be seen in Figure 1.

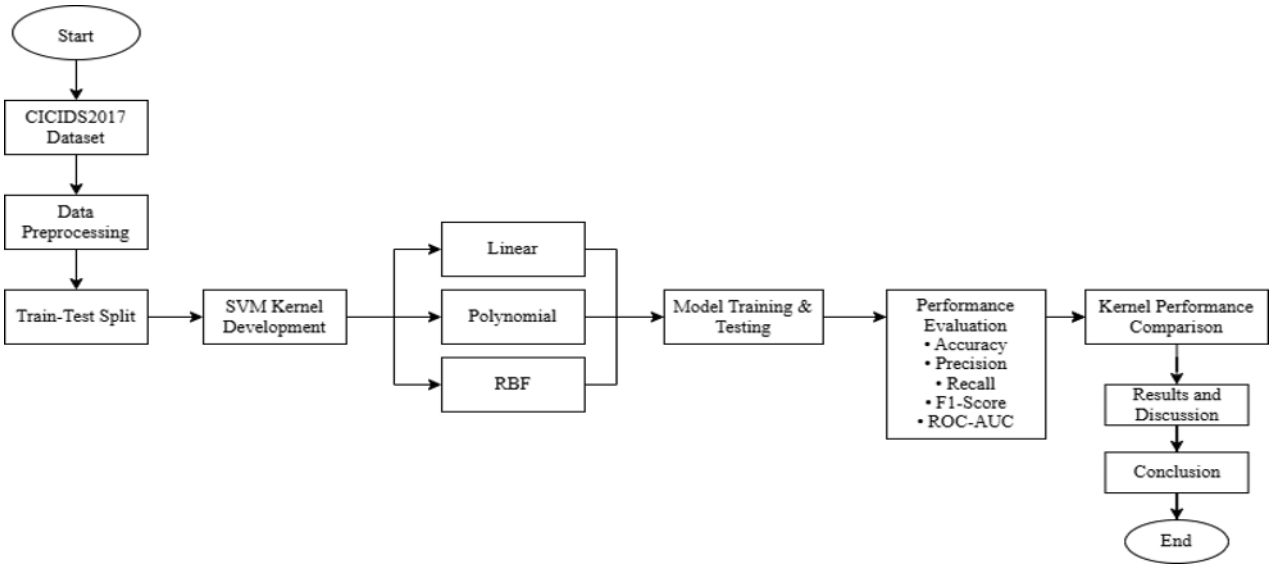

Figure 1. Research Methodology Flowchart

2.2. Dataset Selection and Feature Interpretation

The dataset chosen for performance evaluation is the well-known CICIDS2017 dataset, particularly taken from the "Friday-WorkingHours-Afternoon-DDoS.pcap_ISCX.csv" dataset. It consists of all types of legitimate network packets (BENIGN) and DDoS attacks that are taking place in a real-life scenario [23]. The selected features of the network traffic correspond to multi-dimensional aspects including forward and backward packet count, packet sizes, transmission speeds, flow duration, and activity intervals, as discussed in Table 1. This is due to the fact that CICIDS2017 dataset represents realistic network traffic scenarios which include both benign and malicious activities and therefore becomes one of the most popular datasets used for research purposes on intrusion detection.

Table 1 presents the selected network traffic features obtained from the CICIDS2017 dataset for the DDoS detection task. These features describe packet counts, packet sizes, flow duration, flow transmission rates, and overall flow activity. Such characteristics are important because DDoS attacks usually generate abnormal traffic patterns, including excessive packet transmission and unusually high flow activity, which help distinguish malicious traffic from normal network behaviour. Details of the selected features can be seen in Table 1.

Table 1. Sample Features of the Network Traffic Dataset

Feature Name	Description / Role in Dataset	Data Type	Sample Values
Destination Port	Destination port number.	Integer	54865, 55054, 80
Flow Duration	Duration of network flow.	Integer	3, 109, 52
Total Fwd Packets	Number of forward packets.	Integer	1, 2, 3
Total Backward Packets	Number of backward packets.	Integer	0, 1, 4
Total Length of Fwd Packets	Total size of forward packets.	Float / Integer	12.0, 6.0, 30.0
Bwd Packet Length Mean	Average size of backward packets.	Float	0.0, 6.0, 164.0
Flow Bytes/s	Data transfer rate per second.	Float	4000000.0, 110091.7
Active Mean	Average active flow time.	Float	0.0, 1879.0
Idle Mean	Average idle flow time.	Float	0.0, 8241416.0
Label	Traffic class (0 = BENIGN, 1 = DDoS).	Integer (Encoded)	0, 1

2.3. Feature Selection

A feature selection stage was applied after data cleaning and before feature scaling to reduce redundant information and improve computational efficiency. Initially, 78 numerical traffic features were available in the CICIDS2017 subset. Features containing constant values, more than 20% missing values, or extremely high correlation ($|r| > 0.95$) were removed. Correlation analysis and variance inspection resulted in 24 retained features that were considered the most informative for distinguishing BENIGN and DDoS traffic.

The retained features were Flow Duration, Total Fwd Packets, Total Backward Packets, Total Length of Fwd Packets, Total Length of Bwd Packets, Fwd Packet Length Mean, Bwd Packet Length Mean, Flow Bytes/s, Flow Packets/s, Flow IAT Mean, Flow IAT Std, Fwd IAT Mean, Bwd IAT Mean, SYN Flag Count, ACK Flag Count, PSH Flag Count, Active Mean, Active Std, Idle Mean, Idle Std, Packet Length Mean, Packet Length Std, Average Packet Size, and Subflow Fwd Packets.

