Available Online: https://proceeding.researchsynergypress.com/index.php/cset/index
RSF Conference Series: Engineering and Technology
ISSN 2809-6843 (Online) | 2809-6878 (Print)
Volume 1 Number 1 (2021): 281-290

Software Defined Network: The Comparison of SVM kernel on DDoS Detection

Rifki Indra Perwira[1], Hari Prapcoyo[2]

[1, 2] Department of Informatics, UPN ”Veteran” Yogyakarta,
Indonesia

Abstract

SDN is a new technology in the concept of a network where there is a separation between the data plane and the control plane as the brain that regulates data forwarding so that it becomes a target for DDoS attacks. Detection of DDoS attacks is an important topic in the field of network security. because of the difficulty of detecting the difference between normal traffic and anomalous attacks. Based on data from helpnetsecurity.com, in 2020 there were 4.83 million attempted DoS/DDoS attacks on various services, this shows that network security is very important. Various methods have been used in detecting DDoS attacks such as using a threshold on passing network traffic with an average traffic size compared to 3 times the standard deviation, the weakness of this method is if there is a spike in traffic it will be detected as an attack even though the traffic is normal so that it increases false positives. To maintain security on the SDN network, the reason is that a system is needed that can detect DDoS attacks anomalously by taking advantage of the habits that appear on the system and assuming that if there are deviations from the habits that appear then it is declared a DDoS attack, the SVM method is used to categorize the data traffic obtained from the controller to detect whether it is a DDoS attack or not. Based on the tests conducted with 500 training data, the accuracy is 99,2%. The conclusion of this paper is that the RBF SVM kernel can be very good at detecting anomalous DDoS attacks.

***Keywords**: Ddos, Support Vector Machine, Kernel*

INTRODUCTION

In its development, computer networks have changed from time to time. Software-defined networking (SDN) is a networking concept that is widely used today. The basic concept of SDN is a clear separation between the control plane and forwarding plane, the separation is carried out by a standard protocol on software-defined networking (SDN) networks, namely the OpenFlow protocol, and this network concept implements system abstraction and isolates the complexities that exist in components or subsystems by defines a standard interface, thus achieving the goal of obtaining a high level of efficiency in the network and security (Harja et al., 2019). Currently, human needs are very dependent on the existence of digital information or data. The greater the demand for information, the sharper the increase in incidents or security breaches on network systems. (Ariyanto et al., 2020). The SDN network is a network concept that is centered on the control plane, so this network architecture is the target of Denial of Service (DoS) and Distributed Denial of Service (DDoS) attacks. DoS is the most common type of attack, this attack is used to attack a computer or server on a network by sending fake traffic in large quantities to consume computer resources. Some of the most commonly used DoS attack methods are flooding and TCP SYN attacks. (Saeedi, 2019), while Distributed Denial of Service (DDoS) are DoS attacks that are carried out at the same time. To detect DDoS, machine learning is still one of the solutions in detecting anomaly attacks. The several methods that have been applied in previous studies to classify DoS/DDoS attacks have advantages and disadvantages, including KNN getting high accuracy above 90% but the results are influenced by the amount of training data used, so the accuracy value obtained is unstable. CNN has a high accuracy value, but is more suitable for images, the data flow used in the study must be converted into images before classification is carried out, this makes the processing time longer. Therefore, this study will try to use the machine learning Support Vector Machine (SVM) method in identifying DDoS attacks.

Corresponding author:
rifki@upnyk.ac.id; hari.prapcoyo@upnyk.ac.id
DOI: 10.31098/cset.v1i1.413

LITERATUREREVIEW

Previous research has included:

