1st Pelita International Conference
Volume 01 No 01 September 2023
E-ISSN: 3026-4235

https://jurnal.pelitabangsa.ac.id/index.php/pic

Shielding the Digital Realm with K-Nearest Neighbors in Network Security

Andri Firmansyah[1], Ananto Tri Sasongko[2]
[1,2]Informatic Engineering, Universitas Pelita Bangsa, Indoensia

Abstract
Network security is a paramount concern in today's digitally interconnected world. The constant evolution of cyber threats necessitates innovative approaches to safeguarding the digital realm. This paper explores the application of K-Nearest Neighbors (K-NN) in network security, offering a shield against intrusions and vulnerabilities. The research begins with a comprehensive introduction to the escalating landscape of network security challenges, highlighting the critical role of intrusion detection. K-NN, renowned for its pattern recognition capabilities, is a promising solution to fortify network defenses. The methodological journey involves data collection and preprocessing, where relevant datasets are curated and prepared for analysis. Subsequently, a K-NN model is meticulously crafted, focusing on parameter tuning and optimal K-value selection. Metrics, including accuracy, precision, recall, and F1-score, are employed to assess its performance. The findings provide insights into the model's strengths and limitations, offering a valuable perspective on its suitability for real-world network protection. This research demonstrates the potential of K-Nearest Neighbors in shielding the digital realm, reinforcing network security, and exemplifying the efficacy of machine learning in countering evolving cyber threats. It underscores the significance of proactive measures in preserving the integrity and confidentiality of digital assets in an increasingly interconnected world.

Keywords: Network Security, K-Nearest Neighbors, Intrusion Detection, Cybersecurity, Digital Protection

INTRODUCTION

A DDoS attack is an attack that is launched by sending data packets continuously to a machine or computer network. This may cause resources to become inaccessible or usable to users. DDoS attacks are a variant of DOS attacks, with the main difference being the dispersion of the attack source. This attack is dangerous and threatening because it can overwhelm the network and block access to the server. Many methods are used to detect DDoS attacks, including statistics, machine learning, SDN architecture, blockchain, and others. Researchers have been studying DDoS for decades, and the following article describes various strategies for dealing with such attacks.

The article by Bahta discusses a computer network interference prevention system based on Snort pattern detection. In this article, the author reveals that the limitations of firewalls and network intrusion detection systems encourage the development of new approaches to network security. The author proposes a network intrusion prevention system that is not only capable of detecting attacks but also has a prevention mechanism. To test the effectiveness of this system, the authors tested using three types of attacks, namely scanning (nmap), XSS (Cross Side Scripting), and DDOS (hping). The test results show that the network interference prevention system can detect scanning attacks by detecting 12 packets sent. This system can also detect DDOS attacks by detecting 512 packets sent within 10 seconds. In testing attack prevention, the authors compare connection activity when the server is running with a prevention system and

when the server is running without a prevention system. The results show that when the server runs with the prevention system, the reply of 25 ICMP request packets does not return, indicating that the prevention system has worked (Bahta, 2019).

Putra and Aryadi discussed implementing preventive measures against flooding attacks on TCP and UDP protocols at the PDAM Tirta Musi Palembang Office. This research aims to improve network security and protect against potential threats. The author uses the Action Research method, which consists of several stages. At the diagnostic stage, researchers identified the main problems that exist in the internet network at PDAM Tirta Musi Palembang. After that, the planning stage is carried out, namely designing preventive measures that will be implemented. The next stage is action, where preventive measures are implemented on the PDAM network. The author uses the Action Research method to test the effectiveness of the preventive measures that have been designed. In this research, the author uses simulations to analyze flooding attacks on TCP and UDP protocols using the firewall filter method. Simulation results show that this method effectively prevents flooding attacks (Putra & Ariyadi, 2019).

Pramana et al.'s study discusses identifying Denial of Service (DoS) Attacks on networks using the C4.5 Decision Tree algorithm. DoS and DDoS attacks on computer networks can cause major losses to companies or organizations. Therefore, efforts to secure computer networks and prevent these attacks must be made by installing firewalls, antiviruses, and IDS/IPS devices. This article uses the C4.5 Decision Tree algorithm to identify DoS attacks on computer networks. This algorithm succeeded in achieving an accuracy of 90.68% in trials. In addition, this article also discusses the validation sampling technique used in forming a predictive model for the NSL-KDD training dataset using the Decision Tree C4.5 algorithm and the Naïve Bayes algorithm. Several validation sampling techniques include using K-Fold, Percentage Split, and the last one is using a testing dataset (Pramana et al., 2021)

In his research, Gregorius Hendita discusses designing a web server system scheme using the Cloudflare Magic Transit service to prevent DDoS attacks. This article explains how increased internet use during the COVID-19 pandemic increases the risk of DDoS attacks on web servers. The author suggests using the Cloudflare Magic Transit service as a solution to protect web servers from DDoS attacks. This service acts as an intermediary between the user and the server, receiving and processing all requests, and protecting the server from DDoS attacks without burdening it. This article provides an overview of how Cloudflare Magic Transit works and how it can help improve network security (Gregorius Hendita, 2022).