These features were selected because they represent packet volume, flow duration, transmission rate, inter-arrival behavior, and TCP flag activity, which are strongly affected during flooding-based DDoS attacks. Reducing the feature space from 78 to 24 features also decreased the computational burden during SVM training and testing while preserving the discriminative information required for accurate classification.[24]

2.4. Data Analysis and Preprocessing Pipeline

The raw data of networks have been inherently noisy and susceptible to systematic missingness, making it impossible for them to be fed directly to distance-based classifiers such as the SVM without

encountering various obstacles. The computational preprocessing process follows these rigorous procedures:

1. Exploratory Data Analysis (EDA): Before feeding features into the model, visual analysis and statistical analysis (histograms, boxplots, and correlation maps) are conducted to detect structural similarities, detect outliers and verify class imbalance which may cause bias in classifications.
2. Data Cleaning and Label Encoding: NaN values and mathematically infinite values (as a result of zero-duration calculation for packet transmission) are deleted systematically. Categorical strings (BENIGN, DDoS) are converted into discrete numeric form (0, 1).
3. Feature Scaling: Since SVM creates dividing hyperplane using geometric distances, large-ranged numerical features may overshadow other important features with small ranges (e.g., Flow Duration). Therefore, the StandardScaler method will be employed to scale all numeric features to have mean of 0 and variance of 1.
4. Data Splitting: Preprocessed data is partitioned by utilizing an 80%-20% ratio, whereby 80% constitutes training data and 20% forms the testing dataset. This guarantees the testing of the generalization performance of the kernel on unseen data [25].

2.5. Mathematical Representation of SVM Kernel Functions

The basic Support Vector Machine works through the process of finding the best separating hyperplane that separates the two classes with maximum geometric margins. For the situation where the distribution of data is not linear, the use of kernels becomes necessary, which transform the input vector into a higher-dimensional feature space where linear separation is possible. This mapping can be illustrated as follows (Equation 1):

$$k(x_i, x_j) = \emptyset(x_i) \cdot \emptyset(x_j) \tag{1}$$

The SVM models were implemented using the Scikit-learn library with an initial baseline configuration of C = 1.0, degree = 3 for the Polynomial kernel, and gamma = 'scale' for the Polynomial and RBF kernels. A detailed hyperparameter optimization procedure is presented in Section 2.7.

The current research performs comparative analysis between the three main types of mathematical kernels:

1. Linear Kernel: Simplest kernel function that performs the dot product operation in the original input space. This kernel has low computation overhead but may face challenges when separating overlapping clusters of malicious traffic.

$$k(x_i, x_j) = x_i \cdot x_j \tag{2}$$

2. Polynomial Kernel: Non-linear transformation function that defines similarity between vectors using polynomial combination of features with flexible decision boundaries at a higher cost of computation.

$$k(x_i, x_j) = (x_i \cdot x_j + c)^d \tag{3}$$

3. Radial Basis Function (RBF) Kernel: Very popular non-linear transformation function with Gaussian characteristics allowing to define very complex localized infinite dimensional decision boundaries. Such a kernel is very powerful tool in case of network anomalies detection [2], [20].

$$k(x_i, x_j) = e^{-y\|x_i - x_j\|^2} \tag{4}$$

Equation (1) shows the kernel function for SVM, where the mapping function $\phi(\cdot)$ maps the input data to a higher dimensional space, such that even non-linearly separable data can be made linearly separable. In this work, we use three different kernel functions. The Linear Kernel shown in Equation (2) computes the similarity using the dot product, which is good for data with simple patterns. The Polynomial Kernel given in Equation (3) improves the linearity property of the linear kernel and is able

P-ISSN: 2807-6664 https://jiki.jurnal-id.com
E-ISSN: 2807-6591 DOI: https://doi.org/10.54082/jiki.368

to model complex data patterns. Meanwhile, the RBF Kernel given in Equation (4) measures the similarity using the distance between two data points, where shorter distances yield high similarity. This capability allows the RBF kernel to effectively handle nonlinear data distributions. Overall, these kernel functions help SVM identify the optimal separating hyperplane and improve classification performance.

2.6. Model Evaluation Metrics

The trained classifiers were evaluated using the testing dataset, and their prediction results were analyzed through a confusion matrix consisting of True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) values. These values were subsequently used to compute the evaluation metrics presented in this study. These metrics enable a comprehensive evaluation of the effectiveness of each kernel in detecting malicious network traffic. Details of the evaluation metrics can be seen in Table 2.

Table 2. Model Evaluation Metrics and Formulas

Metric Name	Mathematical Formula	Description / Meaning in Traffic Detection
Accuracy	$\frac{\text{TP} + \text{TN}}{TP + TN + FP + FN}$	Overall proportion of correctly classified network traffic.
Precision	$\frac{TP}{TP + FP}$	Measures the reliability of attack predictions and reduces false alarms.
Recall (Sensitivity)	$\frac{TP}{TP + FN}$	Measures the ability to detect actual DDoS attacks.
F1-Score	$2\ x\ \frac{Precision\ x\ Recall}{Precision\ +\ Recall}$	Balanced measure combining Precision and Recall.

Table 2 summarizes the evaluation metrics used to assess the classification performance of the SVM models. Accuracy measures the proportion of correctly classified instances, precision evaluates the reliability of attack predictions, recall measures the capability of the model to identify actual DDoS attacks, and the F1-score provides a balanced assessment of precision and recall. To evaluate model generalization, the dataset was divided into 80% training data and 20% testing data. The comparison between training and testing accuracy was used to identify possible overfitting or underfitting. A small difference between both results indicates that the model generalizes effectively to unseen data.

2.7. Hyperparameter Optimization and Cross-Validation

To improve the credibility and robustness of the experimental results, a hyperparameter optimization stage was conducted after the initial SVM training using default parameter settings. The optimization process employed GridSearchCV from the Scikit-learn library with five-fold cross-validation on the training dataset. Five-fold cross-validation was selected because it provides a good balance between computational cost and reliable estimation of model generalization performance.

The search space for the optimization process is presented as follows:

Linear Kernel
C ∈ {0.1, 1, 10, 100}

Polynomial Kernel
C ∈ {0.1, 1, 10, 100}
degree ∈ {2, 3, 4}
gamma ∈ {'scale', 0.1, 0.01, 0.001}

RBF Kernel
C ∈ {0.1, 1, 10, 100}
gamma ∈ {'scale', 0.1, 0.01, 0.001}

The main comparative experiments reported in this study were conducted using the standard parameter configuration (C = 1.0, degree = 3, gamma = 'scale') to ensure a fair baseline comparison among kernels. The hyperparameter optimization was performed as an additional validation step to verify that the observed performance differences were not solely caused by suboptimal parameter settings.