1. A computer network is an "interconnected" collection of two or more autonomous computers connected by cables or wireless transmission media. If one computer can shut down another computer, restart it, or exercise other control, this computer is not independent (cannot control another computer with full access rights). According to (Yudianto, 2014) a computer network is a system consisting of several computers that communicate with each other (email, instant messages), share resources (printers, CPUs), and can access various information (web browsers). To be able to achieve its goals, each part of the computer network can request and provide services. The party requesting/receiving the service is called the client, and the party providing/sending the service is called the server. This design is called a client-server system and is used in almost all computer network applications.
2. Software Defined Network (SDN) is a new paradigm in designing, managing, and implementing networks for complex needs (Perwira et al., 2019). The concept of SDN network application is to separate the control and forwarding functions contained in the data plane. In conventional networks the control plane and data plane are still combined on the same device, while in SDN, the control plane is separated into a centralized one, so that network management becomes faster and more efficient.
3. Denial of Service (DoS) and Distributed Denial of Service (DDOS) are attacks that are often encountered in several cybercrime cases, these attacks aim to damage or disrupt services. This attack works by depleting the resources of the server so that the server cannot be accessed and cannot perform its functions. Basically DOS and DDOS are the same attack, but DDOS is a DoS attack that can be said to be structured. With the same mechanism as DoS but the impact caused by a DDOS attack is much greater than a DOS attack (Saeedi, 2019).
4. Intrusion prevention system is an approach in detecting attacks against the network by using software on network applications running on the firewall. The way IPS works (Sianturi, 2018) is to analyze every data packet that runs based on the network protocol and will track behavior that threatens network security. IPS will record and identify packets using signatures to detect traffic on the network which can prevent attacks immediately.
5. Intrusion Detection System was developed to detect attacks on the network by classifying network attacks in the form of DoS and Probing. Several studies on IDS include using the K-Nearest Neighbor method (Ariyanto et al., 2020) which aims to create an IDS application to detect DoS and Probes in a network. IDS is combined with the Deep Learning method for DoS attack detection using the Convolutional Neural Network (CNN) VGG-19 algorithm (Kurniawan, 2020) as a classification in determining the traffic that enters the DoS attack network or not.
6. The Support Vector Machine (SVM) (Somvanshi et al, 2017) was developed for several studies on data classification. This method is also applied in various attack detection techniques against the network. SVM has advantages in classifying nonlinear data with a kernel that can be used to map data into other higher dimensions, besides that the selection of the right parameter values for RBF is able to produce better accuracy compared to other kernel techniques.

In this study, we base some existing literature on the theme of research in the field of detection of attacks on networks. In general, the SVM algorithm is used in text classification (Pratama & Trilaksono, 2015), complaint classification on Facebook iRaise Helpdesk (Fatmawati & Affandes, 2017), spam email classification using SVM and KNN (Pratiwi & Ulama, 2016). Several studies have also compared the accuracy of the SVM method with several other methods. Among them are SVM with Naïve Bayes (Sasongko & Arifin, 2019) to determine the selection of school students' paths. Text classification for the categories of environment, sport, politics and art by using comparisons between SVM, KNN and Naïve Bayes with better accuracy results from SVM.

With reference to these studies, the SVM method was developed in many related studies to improve accuracy in data classification. The linear kernel SVM is used for the classification of hepatitis disease diagnosis (Nurajizah, 2016) with the results of the RBF kernel having the highest accuracy of 96% while the linear kernel is 83%. Comparison of kernel accuracy between RBF, Linear, Sigmoid and Polynomial kernels in terms of cancer cell classification (Dsouza & Ansari, 2018) based on accuracy, kappa value and sensitivity, the highest value was obtained by RBF with an accuracy of 96.47%. Evaluation of SVM using various kernel techniques (Hossain & Miah, 2016) for customer churn prediction with the highest accuracy results obtained by the Laplacian kernel of 94.55%. The generalizability of SVM for the recognition of splice sites in DNA (Kerami & Murfi, 2010) for pattern recognition that is linear and non-linear, the results obtained are 96.8% accuracy.

The difference between this study and previous research is in the data used and how the data is collected based on the type of DDOS attack applied, this study uses the values of Speed_of_Ip, Speed_of_Flow_Entries, and Ratio_of_Pair-Flow_Entries as feature datasets. This research will also use the Support Vector Machine algorithm and apply four kernels (Linear, RBF, Sigmoid, and Polynomial) to see the difference in accuracy, precision and recall obtained.

RESEARCH METHODOLOGY

The research methodology is carried out in several steps which are generally illustrated in the figure below.

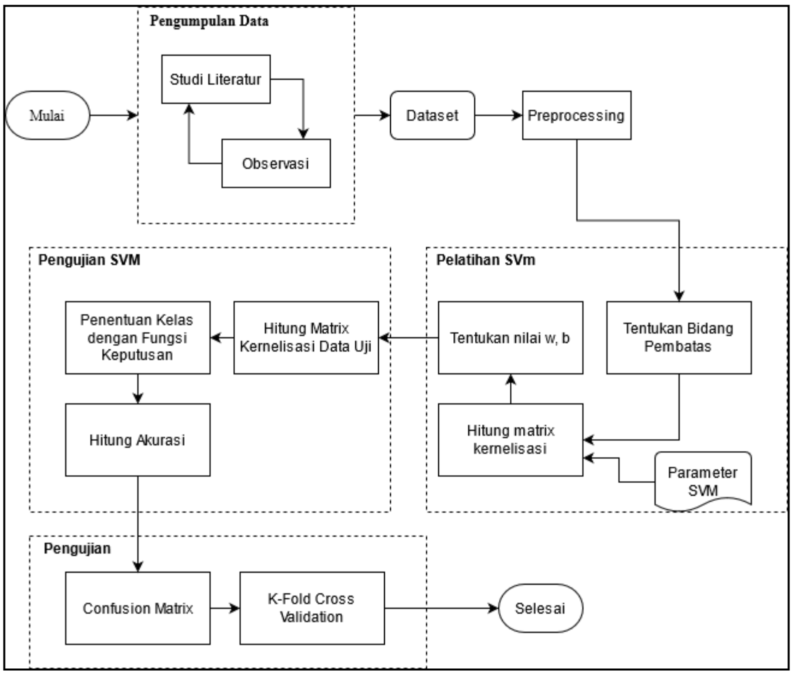

Figure 1. Research stages.