Hijriyanto and Ulum, in their article, discuss a comparison between Mod Evasive and DDoS Deflate in dealing with Denial of Service (DoS) attacks with slow post techniques on web servers. The author explains that a web server must be able to serve users when needed, but DoS attacks with slow post techniques can make services on the web server inaccessible. The author finds that DDoS Deflate is a better method than Mod Evasive in overcoming DoS attacks because DDoS Deflate can detect and terminate excessive connections according to the configurations made. This article provides an overview of how the two methods work and how they can help improve web server security. This article also provides information on how a slow post DoS attack works and how safeguards can be used to reduce its impact (Hijriyanto & Ulum, 2021).

An article by Alharbi et al. discusses using the KNN (K-Nearest Neighbors) algorithm, which is optimized to detect Denial-of-Service (DoS) attacks on IPv6 networks. The authors explain how increased IPv6 network traffic makes traditional intrusion detection systems like Snort less effective in detecting DoS attacks. To overcome this problem, the author suggests using the KNN algorithm, which is optimized using dimensionality reduction techniques and feature weighting based on the rate of information gain. Experimental results show that the proposed algorithm can improve the performance of DoS attack detection in IPv6 networks compared with

the traditional KNN algorithm. This article also explains how the optimized KNN algorithm can be used to improve the security of IPv6 networks from DoS attacks (Alharbi et al., 2021).

The study conducted by Farradhika Muntaka et al. explains the implementation of an Athena-based Intrusion Prevention System (IPS) to prevent Distributed Denial of Service (DDoS) attacks on Software-Defined Network (SDN) architectures. This study applies an Athena-based IPS to prevent and mitigate the impact of DDoS attacks, especially TCP SYN floods and UDP floods. Two test scenarios were conducted to determine IPS performance. The first scenario compares the impact of DDoS attacks without and with IPS applied to throughput and CPU usage on the controller. The second scenario compares the speed of the prevention function based on the features in the detection model. The first test results show that IPS can prevent DDoS attacks, as evidenced by the decrease in throughput. IPS can also reduce the CPU load on the controller when the attack is carried out by 4.95% and 7.9%, respectively. The second test's results concluded that the more appropriate and correct features used for training, the faster the IPS was in recognizing dangerous host characteristics (Farradhika Muntaha et al., 2019).

Siahaan researched how to prevent DDoS (Distributed Denial of Service) attacks on email servers. Email servers are very important for companies, but with the development of information technology and internet crime, email servers can be attacked and disrupt user services. DDoS attacks can cripple server performance by sending multiple packets from multiple source IP addresses to a single target. To prevent this attack, Fail2ban can be used to detect unusual activity and perform automatic blocks. This study uses the NDLC method: analysis, design, simulation, and implementation. With fail2ban implementation, DDoS attacks on email servers can be prevented, and incoming spam can be reduced by up to 21% (Siahaan, 2021).

In their study, Kachavimath et al. explained the detection of Distributed Denial of Service (DDoS) attacks using the Naïve Bayes and K-Nearest Neighbor algorithms for network security. DDoS is one of the most popular attacks across multiple network layers, which aims to disrupt the normal traffic of a targeted server, service, or network by flooding the target infrastructure with abnormal flood traffic. This article presents a detection model for DDoS attacks to improve enterprise network security through machine learning. The machine learning framework extracts high-level features, identifies hidden patterns of network traffic, and detects DDoS attacks. The experimental results show better performance of the K-Nearest Neighbor and Naïve Bayes algorithms than conventional learning models (Kachavimath et al., 2020).

Distributed Denial of Service (DDoS) attacks have been ravaging network availability for decades, and there is still no effective defense mechanism against them, as discussed by Dong and Sarem. However, the emergence of Software Defined Networking (SDN) provides a new way to consider defense against DDoS attacks. In this article, the authors propose two methods to detect DDoS attacks on SDN. One method adopts the DDoS attack level to identify DDoS attacks. Another method uses the enhanced K-Nearest Neighbors (KNN) algorithm based on Machine Learning (ML) to find DDoS attacks. The results of the theoretical analysis and experimental results on the dataset show that the proposed method can detect DDoS attacks better than other methods (Dong & Sarem, 2020).