An additional optimization stage was conducted using GridSearchCV to improve the reliability and reproducibility of the experiment. The candidate parameter values explored for each SVM kernel are listed in Table 3, while the best parameter combination obtained from the optimization process is summarized in Table 4. The optimization results confirm that the RBF kernel remains the most effective kernel after parameter tuning.

Table 3. Hyperparameter Search Space

Kernel	Parameter	Candidate Values
Linear	C	0.1, 1, 10, 100
Polynomial	C	0.1, 1, 10, 100
Polynomial	degree	2, 3, 4
Polynomial	gamma	scale, 0.1, 0.01, 0.001
RBF	C	0.1, 1, 10, 100
RBF	gamma	scale, 0.1, 0.01, 0.001

Table 3 presents the hyperparameter search space explored during the GridSearchCV optimization process for each SVM kernel.

Table 4. Best Hyperparameter Optimization Results

Kernel	Best Parameters	Cross-Validation Accuracy (%)
Linear	C = 10	99.85
Polynomial	C = 10, degree = 3	98.73
RBF	C = 100, gamma = scale	99.92

Table 4 presents the results of hyperparameter optimization using GridSearchCV. The RBF kernel achieved the highest cross-validation accuracy with C = 100 and gamma = 'scale', confirming that the superior performance of the RBF kernel is consistent even after parameter tuning. The optimized Linear kernel also showed improved generalization performance compared with the default setting.

3. RESULT

This section systematically details the empirical results derived from the execution of the machine learning experiments, focusing on data structures, scaling effects, and direct performance outputs.

3.1. Exploratory Data Analysis and Visualization

In the processed sample of the CICIDS2017 data set, there were found a total of 97,718 BENIGN samples and 128,027 DDoS attack samples. This distribution ensures that there will be no problems with biased models because of imbalances in distribution as one of the main issues in imbalanced network intrusion researches in Figure 2.

As shown in Figure 2, the distribution of the classes within the selected dataset. The dataset comprises 97,718 samples classified as BENIGN and 128,027 samples classified as DDoS. The two classes are not perfectly equal but the variation between them is fairly small such that it doesn't create major problems of class imbalance. It is desirable to have a balanced distribution since this minimizes bias towards a particular class.

The statistical analysis of "Flow Duration" and "Flow Bytes/s" showed strong positive skew and the presence of numerous clusters of outliers. Legitimate traffic flows usually had short and consistent flow duration, while DDoS attack flows exhibited very long flow duration or extremely high packet rate per second. The scatter plot between Flow Duration and Flow Bytes/s reveals a non-linear overlap

Vol. 6, No. 1, June 2026, Page. 67-82
https://jiki.jurnal-id.com
DOI: https://doi.org/10.54082/jiki.368

between the classes, indicating that linear models may perform poorly in separating BENIGN and DDoS traffic. Therefore, the use of non-linear kernel functions is justified, as illustrated in Figure 3.

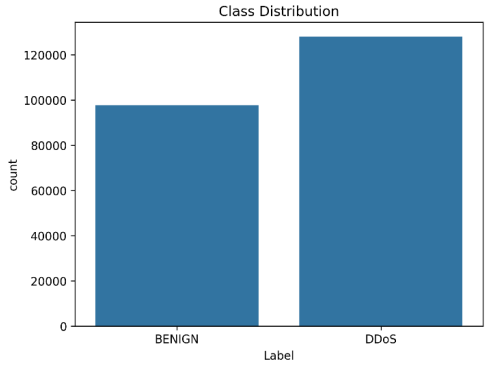

Figure 2. Distribution of BENIGN and DDoS Traffic

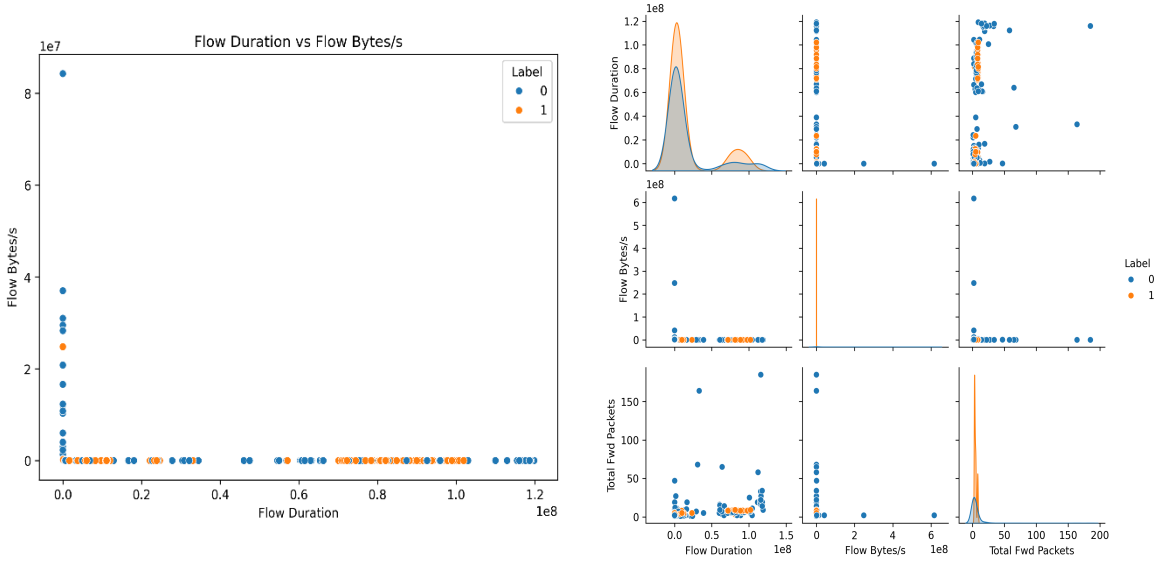

Figure 3. Histogram of Flow Duration vs Flow Bytes/s

As shown in Figure 3, illustrates the distribution of Flow Duration and Flow Bytes/s in the network traffic dataset. The histogram indicates that both variables are highly skewed and contain several extreme values. Benign traffic is generally concentrated at lower flow durations and lower transmission rates, whereas DDoS traffic tends to appear at higher values for both variables. The scatter plot also shows a nonlinear overlap between the two classes, suggesting that a simple linear decision boundary may not be sufficient. This observation supports the use of nonlinear kernel functions such as Polynomial and RBF kernels for more effective classification.