The research phase begins with data collection with the aim of creating a detection system. In this data collection process there are two stages, namely literature study and observation. This literature study carried out the data collection process by analyzing problems related to DDOS attacks that occurred on SDN. While the observation is done by collecting logs from the packages in the SDN. The two processes produce a dataset for training/preprocessing then training and testing the SVM method.

Observation.

SDN architecture management is done with a centrally created control plane. Where one controller (control plane) can configure more than one data plane (forwarding plane) directly sent by the host (data plane) it will definitely pass through the controller which acts as the control plane. There are 4 new features that will be created to be used in detecting DDOS on SDN networks, namely: 1. Speed of IP Source (This feature counts the total number of IP sources that enter the network with a predetermined time interval), 2. Flow count of the traffic (each incoming network traffic has a certain number of flow counts), 3. Speed of flow entries (SFE is the total number of flow entries that enter the switch at certain intervals), and 4. Ratio of pair-flow entries (total flow entries that enter the switch) entered into the switch which is the interactive IP divided by the total number of flows in a certain period T).

Literature study.

Literature Study is an activity carried out by collecting references from journals and books which aims to strengthen the theoretical basis in analyzing and solving problems. From these studies, there are four related feature extraction methods in the paper Safe-Guard Scheme for Protecting Control plane Against DDOS attacks in SDN (Wang et al., 2019) namely traffic data collection and byte rate calculation, symmetric and asymmetric flow variations, and counts the small number of packets that enter the network. A similar approach to the Density Peak Clustering algorithm (He et al., 2017) is to detect anomalies by collecting traffic features and correlation characteristics of malicious traffic.

Preprocessing

Before the testing stage, the data model needs to be preprocessed first, which is called preprocessing with data extraction that will be used for training data so as to form connection record features. Followed by the data normalization process using the SVM method, namely by classifying the data into two labels between -1 for non-DDoS attacks and 1 for DDoS attacks.

Training SVM

In training data training using the SVM method in general, the stages are as shown in Figure 2 below.

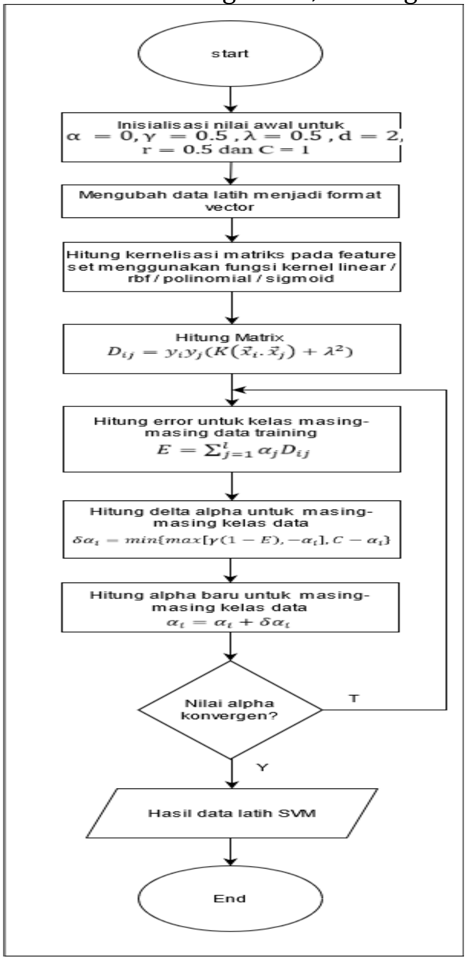
start
Inisialisasi nilai awal untuk $\alpha = 0, \gamma = 0.5, \lambda = 0.5, d = 2, r = 0.5$ dan $C = 1$
Mengubah data latih menjadi format vector
Hitung kernelisasi matriks pada feature set menggunakan fungsi kernel linear / rbf / polinomial / sigmoid
Hitung Matrix $D_{ij} = y_i y_j(K(\vec{x}_i . \vec{x}_j) + \lambda^2)$
Hitung error untuk kelas masing-masing data training $E = \sum_{j=1}^{l} \alpha_j D_{ij}$
Hitung delta alpha untuk masing-masing kelas data $\delta\alpha_i = min\{max[\gamma(1-E), -\alpha_i], C - \alpha_i\}$
Hitung alpha baru untuk masing-masing kelas data $\alpha_i = \alpha_i + \delta\alpha_i$
Nilai alpha konvergen?
T
Y
Hasil data latih SVM
End

Figure 2. Flowchart training process SVM.