Xu et al. explained in their article that an efficient method of detecting DDoS (Distributed Denial of Service) attacks on SDN (Software Defined Networking) networks is based on K-FKNN (K-means++ and Fast K-Nearest Neighbors). This research aims to improve the security of SDN networks, which are vulnerable to DDoS attacks. The proposed method uses the K-means++ algorithm to classify training data and Fast K-Nearest Neighbors to detect network flows. The experimental results show that the K-FKNN method improves detection accuracy and efficiency compared to the usual K-Nearest Neighbors method and has high precision and stability in detecting DDoS attacks on SDN networks (Xu et al., 2019).

RESEARCH METHOD

This research method aims to implement an intrusion detection approach using the K-Nearest Neighbors (K-NN) algorithm to prevent cyber attacks on computer networks. The research design used consists of several stages, which are explained as follows:

Data collection

The data for this study was obtained through valid and relevant sources, which include information about cyberattacks and related network events. The dataset used includes representative cyber and normal (non-attack) attacks.

Data Preprocessing

The collected data is processed before being used in the intrusion detection approach. The preprocessing process includes steps for cleaning data, normalizing, removing irrelevant attributes, and handling missing values.

K-NN Model Training:

The preprocessed data is used to train the K-NN model. This step involves selecting the optimal K value, calculating the distance between data points, and determining the majority class based on nearest neighbors. K-NN algorithm

Algorithm The k-nearest neighbors classification algorithm

Input: Training set $D = \{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_n, y_n)\}$
number of nearest neighbors k
a distance metric $d(\mathbf{x}, \mathbf{y})$
a test sample $\mathbf{x}$

for each training sample $(\mathbf{x}_i, y_i) \in D$ **do**
Compute $d(\mathbf{x}, \mathbf{x}_i)$, the distance between $\mathbf{x}$ and $\mathbf{x}_i$
Let $N \subseteq D$ be the set of training samples with the k smallest distances $d(\mathbf{x}, \mathbf{x}_i)$
return the majority label of the samples in N

Figure 1. K-NN Algorithm in Pseudo-Code

Model Evaluation

After the K-NN model is trained, an evaluation is carried out to measure the model's performance in detecting cyber attacks. The evaluation metrics include accuracy, precision, recall, and F1 score.

Analysis and Interpretation of Results

The evaluation results are used to analyze and interpret the performance of the K-NN model in intrusion detection. A comparison between cyberattack detection and non-attack detection provides insight into the effectiveness of the K-NN approach in deterring cyberattacks.

This research design is expected to understand better the capabilities and effectiveness of the K-NN approach in network intrusion detection to protect computer networks from cyber attacks.

Experimental Design

Following are the steps for using the Python program to conduct network intrusion detection experiments with the K-Nearest Neighbors approach:

Environmental Preparation

Install Python: Make sure Python is installed on your computer. Install required libraries: Install libraries such as pandas, scikit-learn, and matplotlib using pip or a suitable package manager.

Dataset Preparation

Import library: Import the required libraries, such as pandas, for data manipulation. Load dataset: The panda's library loads datasets from CSV files or other data sources. Preprocessing data: Perform data preprocessing as needed, such as deleting or replacing missing values, encoding, or scaling attributes.

Model Building

Import library: Import the required libraries, such as scikit-learn for creating the K-Nearest Neighbors model. Model initialization: Create a K-Nearest Neighbors model object by setting parameters such as the number of neighbors (K) and the distance metric used. Data split: Split the dataset into training and test subsets using scikit-learn's train_test_split.

Model Training and Evaluation

Model training: Train the K-Nearest Neighbors model using the training subset by calling the fit method on the model objects. Model testing: Use the test subset to make predictions using the predict method on the model object. Model evaluation: Calculate evaluation metrics such as accuracy, precision, recall, and f1-score using evaluation methods such as scikit-learn's accuracy_score and classification_report.

Results Analysis

Metric analysis: Analyze evaluation results to understand model performance in intrusion detection. Pay attention to accuracy, precision, recall, and f1-score to better understand the model's strengths and weaknesses. Visualization: Use a library such as matplotlib to create visualizations such as a confusion matrix or other graphs that can assist in analyzing results.

In carrying out this experiment, the Python program will be used to implement these steps. Relevant Python libraries or modules, such as pandas for data manipulation, scikit-learn for building and training K-NN models, and matplotlib for visualizing results, can be used to implement this experiment.