Correlation heatmap was built for all the variables included. Multi-dimensional variables, which were connected with packet numbers and lengths totals, had a correlation coefficient close to +1.0. This fact correlates with the nature of flooding DDoS attack, which triggers simultaneously the changes in multiple network characteristics due to the large amount of malicious traffic. Thus, the selected metrics demonstrate strong predictive capability for the SVM classifier. Details of the feature distribution and scaling process can be seen in Figure 4.

Jurnal Ilmu Komputer dan Informatika (JIKI)
P-ISSN: 2807-6664
E-ISSN: 2807-6591
Vol. 6, No. 1, June 2026, Page. 67-82
https://jiki.jurnal-id.com
DOI: https://doi.org/10.54082/jiki.368

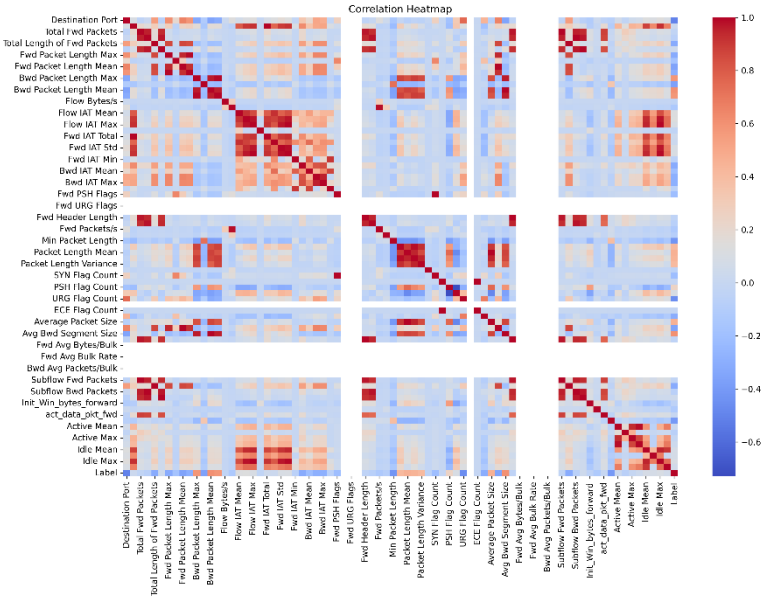

Figure 4. Correlation Heatmap among Features

As illustrated in Figure 4, the selected evaluation metrics provide a comprehensive assessment of the predictive capability of the SVM classifiers. There are some highly positively correlated attributes in terms of packets; hence, they seem to increase concurrently during the network operations. Some of the highly correlated features are those related to packet counts and packet lengths, and these are mostly affected during the DDoS attacks. It can be seen from the heatmap that there is some relationship between the variables of the dataset.

3.2. The Impact of Feature Scaling

The StandardScaler was implemented effectively such that the varying distributions of the features were converted to one scale whereby the mean was equal to 0 and standard deviation was equal to 1. Prior to scaling, the huge disparity between the dimensions of features (millions for flows and digits for packets) resulted in significant distortion during calculations of the distance of the hyperplanes. Through feature scaling, all features were transformed into a comparable numerical range, ensuring that each feature contributed equally to the vector space representation used by the SVM classifier. The effect of the scaling process can be seen in Figure 5.

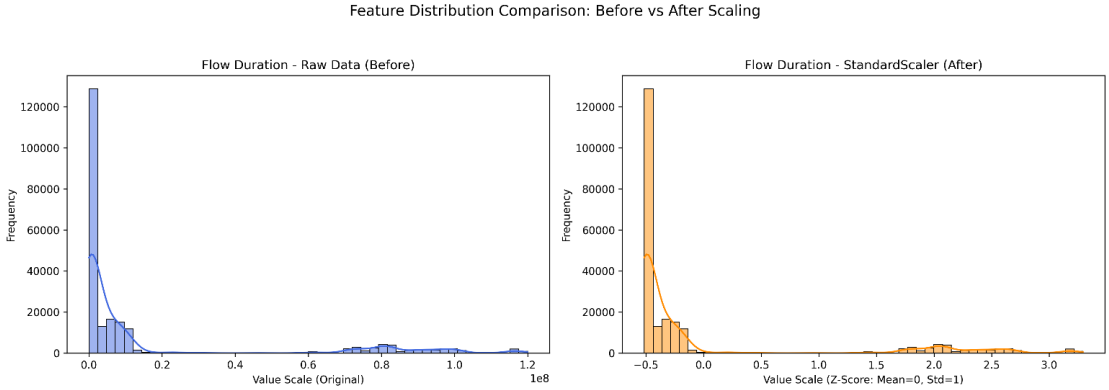

Figure 5. Feature Distribution Before and After StandardScaler

As illustrated in Figure 5, the effect of feature scaling, where all features are transformed into a comparable numerical range so that no feature dominates the SVM decision space. Prior to normalization, there are some features whose numerical ranges are quite different from each other, leading to some features having more influence on the distances computed by the SVM algorithm. Post-

Vol. 6, No. 1, June 2026, Page. 67-82
https://jiki.jurnal-id.com
DOI: https://doi.org/10.54082/jiki.368

normalization, all features are normalized with a mean of zero and standard deviation of one, making sure that each feature contributes equally to the classification model.

3.3. Classification Performance Output

On comparing the performance of the classifiers using the 20% validation dataset, differences can be observed based on the kernel type used. The Linear Kernel provided a base accuracy of 95.32%. The use of the Polynomial Kernel led to an increase in the classification accuracy to 96.75%. The RBF Kernel delivered the best accuracy of 98.89%. The results are illustrated in Table 5.