SVM training with linear kernel.

The steps of the data training process using a linear kernel are carried out one by one with the general stages starting from initializing the initial value for α,γ,λ and C with value α = 0, γ = 0.5 , λ = 0.5 and C = 1. Converting the training data into vector format and the kernelization process is carried out by looking for the matrix value $K(x_i, x_j)$ with equation $K(x_1, x_1) = (x_1{}^T x_1)$. Hessian matrix calculations are used with linear kernelization to get the results of all data and calculate the error value by looking for the value of delta alpha:

$x_1 = Min(Max(0.5(1 - (0.596)), -0.5), 1 - 0.5)$

$x_1 = Min(0.202, 0.5)$

$x_1 = 0.202$

After that, the new alpha value is used using the equation $x_1 = 0.5 + 0.202 = 0.702$ for $x_1 until\ x_3$ while $x_4 = 0.5 - 0.167575 = 0.332425$ for $x_4\ until\ x_6$ with iteration for 5,6,7 times 15 times get value α and the result is:

$$\alpha_i = \begin{matrix} 0.8390 \\ 0.9829 \\ 0.8687 \\ 0.3745 \\ 0.3056 \\ [\,0.5\,] \end{matrix}$$

Dari nilai α diatas dilanjutkan dengan mencari nilai bias menggunakan persamaan $b = -\frac{1}{2}(\langle \vec{w}.x_{-1} \rangle + \langle \vec{w}.x_{+1} \rangle)$ dengan melakukan perhitungan terlebih dahulu nilai w:

$\vec{w}_{i-}$ adalah bobot *dot product* data dengan alpha terbesar di kelas negatif.

$\vec{w}_{i+}$ adalah bobot *dot product* data dengan alpha terbesar di kelas positif.

$\vec{w}.x_-$ (kelas negatif) = $(-1 * 0.8390 * 0.0329) + (-1 * 0.9829 * 0.0449) + (-1 * 0.8687 * 0.0358) + (1 * 0.3745 * 0.0915) + (1 * 0.3056 * 0.1332) + (1 * 0.5 * 0.1969) = 0.070590873$

$\vec{w}.x_+$ (kelas positif) = $(-1 * 0.8390 * 0.1969) + (-1 * 0.9829 * 0.3015) + (-1 * 0.8687 * 0.2215) + (1 * 0.3745 * 0.7131) + (1 * 0.3056 * 1.0762) + (1 * 0.5 * 1.6301) = 0.757051907$

Makanilai $b = -\frac{1}{2}(0.070590873 + (0.757051907)) = -0.41382139$

SVM training with polynomial kernel.

The data training process using a polynomial kernel is carried out one by one by initializing the initial value for α, γ, λ and C with value α = 0, γ = 0.5 , λ = 0.5 , d = 2, r = 0.5 and C = 1 and training data is exchanged into *vector* format. For matriks $K(x_i, x_j)$ polynomial kernel similar to linear kernela is added +1 and exponential d=2. Continue into hessian matriks

$D_{1,1} = (-1)(-1)(1.0668) + (0.5^2) = 1.3168$

$D_{1,2} = (-1)(-1)(1.0918) + (0.5^2) = 1.3418$

With polynomial kernelization, the results are obtained for all matrix data and find the error value with the formula $x_1 = (1.3168 + 1.3418 + 1.3227 - 0.9414 - 1.0342 - 1.1825) * 0.5 = 0.4116$ to be used to find the value of delta alpha:

$x_1 = Min(Max(0.5(1 - (0.4116)), -0.5), 1 - 0.5)$

$x_1 = Min(0.2942, 0.5)$

$x_1 = 0.2942$

Generates a delta alpha matrix value to find a new alpha value $x_1 = 0.5 + 0.2942 = 0.7942$ for $x_1\ until\ x_3$ while $x_4 = 0.5 - 0.5 = 0$ produce a matrix followed by the iteration process 5,6,7 with 15 times within value α the result is :

$$\alpha_i = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0.5 \\ 0.5 \\ 0.5 \end{bmatrix}$$

By calculating the bias is $b = -\frac{1}{2}(\langle \vec{w}.x_{-1} \rangle + \langle \vec{w}.x_{+1} \rangle)$. First, calculate the value of w :