RESULTS AND DISCUSSIONS

The experimental results provide information about the performance of the classification model that has been tested using K-Nearest Neighbors (K-NN).

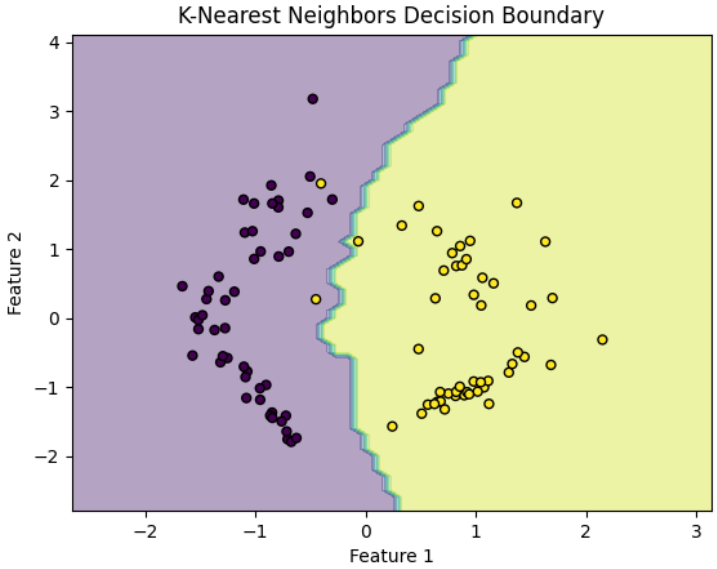

Figure 2. K-NN Decision Boundary

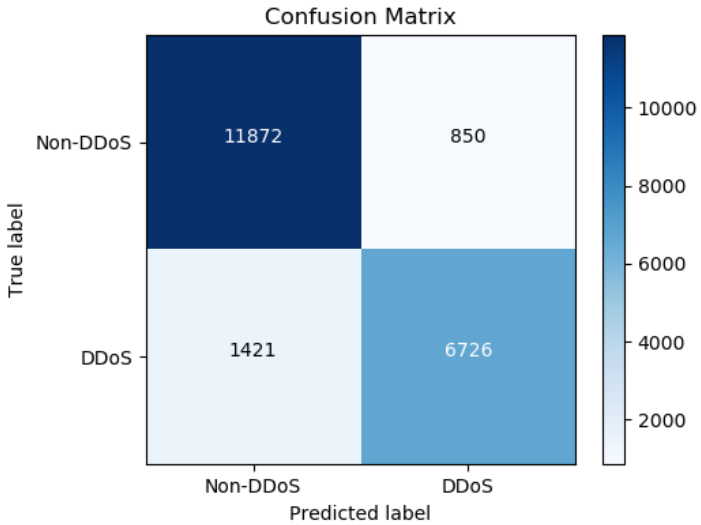

Figure 2. Confusion Matrix

Accuracy: **0.8911783027456993**

Accuracy measures how well the model can predict the entire test data correctly. In this case, the accuracy reaches 0.8911783027456993, which means the model has an accuracy rate of about 89.12%. This shows that the K-Nearest Neighbors (KNN) model performs well predicting the target class.

Confusion Matrix

The confusion matrix shows the number of correct and incorrect predictions for each class. In this case, we have: True Positive (TP): 11872, namely the number of samples that were

actually labeled 1 and correctly predicted as 1. False Positive (FP): 850, namely the number of samples that were actually labeled 0 but were incorrectly predicted as 1. False Negative (FN): 1421, namely the number of samples that were actually labeled 1 but were incorrectly predicted as 0. True Negative (TN): 6726, namely the number of samples that were actually labeled 0 and correctly predicted as 0.

The confusion matrix helps us understand where the model makes mistakes in predicting the target class. In this case, the model tends to be better at predicting class 0 (non-DDoS) than class 1 (DDoS).

Classification Report

The classification report summarizes evaluation metrics such as precision, recall, and f1-score for each class and the metrics' average values (macro average and weighted average). recision measures how good the model is at correctly predicting the class of all positive predictions. Class 0 precision is 0.89, while class 1 precision is 0.89. This shows that the model has good abilities in identifying both classes well. Recall measures how well the model can retrieve (remember) positive classes. Class 0 recall is 0.93, which means the model has a high success rate in recognizing class 0. However, class 1 recall is 0.83, indicating that the model may have little difficulty recognizing class 1. F1-score is the harmonic mean of precision and recall. The F1-score of class 0 is 0.91, while the F1-score of class 1 is 0.86. F1-score provides an overall picture of model performance by considering precision and recall simultaneously. Support shows the number of samples in each class. Overall, the K-NN model performs well with high precision, recall, and f1-score for class 0 (non-DDoS), but slightly lower for class 1 (DDoS).