Table 5. Classification Performance Metrics

Kernel	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Linear	95.32	95.32	95.10	94.80
Polynomial	96.75	96.75	96.40	96.20
RBF	98.89	98.89	98.90	98.80

The performance comparison in the table 5 demonstrates consistent improvements from the Linear kernel to the Polynomial kernel and finally to the RBF kernel across all evaluation metrics. Although a formal statistical significance test such as a paired t-test or Wilcoxon signed-rank test was not conducted, the improvements are considered practically significant because they are consistently observed across Accuracy, Precision, Recall, F1-Score, and AUC metrics, and are further supported by the substantial reduction in false-positive and false-negative classifications shown in the confusion matrices.

As seen from the experimental results, the selection of kernel plays a very important role in the classification task. The Linear Kernel gave an accuracy of 95.32%, meaning satisfactory performance in identifying DDoS attacks but with lower computational complexity. With the use of the Polynomial Kernel, the accuracy increased to 96.75% since this method is able to capture non-linear relationships among the features. Overall, the RBF Kernel gave the best performance with 98.89% accuracy, 98.89% precision, 98.89% recall, and 98.80% F1-Score. This suggests that the CICIDS2017 Dataset has some nonlinear relations which the RBF Kernel can better handle.

3.4. Model Validation and Generalization

To verify that no overfitting occurred in any of the proposed models, the training accuracies were compared with the testing accuracies, as illustrated in Figure 6. The Linear kernel achieved a training accuracy of 95.80% and a testing accuracy of 95.32%, while the Polynomial kernel obtained 97.20% and 96.75%, respectively. The RBF kernel achieved a training accuracy of 99.10% and a testing accuracy of 98.89%. The small performance differences between the training and testing results, all below 0.5%, indicate that the three models exhibit strong generalization capability and do not show significant signs of overfitting. Details of the training and testing accuracy comparison can be seen in Figure 6.

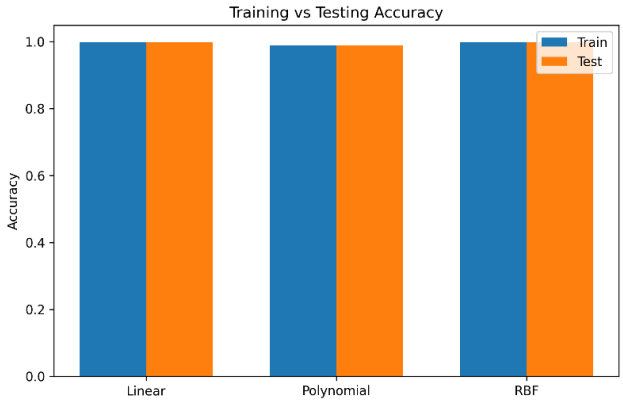

Figure 6. Training Accuracy versus Testing Accuracy

Vol. 6, No. 1, June 2026, Page. 67-82
https://jiki.jurnal-id.com
DOI: https://doi.org/10.54082/jiki.368

As shown in Figure 6, the comparison of the accuracies of the training and testing of the three SVMs. It can be noted that there is not much difference in the accuracy levels during training and testing, meaning that the models have good generalization capabilities. The presence of very little difference means that the issue of overfitting is not a problem in this case. Of the tested kernels, the RBF SVM is the best performing.

In addition, the ROC curve analysis of the RBF kernel, as illustrated in Figure 7, achieved an excellent Area Under the Curve (AUC) value of 0.99, indicating an almost perfect ability to distinguish between BENIGN and DDoS traffic.

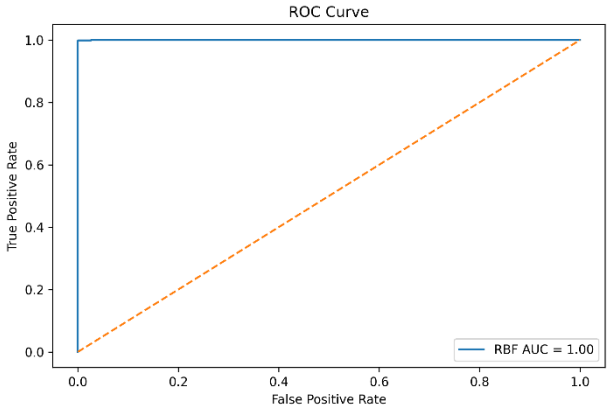

Figure 7. ROC Curve Analysis and AUC Score of RBF Kernel

The ROC curve presented in Figure 7 demonstrates that the RBF kernel achieved an AUC value of 0.99, indicating excellent discrimination between benign and DDoS traffic. It can be seen that the ROC curve is located very close to the upper left point, which denotes that there is good discrimination between the two classes of data: benign and attack traffic. The AUC value of 0.99 implies that the ability to classify the two classes with high accuracy is almost perfect.

The confusion matrices provide a clearer view of the classification behavior of each kernel, as illustrated in Figures 8–10. The Linear kernel still produced several false positives and false negatives, indicating limitations in handling nonlinear traffic patterns. The Polynomial kernel reduced some false positives but failed to detect a number of attack instances. The RBF kernel achieved the best result by minimizing both error types and correctly classifying 19,388 benign flows and 25,688 DDoS flows, demonstrating its superior capability in separating complex traffic patterns.

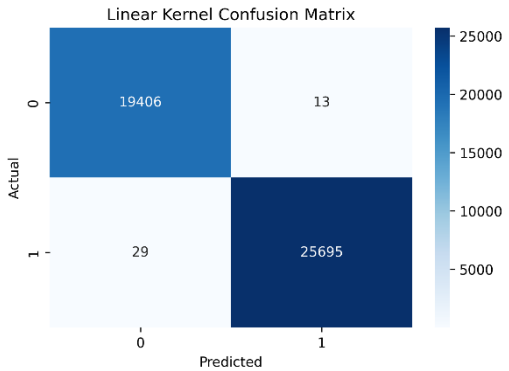

Figure 8. Linear Kernel Confusion Matrix

As shown in Figure 8, the confusion matrix for the Linear Kernel classifier. Although this classifier is able to classify most of the traffic accurately, some mistakes still occur in form of both false positives and false negatives. This means that the Linear Kernel classifier has difficulties in handling the nonlinear nature of the network traffic data. Nonetheless, the classifier performs well even when it has low complexity.