$\vec{w}_{i-}$ is weighted *dot product* with alpha value largest in the negative class.
$\vec{w}_{i+}$ is weighted *dot product* with alpha value largest in the positive class.
$\vec{w}.x_{-}$ (negative class) $= (-1 * 1 * 1.0668) + (-1 * 1 * 1.0918) + (-1 * 1 * 1.0727) + (1 * 0.5 * 1.1914) + (1 * 0.5 * 1.2842) + (1 * 0.5 * 1.4325) = -1.27725$.
$\vec{w}.x_{+}$ (positive class) $= (-1 * 1 * 1.4325) + (-1 * 1 * 1.6938) + (-1 * 1 * 1.492) + (1 * 0.5 * 2.9347) + (1 * 0.5 * 4.3105) + (1 * 0.5 * 6.917) = 2.4628$.
With value $b = -\frac{1}{2}(-1.27725 + (2.4628)) = -0.592775$

SVM training with Sigmoid kernel.

The data training process using the sigmoid kernel is carried out one by one by initializing the initial value for α, γ, λ and C with value $\alpha = 0, \gamma = 0.5$, $\lambda = 0.5$, d = 2, r = 0.5 and C = 1 by converting the data into a format *vector* and perform the kernelization process to find the matrix value $K(x_i, x_j)$ with $K(x_1, x_1) = tanh(\gamma * (x_1{}^T x_1) + r)$. After the matrix is formed, the calculation is carried out $D_{1,1} = (-1)(-1)(0.475) + (0.5^2) = 0.725$ and $D_{1,2} = (-1)(-1)(0.4796) + (0.5^2) = 0.7296$ followed by sigmoid kernelization to find errors with the equation $x_1 = (0.725 + 0.7296 + 0.7261 - 0.2474 - 0.2629 - 0.286) * 0.5 = 0.6922$ to get the error matrix. The delta alpha value is obtained by

$x_1 = Min(Max(0.5(1 - (0.6922)), -0.5), 1 - 0.5)$

$x_1 = Min(0.1539 , 0.5)$

$x_1 = 0.1539$

Which produces a delta alpha matrix while to get a new alpha value using the equation $x_1 = 0.5 + 0.1539 = 0.6539$ dan $x_4 = 0.5 + 0.0287 = 0.5287$ with process iteration 5,6,7 into 15 times with α value is produce:

$$\alpha_i = \begin{bmatrix} 0.72233457 \\ 0.75557228 \\ 0.729980269 \\ 0.530447698 \\ 0.492902208 \\ 0.457791613 \end{bmatrix}$$

By finding the value of the bias with the following equation: $b = -\frac{1}{2}(\langle \vec{w}.x_{-1} \rangle + \langle \vec{w}.x_{+1} \rangle)$.

Firstly, calculate the value of w :
$\vec{w}_{i-}$ is weighted *dot product* with biggest alpha data in negative class.
$\vec{w}_{i+}$ is weighted *dot product* with biggest alpha data in positive class.
$\vec{w}.x_{-}$ (negatif class) $= (-1 * 0.7223 * 0.475) + (-1 * 0.7556 * 0.4796) + (-1 * 0.7299 * 0.4761) + (1 * 0.5304 * 0.4974) + (1 * 0.4929 * 0.5129) + (1 * 0.4578 * 0.536) = -0.29099446$
$\vec{w}.x_{+}$ (positive class) $= (-1 * 0.7223 * 0.536) + (-1 * 0.7556 * 0.5722) + (-1 * 0.7299 * 0.5447) + (1 * 0.5304 * 0.6945) + (1 * 0.4929 * 0.7772) + (1 * 0.4578 * 0.8656) = -0.069386098$
With value $b = -\frac{1}{2}(-0.29099446 - 0.069386098) = 0.180190279$

SVM training with RBF kernel

The data training process using the RBF kernel is carried out one by one as the process below with the first step is initializing the initial value for α,γ,λ and C with values α = 0,γ = 0.5 , λ = 0.5 , d = 2, r = 0.5 and C = 1 with the training data is changed into *vector* for finding matrix $K(x_i, x_j)$ with equation $K(x_1, x_1) = \exp(-3||x_1 - x_1||^2)$. Continuing to search hessian matrix with formula $D_{1,2} = (-1)(-1)(0.9799) + (0.5^2) = 1.2299$ to get RBF kernelization. While the error is calculated using the equation $x_1 = (1.25 + 1.2299 + 1.2486 - 0.3596 + 0.0117 + 0.2277) * 0.5 = 1.80415$ which will produce an error matrix that is used to find delta alpha