CONCLUSION

This experiment tests the classification model using the K-Nearest Neighbors (K-NN) algorithm for network intrusion detection. Good Accuracy: The K-NN model tested had an accuracy rate of around 89.12%, which shows that this model can predict whether an event is an intrusion. Confusion Matrix: The confusion matrix provides a detailed picture of model performance, including the number of True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN). This provides insight into the types of errors made by the model. Classification Report: The classification report provides metrics such as precision, recall, and F1-score for each class. The model has good precision and recall values for both classes, showing the ability to identify the most positive (intrusive) and negative (non-intrusive) instances. Summary Weights and Summary Averages: Summary weights and macro averages provide a holistic view of model performance considering the balance of classes in the dataset. In the context of network security, intrusion detection is an important aspect of protecting systems and data from cyber-attacks. The tested K-NN model can be useful in detecting suspicious activity in the network.

References

Alharbi, Y., Alferaidi, A., Yadav, K., Dhiman, G., & Kautish, S. (2021). Denial-of-Service Attack Detection over IPv6 Network Based on KNN Algorithm. *Wireless Communications and Mobile Computing*, *2021*. https://doi.org/10.1155/2021/8000869

Bahta, S. D. (2019). SISTEM PENCEGAHAN GANGGUAN JARINGAN KOMPUTER BERBASIS DETEKSI POLA SNORT. *JURASIK (Jurnal Sistem Informasi Dan Komputer)*, *1*(1), 14–20.

Dong, S., & Sarem, M. (2020). DDoS Attack Detection Method Based on Improved KNN with the Degree of DDoS Attack in Software-Defined Networks. *IEEE Access*, *8*, 5039–5048. https://doi.org/10.1109/ACCESS.2019.2963077

Farradhika Muntaha, M., Hari Trisnawan, P., & Primananda, R. (2019). *Implementasi Intrusion Prevention System (IPS) berbasis Athena untuk Mencegah Serangan DDoS pada Arsitektur Software-Defined Network (SDN)* (Vol. 3, Issue 7). http://j-ptiik.ub.ac.id

Gregorius Hendita. (2022). Sistem Firewall untuk Pencegahan DDOS ATTACK di Masa Pandemi Covid-19. *Journal of Informatics and Advanced Computing* , *3*(1), 52–56. https://journal.univpancasila.ac.id/index.php/jiac/article/view/3853

Hijriyanto, B., & Ulum, F. (2021). Comparison of Mod_Evasive and DDoS Deflate for Slow Post Attack Mitigation. *Jurnal Teknologi Informasi* , *20*(1), 59–68. http://publikasi.dinus.ac.id/index.php/technoc/article/view/4116/2164

Kachavimath, A. V., Nazare, S. V., & Akki, S. S. (2020). Distributed Denial of Service Attack Detection using Naïve Bayes and K-Nearest Neighbor for Network Forensics. *2nd International Conference on Innovative Mechanisms for Industry Applications, ICIMIA 2020 - Conference Proceedings*, 711–717. https://doi.org/10.1109/ICIMIA48430.2020.9074929

Pramana, M., Setyati, E., & Ferdinandus, F. X. (2021). Identifikasi Serangan Denial Of Service (Dos) Di Jaringan Dengan Algoritma Decision Tree C4.5. *Wahana : Tridarma Perguruan Tinggi* , *2*(2), 13–29. http://jurnal.unipasby.ac.id/index.php/whn

Putra, A., & Ariyadi, T. (2019). IMPLEMENTASI PENCEGAHAN TERHADAP SERANGAN FLOODING ATTACK TCP DAN UDP DI KANTOR PDAM TIRTA MUSI PALEMBANG. *JURASIK (Jurnal Sistem Informasi Dan Komputer)*, *1*(1), 14–20. https://ejournal.stmik-tm.ac.id/index.php/jurasik/article/view/6

Siahaan, M. (2021). Mencegah Serangan DDoS (Distributed Denial of Service) Terhadap Email Server Prevent Distributed Denial of Service (DDoS) Attacks Against Email Servers. *SCIENCE TECH: Jurnal Ilmu Pengetahuan Dan Teknologi*.

Xu, Y., Sun, H., Xiang, F., & Sun, Z. (2019). Efficient DDoS Detection Based on K-FKNN in Software Defined Networks. *IEEE Access*, *7*, 160536–160545. https://doi.org/10.1109/ACCESS.2019.2950945