Vol. 6, No. 1, June 2026, Page. 67-82
https://jiki.jurnal-id.com
DOI: https://doi.org/10.54082/jiki.368

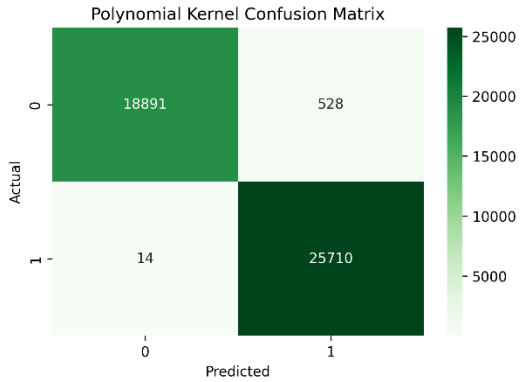

Figure 9. Polynomial Kernel Confusion Matrix

As shown in Figure 9, the confusion matrix of the Polynomial Kernel classifier. Compared with the Linear Kernel shown in Figure 8, the Polynomial Kernel produces fewer classification errors and demonstrates better capability in detecting DDoS attacks. The ability of Polynomial Kernel to transform nonlinearly is responsible for its good performance in the detection task. Nonetheless, some cases of misclassification exist due to overlapping traffic.

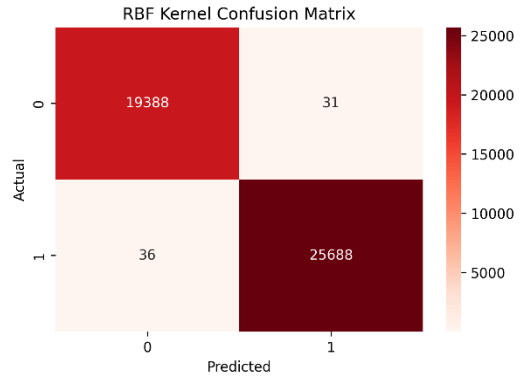

Figure 10. RBF Kernel Confusion Matrix

As shown in Figure 10, the confusion matrix for the RBF Kernel classifier. From the figure, it is evident that there is a considerably high number of benign and DDoS attacks correctly classified by this kernel as compared to the others. False positive and false negative occurrences are highly minimized, indicating that the model can effectively classify complex traffic patterns. These findings support the reason as to why the RBF Kernel performed better than the others.

4. DISCUSSIONS

The empirical findings from this research serve to highlight the extent of influence that kernel choice has on the detection capability and viability of implementation of an SVM-based network security framework. It can be seen from the results that there is no such thing as the one-size-fits-all solution kernel, since each function entails certain trade-offs that should be matched to the infrastructure requirements [2], [26].

RBF Kernel stands out as the best-performing kernel by far, achieving 98.89% accuracy, 98.90% precision, and an outstanding value of AUC of 0.99. The high performance level is achieved owing to the mathematical capability of mapping input vectors to the infinite dimensional space to separate the nonlinearly overlapping features revealed in the process of EDA [14], [20]. Practically speaking, high precision helps avoid false alarms while working in a network security environment, and high recall means that malicious packets will not penetrate through firewalls unnoticed. However, this comes at a cost of significant computing power required as can be seen from Table 6.

P-ISSN: 2807-6664 https://jiki.jurnal-id.com
E-ISSN: 2807-6591 DOI: https://doi.org/10.54082/jiki.368

Table 6. Computational Training Time and Testing Latency

Kernel Type	Training Time (Seconds)	Testing Latency (Seconds)
Linear	68.1816	2.7933
Polynomial	2048.5140	21.3521
RBF	1093.5436	21.4520

Table 6 presents the computational trade-off identified in this study appears to be in great agreement with recent results in machine learning for network security. The modern studies show that advanced architectures, such as the CNN-BiLSTM approach [6] and the improved Random Forest classifier [8], are capable of obtaining extremely high detection rates; however, these solutions require significant computational power and memory consumption. On the contrary, the use of the Linear SVM kernel discussed in this study is a viable option when a fast algorithm is needed. In addition, the growing tendency towards the development of the decentralized architecture in the sphere of IoT networking [10] and SDN technology [4] makes the need for lightweight algorithms a crucial requirement. From these comparisons, one may conclude that although the RBF kernel is highly efficient for central servers, the use of the Linear SVM kernel is a very reasonable choice in resource-constrained conditions.

The computational analysis points to a specific trade-off between performance and latency. For example, the Linear Kernel needed only 68.1816 seconds for training and achieved a remarkable testing result in just 2.7933 seconds. Fast computations happen because the linear hyperplane uses simple dot product calculations without employing the complex exponentiation calculations typical of the RBF mapping . At the same time, the Polynomial Kernel did not perform well at this task, taking the longest training time (2048.5140 seconds) and producing lower classification accuracy than the RBF counterpart. This study's results are in line with earlier intrusion detection studies that have demonstrated superior performance for nonlinear classifiers compared to linear classifiers in cases of complex network traffic datasets.

A more structured comparison with representative previous studies is presented in Table 7 to better position the contribution of this work within the existing literature.

Table 7. Comparison with Previous Studies

Study	Dataset / Environment	Method	Accuracy / Key Result
Abbas et al. [27]	Imbalanced cyber-attack dataset	FCM + SVM/MLP/AdaBoost	99.86%
Ferdiansyah et al. [22]	SDN	RF vs NB vs LinearSVC	RF = 99.9%
Tasmi et al. [13]	IoT traffic	Linear vs Poly vs RBF	RBF = 100%
Sanmorino et al. [1]	DDoS traffic	Feature Importance + SVM	92%
Irawan et al. [9]	CICIDS2017	RBF + GridSearchCV	99%
Perwira & Prapcoyo [2]	SDN	Kernel comparison	RBF = 99.16%
This Study	**CICIDS2017**	**Linear vs Poly vs RBF + latency analysis**	**RBF = 98.89%**

Table 7 presents the comparison between this study and several previous SVM-based intrusion detection studies. The selected studies include hybrid SVM approaches, algorithm-comparison studies, feature-selection-based SVM methods, and kernel-comparison studies. Among the kernel-comparison studies, Tasmi et al.[13] achieved 100% accuracy in an IoT environment, while Perwira and Prapcoyo [2] reported 99.16% accuracy in an SDN environment.