$x_1 = Min(Max(0.5(1 - (1.80415)), -0.5), 1 - 0.5)$

$x_1 = Min(-0.402075, 0.5)$

$x_1 = -0.402075$

Which produces a delta alpha matrix while to get a new alpha value using the equation $x_1 = 0.5 - 0.402075 = 0.097925$ dan $x_4 = 0.5 + 0.12085 = 0.62085$ with iteration process 5,6,7 into 15 times and α value is produce:

$$\alpha_i = \begin{matrix} 0.2877 \\ 0.2976 \\ 0.2876 \\ 0.6594 \\ 0.3305 \\ [0.3134] \end{matrix}$$

By finding the value of the bias with the following equation $b = -\frac{1}{2}(\langle\vec{w}.x_{-1}\rangle + \langle\vec{w}.x_{+1}\rangle)$ first calculate the value of w: $\vec{w}_{i-}$ is weighted *dot product* data with the largest alpha in the negative class. $\vec{w}_{i+}$ is weighted *dot product* data with the largest alpha in the positive class.

$\vec{w}.x_-$ (negative class) = $(-1 * 0.2877 * 1) + (-1 * 0.2976 * 0.9799) + (-1 * 0.2876 * 0.9986) + (1 * 0.6594 * 0.6096) + (1 * 0.3305 * 0.2383) + (1 * 0.3134 * 0.0223) = -0.378775151$

$\vec{w}.x_+$ (positive class) = $(-1 * 0.2877 * 0.0223) + (-1 * 0.2976 * 0.038) + (-1 * 0.2876 * 0.0253) + (1 * 0.6594 * 0.2108) + (1 * 0.3305 * 0.5667) + (1 * 0.3134 * 1) = 0.61473118$

Then value $b = -\frac{1}{2}(-0.378775151 + (0.61473118)) = -0.117978014$

FINDING AND DISCUSSION

SVM training

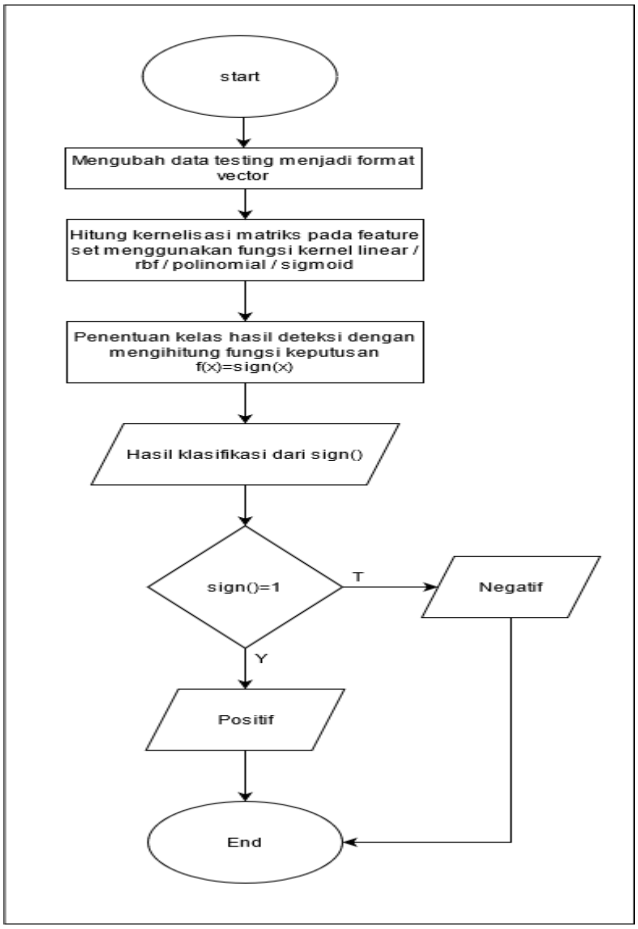
start
Mengubah data testing menjadi format vector
Hitung kernelisasi matriks pada feature set menggunakan fungsi kernel linear / rbf / polinomial / sigmoid
Penentuan kelas hasil deteksi dengan mengihitung fungsi keputusan f(x)=sign(x)
Hasil klasifikasi dari sign()
sign()=1
T
Negatif
Y
Positif
End

Figure 3. Flowchart SVM training.

In the testing phase, testing the model that has been generated at the previous training stage. The test data after going through the preprocessing process and the process of changing the data format will produce a test vector as follows:x = [0.865811966, 0.865811966, 0.100001324].

Linear kernel training

In linear kernel testing calculate dot product with all training data from x_1 to x_6 using kernel function:$K(x, x_1)$ = $(x^T x_1)$ = ([0.8658,0.8658, 0.1001]^T[0.1068 0.1 0.1068]) = 0.1898

The calculation is carried out until the 6th data, and the kernel function is obtained:

Table 4.1 Results of Kernel Training Data Functions.

$K(x, x_1)$	$K(x, x_2)$	$K(x, x_3)$	$K(x, x_4)$	$K(x, x_5)$	$K(x, x_6)$
0.1898	0.2905	0.2135	0.6864	1.0357	1.5685

To find out the class of testing data, the decision function is calculated using the equation $f(x) = \vec{w}.x + b$ atau $f(x) = \sum^{m}_{i=1} \alpha_i y_i K(x, x_i) + b$. Based on this, if the value of the calculation result is 1, it can be seen that the data flow used in SVM testing using a linear kernel is a DDoS attack.

Polynomial kernel training

Dot product calculation for kernel polynomial with all training data from x_1 to x_6 yields kernel function:

$K(x, x_1) = (x^T x_1 + 1)^d$ = ([0.8658,0.8658,0.1001]^T[0.1068 0.1 0.1068] + 1)^2 = 1.4157

The calculation is carried out until the 6th data, and the kernel function is obtained:

Table 4.2 Results of Kernel Training Data Functions..

$K(x, x_1)$	$K(x, x_2)$	$K(x, x_3)$	$K(x, x_4)$	$K(x, x_5)$	$K(x, x_6)$
1.4157	1.6654	1.4726	2.844	4.1441	6.5972

To find out the class of testing data, the decision function is calculated using the following equation:

$f(x) = \vec{w}.x + b$ or $f(x) = \sum^{m}{}_{i=1}\ \alpha_i y_i K(x, x_i) + b$. Based on the calculations obtained above, it is known that the data flow used in SVM testing using the Polynomial kernel is a DDoS attack.

Sigmoid kernel training

Dot product training for sigmoid kernel all training data from x_1 to x_6 using kernel functions:

$K(x_1, x_1) = tanh(\gamma * ({x_1}^T x_1) + r)$

$= tanh(\gamma * ([0.8658, 0.8658, 0.1001]^T[0.1068, 0.1, 0.1068]) + r) = tanh(0.5 * 0.1898 + 0.5)$

$= 0.5335$

The calculation is carried out until the 6th data, and the kernel function is obtained:

Table 4.3 Results of Kernel Training Data Functions.

$K(x, x_1)$	$K(x, x_2)$	$K(x, x_3)$	$K(x, x_4)$	$K(x, x_5)$	$K(x, x_6)$
0.5335	0.5685	0.5419	0.6876	0.769	0.8577

To find out the class of testing data, the decision function is calculated using the following equation:

$f(x) = \vec{w}.x + b$ or $f(x) = \sum^{m}{}_{i=1}\ \alpha_i y_i K(x, x_i) + b$. Based on the calculations obtained above, it can be seen that the data flow used in SVM testing using a linear kernel is a DDoS attack.

RBF kernel training

Testing dot product for RBF kernel with all training data x_1 to x_6 using kernel functions:$K(x, x_1) = \exp(-3\|x - x_1\|^2)$

$= \exp(-3\|[0.8658, 0.8658, 0.1001] - [0.1068\quad 0.1\quad 0.1068]\|^2)$

$= \exp(-3\|[0.758974359\ \ 0.765811966\ \ -0.006836282]\|^2)$

$= \exp(-3.487670338)$

$= 0.030572012$

The calculation is carried out until the 6th data, and the kernel function is obtained:

$K(x, x_1)$	$K(x, x_2)$	$K(x, x_3)$	$K(x, x_4)$	$K(x, x_5)$	$K(x, x_6)$
0.0306	0.0510	0.0346	0.2579	0.6384	0.9930

Tabel 4.4 Results of Kernel Training Data Functions.

To find out the class of testing data, the decision function is calculated using the equation berikut :

$f(x) = \vec{w}.x + b$ or $f(x) = \sum^{m}{}_{i=1}\ \alpha_i y_i K(x, x_i) + b$ Based on the calculations obtained above, it can be seen that the data flow used in SVM testing using the RBF kernel is a DDoS attack.

Accuracy comparations

From the test results, the comparison of the kernel on SVM for DDoS detection produces:

Table 4.5

No	Kernel	Accuracy	Cross Validation
1	Linear	98.29411764705885	99.75308641975309
2	Polynomial	98.52941176470588	99.26227040048178
3	RBF	99.26470588235294	99.58374859975309
4	Sigmoid	98.17647058823145	99.01535682023486

CONCLUSION AND FURTHER RESEARCH

The conclusion of this study is to show the percentage of accuracy of the existing kernels on the support vector machine. The result of the highest accuracy of DDoS detection using SVM is the RBF kernel with 99.26% with a dataset of the number of ips entered during the attack simulation using a mininet connected to the OVS controller. the number of incoming packets and the number of recorded IP addresses will affect the dataset and accuracy results. For other kernels, SVM also has high accuracy and balanced results. This proves that SVM can be used in the detection of distributed andial of service attacks.