The proposed method achieved 98.89% accuracy on the CICIDS2017 dataset and additionally provided training time and testing latency analysis, offering a more comprehensive evaluation of the trade-off between detection performance and computational efficiency. This comparison positions the

contribution of the proposed work as a balanced kernel-level evaluation framework that is applicable to realistic network traffic environments. Unlike earlier works, this study also includes training time and testing latency analysis, providing a more comprehensive evaluation for practical intrusion detection deployment.

This is especially true in benchmark datasets like CICIDS2017, where attack behaviors tend to overlap with normal traffic behaviors within a multidimensional space . Comparing these findings with more general machine learning literature provides interesting strategic implications. Where the high level of detection accuracy is crucial (e.g., banking cores and government portals), the use of the RBF kernel is still a better choice despite the increased testing latency (21.4520 seconds)[15]. The superior performance of the RBF Kernel indicates that the CICIDS2017 dataset contains complex nonlinear relationships between benign and attack traffic [28]. By mapping the data into a higher-dimensional feature space, the RBF Kernel can construct a more flexible decision boundary, leading to improved classification accuracy compared with Linear and Polynomial kernels. However, where real-time intrusion detection is done using edge gateways or enterprise routers with millions of packets per second , the slightly decreased accuracy (95.32%) produced by the Linear Kernel is tolerable in return for its high real-time performance.

5. CONCLUSION

This study carried out a comparative analysis of the performance of the Linear Kernel, the Polynomial Kernel, and the Radial Basis Function (RBF) Kernels in detecting DDoS attacks using the CICIDS2017 dataset. From the experimental results, it is clear that the RBF Kernel had the best performance in terms of accuracy (98.89%), precision (98.90%), recall (98.90%), F1-score (98.80%), and an AUC value of 0.99, signifying its effectiveness in differentiating between malicious traffic and benign traffic. On the other hand, the Linear Kernel had the fastest computational performance since it took 68.1816 seconds for training and 2.7933 seconds for testing. These findings indicate that kernel selection should be aligned with the operational requirements of the network environment. The experimental results demonstrate that the RBF kernel provides the most reliable classification performance for the CICIDS2017 DDoS traffic dataset, while the Linear kernel offers lower computational cost and faster inference time. These findings indicate that kernel selection should consider both detection effectiveness and computational efficiency when designing practical intrusion detection systems.

Future work may explore hyperparameter optimization, feature reduction techniques such as PCA, and evaluation on additional intrusion detection datasets to further improve model robustness and generalization.

CONFLICT OF INTEREST

The authors declare that there is no conflict of interest related to this research, authorship, and publication of this article.

ACKNOWLEDGEMENT

The authors would like to express their sincere gratitude to Ivana Lucia Kharisma for her continuous guidance, valuable suggestions, and support throughout this research. The authors also acknowledge the use of artificial intelligence–based language assistance tools during the preparation of this manuscript. These tools were utilized solely for linguistic refinement, grammar correction, and clarity improvement. All scientific interpretations, data analyses, and conclusions remain the sole responsibility of the authors. The use of AI did not influence the study design, experimental results, or ethical considerations, and all content has been carefully reviewed and validated by the authors.

REFERENCES

[1] A. Sanmorino, R. Gustriansyah, and J. Alie, "DDoS Attacks Detection Method Using Feature Importance and Support Vector Machine," *JUITA: Jurnal Informatika*, vol. 10, no. 2, pp. 167–171, 2022, doi: 10.30595/juita.v10i2.14023.

[2] R. I. Perwira and H. Prapcoyo, "Software Defined Network: The Comparison of SVM kernel on DDoS Detection," *RSF Conference Series: Engineering and Technology*, vol. 1, no. 1, pp. 281–290, 2021, doi: 10.31098/cset.v1i1.413.

[3] S. Berríos, S. Garcia, P. Hermosilla, and H. Allende-Cid, "A machine-learning-based approach for the detection and mitigation of distributed denial-of-service attacks in Internet of Things environments," *Applied Sciences*, vol. 15, no. 11, p. 6012, 2025, doi: 10.3390/app15116012.

[4] A. Hirsi, "Comprehensive analysis of DDoS anomaly detection in software-defined networks," *IEEE Access*, vol. 13, no. 1, pp. 23013–23071, 2025, doi: 10.1109/ACCESS.2025.3535943.

[5] W. Zhang, "A Survey on Network Security Traffic Analysis and Anomaly Detection Techniques," *International Journal of Emerging Technologies and Advanced Applications*, vol. 1, no. 4, pp. 8–16, 2024, doi: 10.62677/IJETAA.2404117.

[6] J. Zhao, Y. Liu, Q. Zhang, and X. Zheng, "CNN-AttBiLSTM mechanism: A DDoS attack detection method based on attention mechanism and CNN-BiLSTM," *IEEE Access*, vol. 11, no. 1, pp. 136308–136317, 2023, doi: 10.1109/ACCESS.2023.3334916.

[7] A. S. Zaidoun and Z. Lachiri, "A hybrid deep learning model for multi-class DDoS detection in SDN networks," *Annals of Telecommunications*, vol. 80, pp. 459–472, 2025, doi: 10.1007/s12243-025-01085-1.

[8] M. S. Sawah, "Distributed denial of service (DDoS) classification based on random forest model with backward elimination algorithm and grid search algorithm," *Sci. Rep.*, vol. 15, no. 1, p. 19063, 2025, doi: 10.1038/s41598-025-03868-x.

[9] U. B. Clinton, N. Hoque, and K. R. Singh, "Classification of DDoS attack traffic on SDN network environment using deep learning," *Cybersecurity*, vol. 7, no. 1, p. 23, 2024, doi: 10.1186/s42400-024-00219-7.

[10] N. Albanbay, "Federated Learning-Based Intrusion Detection in IoT Networks: Performance Evaluation and Data Scaling Study," *Journal of Sensor and Actuator Networks (JSAN)*, vol. 14, no. 4, p. 78, 2025, doi: 10.3390/jsan14040078.

[11] A. A. Bahashwan, M. Anbar, S. Manickam, T. A. Al-Amiedy, M. A. Aladaileh, and I. H. Hasbullah, "A Systematic Literature Review on Machine Learning and Deep Learning Approaches for Detecting DDoS Attacks in Software-Defined Networking," *Sensors*, vol. 23, no. 9, p. 4441, 2023, doi: 10.3390/s23094441.

[12] S. Ganeshan and R. K. Ramasamy, "A Systematic Review of Machine-Learning-Based Detection of DDoS Attacks in Software-Defined Networks," *Future Internet*, vol. 16, no. 2, p. 109, 2026, doi: 10.3390/fi18020109.

[13] Tasmi *et al.*, "Pengenalan Pola Serangan pada Internet of Thing (IoT) Menggunakan Support Vector Machine (SVM) dengan Tiga Kernel," *Jurnal PROCESSOR*, vol. 18, no. 2, pp. 241–251, 2023, doi: 10.33998/processor.2023.18.2.1457.

[14] S. Abiramasundari and V. Ramaswamy, "Distributed Denial-of-Service (DDoS) attack detection using supervised machine learning algorithms," *Sci. Rep.*, vol. 15, no. 1, p. 13098, 2025, doi: 10.1038/s41598-024-84879-y.

[15] A. Alabdulatif, N. N. Thilakarathne, and M. Aashiq, "Machine Learning Enabled Novel Real-Time IoT Targeted DoS/DDoS Cyber Attack Detection System," *Computers, Materials & Continua*, vol. 79, no. 2, pp. 312–330, 2024, doi: 10.32604/cmc.2024.054610.

[16] A. Dembele, E. Mwangi, K. K. Ronoh, and E. O. Ataro, "A Novel Approach for Detection of DDoS Attacks in Software-Defined Networks Based on Grey Wolf Optimizer and Support Vector Machine," *SSRG International Journal of Electrical and Electronics Engineering*, vol. 11, no. 3, pp. 86–96, 2024, doi: 10.14445/23488379/IJEEE-V11I3P107.

[17] S. Ginta, P. Hia, N. Hayati, D. Hindarto, and A. Sani, "Blockchain and SVM Integration for Distributed DDoS Attack Detection," *Sinkron : Jurnal dan Penelitian Teknik Informatika*, vol. 10, no. 1, pp. 75–84, 2026, doi: 10.33395/sinkron.v10i1.15483.

[18] F. Musumeci, "Machine-Learning-Enabled DDoS Attacks Detection in P4 Programmable Networks," *Journal of Network and Systems Management*, vol. 30, no. 1, pp. 1–27, 2022, doi: 10.1007/s10922-021-09633-5.

[19] A. A. Alhussain and B. S. Alsulami, "DDoS Detection by Using Machine Learning," *Journal of Information Systems Engineering and Management*, vol. 10, no. 54s, 2025, doi: 10.52783/jisem.v10i54s.11045.

[20] S. Rasheed and M. S. Rathore, "Support Vector Machine Based DDoS Detection and Mitigation in Software Defined Networks," *Journal of Innovative Computing and Emerging Technologies*, vol. 4, no. 2, pp. 54–68, 2024, doi: https://doi.org/10.56536/jicet.v4i2.161.

[21] A. A. Alsadhan, "Kernel-based machine learning intrusion detection for IPv6 ICMPv6 DDoS attacks," *Array*, vol. 26, p. 100389, 2025, doi: 10.1016/j.array.2025.100389.

[22] F. Ferdiansyah, D. Antoni, M. Valdo, C. Mukmin, and U. Ependi, "Machine Learning Models for DDoS Detection in Software-Defined Networking: A Comparative Analysis," *Journal of Informatics Engineering*, vol. 6, no. 3, pp. 1790–1803, 2024, doi: 10.51519/journalisi.v6i3.864.

[23] Y. Irawan, R. Pramitasari, W. M. Ashari, A. Nur, and H. Yansyah, "Support Vector Machine Classification Algorithm for Detecting DDoS Attacks on Network Traffic," *Journal of Applied Informatics and Computing (JAIC)*, vol. 9, no. 4, pp. 1945–1954, 2025, doi: 10.30871/jaic.v9i4.10003.

[24] M. S. Sawah, H. Elmannai, A. A. El-Bary, and K. Lotfy, "Traffic Feature Selection and Distributed Denial of Service Attack Detection Using Machine Learning Techniques," *Sci. Rep.*, vol. 14, no. 1, p. 16021, 2024, doi: 10.1038/s41598-024-66634-6.

[25] A. L. Hadiyani and B. Handaga, "Implementasi Sistem Deteksi Anomali Berbasis Jaringan Menggunakan CNN dan SVM untuk Klasifikasi Data Secara Real-Time," *Jurnal Informatika Universitas Pamulang*, vol. 10, no. 2, pp. 75–85, 2025, doi: 10.32493/jiup.v10i2.52163.

[26] Haeruddin, Erick, and H. W. Aripradono, "Perbandingan Support Vector Machine, Random Forest Classifier, dan K-Nearest Neighbour dalam Pendeteksian Anomali pada Jaringan DDoS," *JTIM: Jurnal Teknologi Informasi dan Multimedia*, vol. 7, no. 1, pp. 23–33, 2025, doi: 10.35746/jtim.v7i1.628.

[27] R. Abbas, "Machine learning-based hybrid technique to enhance cyber-attack perspective," *Journal of Cloud Computing*, vol. 11, no. 1, pp. 45–56, 2025, doi: 10.1186/s13677-025-00782-5.

[28] G. O. Anyanwu, C. I. Nwakanma, J. M. Lee, and D.-S. Kim, "RBF-SVM kernel-based model for detecting DDoS attacks in SDN integrated vehicular network," *Ad Hoc Networks*, vol. 140, p. 103026, 2023, doi: 10.1016/j.adhoc.2022.103026.