Acknowledgement

The authors would like to thank the Institute for Research and Community Service at Universitas Pembangunan Nasional Veteran Yogyakarta, Indonesia, for providing funds for this research.

REFERENCES

Ariyanto, Y., Al, V., Firdaus, H., & Pramana, H. (2020). *Klasifikasi Jenis serangan DOS dan Probing pada IDS menggunakan metode K- Nearest Neighbor*. *3*, 1–5.

Dsouza, K. J., & Ansari, Z. A. (2018). Experimental exploration of support vector machine for cancer cell classification. *Proceedings - 2017 IEEE International Conference on Cloud Computing in Emerging Markets, CCEM 2017*, *2018-Janua*, 29–34. https://doi.org/10.1109/CCEM.2017.15

Fatmawati, & Affandes, M. (2017). *Klasifikasi Keluhan Menggunakan Metode Support Vector Machine (SVM) (Studi Kasus : Akun Facebook Group iRaise Helpdesk)*. *3*(1), 24–30.

He, D., Chan, S., Ni, X., & Guizani, M. (2017). Software-Defined-Networking-Enabled Traffic Anomaly Detection and Mitigation. *IEEE Internet of Things Journal*, *4*(6), 1890–1898. https://doi.org/10.1109/JIOT.2017.2694702

Hossain, M. M., & Miah, M. S. (2016). Evaluation of different SVM kernels for predicting customer churn. *2015 18th International Conference on Computer and Information Technology, ICCIT 2015*, 1–4. https://doi.org/10.1109/ICCITechn.2015.7488032

Kerami, D., & Murfi, H. (2010). Kajian Kemampuan Generalisasi Support Vector Machine Dalam Pengenalan Jenis Splice Sites Pada Barisan Dna. *MAKARA of Science Series*, *8*(3), 89–95. https://doi.org/10.7454/mss.v8i3.451

Nurajizah, S. (2016). Penerapan Metode Support Vector Machine Berbasis Particle. *Jurnal Techno Nusa Mandiri*, *X*(2), 209–216. https://doi.org/10.20527/klik.v3i1.39

Perwira, R. I., Fauziah, Y., Mahendra, I. P. R., Prasetyo, D. B., & Simanjuntak, O. S. (2019). Anomaly-based Intrusion Detection and Prevention Using Adaptive Boosting in Software-defined Network. *Proceeding - 2019 5th International Conference on Science in Information Technology: Embracing Industry 4.0: Towards Innovation in Cyber Physical System, ICSITech 2019*, 188–192. https://doi.org/10.1109/ICSITech46713.2019.8987531

Pratiwi, S. N. D., & Ulama, B. S. S. (2016). Klasifikasi Email Spam dengan Menggunakan Metode Support Vector Machine dan k-Nearest Neighbor. *Jurnal Sains Dan Seni ITS*, *5*(2), 344–349.

Saeedi, K. (2019). *Machine Learning for Ddos Detection in Packet Core Network for IoT*.

Sasongko, T. B., & Arifin, O. (2019). Implementasi Metode Forward Selection pada Algoritma Support Vector Machine (SVM) dan Naive Bayes Classifier Kernel Density (Studi Kasus Klasifikasi Jalur Minat SMA). *Jurnal Teknologi Informasi Dan Ilmu Komputer*, *6*(4), 383–388. https://doi.org/10.25126/jtiik.201961000

Sianturi, A. H. (2018). Simulasi Pencegahan Serangan Denial of Service (DoS) Pada Intrusion Prevention System (IPS) dan algoritma genetika. *Simulasi Pencegahan Serangan Denial of Service (DoS) Pada Intrusion Prevention System (IPS) Dan Algoritma Genetika*, 44–48.

Somvanshi, M., Chavan, P., Tambade, S., & Shinde, S. V. (2017). A review of machine learning techniques using decision tree and support vector machine. *Proceedings - 2nd International Conference on Computing, Communication, Control and Automation, ICCUBEA 2016*.

Wang, Y., Hu, T., Tang, G., Xie, J., & Lu, J. (2019). SGS: Safe-Guard Scheme for Protecting Control Plane Against DDoS Attacks in Software-Defined Networking. *IEEE Access*, *7*, 34699–34710. https://doi.org/10.1109/ACCESS.2019.2895092

Yudianto, M. J. N. (2014). Jaringan Komputer dan Pengertiannya. *Ilmukomputer.Com*, *Vol.1*, 1–10.