Journal of Artificial Intelligence and Engineering Applications

Website: https://ioinformatic.org/

15th February 2026. Vol. 5. No. 2; e-ISSN: 2808-4519

Streamlit Based Network Intrusion Detection System Prototype with Machine Learning Algorithm

Tiara Maulida[1]*, Muhammad Nandi Buchari[2], Teofilus Tirta Jumata[3], Putra Pratama Syahrival[4], Ali Mustopa[5]

[1,2,3,4,5] *Universitas Bina Sarana Informatika*
tiaramaulida703@gmail.com[1*], nandibuchari408@gmail.com[2], teofilus819@gmail.com[3], upuc655@gmail.com[4], ali.aop@bsi.ac.id[5]

Abstract

Computer network security has become a crucial elemen in the digital era, with the increasing risk of attacks that could potentially disrupt systems and access critical data. An Intrusion Detection System (IDS) powered by Machine Learning is one effective way to automatically detect suspicious network activity. This study aims to create a prototype of a network Intrusion Detection System using Streamlit that applies Machine Learning algorithms, including Naïve Bayes and Random Forest, to classify normal network activity as an attack. The method used in this study is a quantitative approach with an experimental design utilizing a public dataset of labeled network traffic. The research process includes the stages of initial data processing, feature selection, model creation, performance evaluation, and implementation of the Streamlit interface. Test results show that the Naïve Bayes algorithm has the best performance, with an accuracy level reaching 0.8000, an error rate of 0.2000, and an F1 Score of 0.7273. Random Forest recorded an accuracy level of 0.7333, an error rate of 0.2667, and a lower F1 Score of 0.3333. These findings demonstrate that Naïve Bayes is more effective at detecting intrusions and recognizing anomalous network traffic patterns. The Streamlit based system implementation successfully provides an interactive and userfriendly interface, allowing users to perform analysis and understand classification result without in-depth technical expertise. Given the foregoing, the network intrusion detection system prototype built with Streamlit and a Machine Learning algorithm is considered suitable as a simple, informative, interactive, and efficient network security support tool. This research paves the way for future developments, such as the implementation of Deep Learning models and the integration of live network monitoring.

Keywords: *Intrusion Detection, Machine Learning, Random Forest, Naïve Bayes, Streamlit*

1. Introduction

Network security is a crucial aspect for computer users, given the importance of safety and comfort when using technology in everday life. The field of computer network security continues to evolve and improve, as does the criminal technology used by digital criminal. Some criminal technologies even surpass the capabilities of computer expert, making network security difficult to guarantee. As the processes of computer crime become increasingly difficult to understand, effective preventative measures for computer network security are crucial to reduce the potential for cybercrime [1].One way to protect a computer is by implementing Intrusion Detection System (IDS) technology. An IDS serves as an early warning system to detect attacks on a computer network. This system will notify network administrators if there is a threat to the computer network. In addition, an Intrusion Detection System (IDS) also records all attempts and activities that have the potential to disrupt the network, as well as other attacks that may occur. The purpose of this research is to apply an IDS to a network system and analyze the IDS logs to understand the types and categories of attacks on a computer network. In-depth analysis of the IDS logs will be conducted to improve computer network security. Network administrators can evaluate or improve the logs stored by the Intrusion Detection System (IDS). The results of the Intrusion Detection System (IDS) log analysis can be used to identify various types of attacks aimed at a computer network, allowing network administrators to make improvements, reconfigure the network, and implement certain applications to improve the security of the managed network [2].

This research aims to create a prototype of a web-based network intrusion detection system using Streamlit. This system can classify network activity into normal or intrusion using Machine Learning algorithms. The application of Machine Learning algorithms in the system to detect intrusions provides a deeper understanding of the effectiveness of simple probabilistic classification methods in identifying anomalies in network traffic. The benefits of this research can produce a prototype using Streamlit that makes it easier for users without a technical background to view detection result and make quick decisions for mitigation steps or prevent, reduce and control networks from attacks so that computers remain safe and stable [3]. The scope of this research focuses on results that can provide scientific comparisons, limited to the creation and evaluation of a prototype web-based intrusion detection system that uses Streamlit and Machine Learning algorithms for binary classification of normal or having an attack (intrusion), by utilizing a dataset uploaded by users. This research does not include production-scale implementation, real-time network monitoring on large organizational infrastructures or comprehensive comparisons with all modern algorithms such as Deep Learning so the results are more prototypical and experimental [4]. This research

uses a quantitave approach with a prototype development experimental design to build and test a webbased network intrusion detection system and Machine Learning. This design follows current Indonesian journal practices in the field of IDS and Machine Learning. As an illustration, this research implements Machine Learning for intrusion detection on computer networks using a supervised model and evaluating the algorithms performance with quantitative metrics.

1.1. Data Preparation

Secondary data collection is carried out by downloading public datasets, for example, labeled network traffic datasets that have been used in recent research in Indonesia related to Intrusion Detection Systems (IDS) using Machine Learning. Once the dataset is received, the next step is data cleaning and selection, namely the process of cleaning the data to produce missing values and selecting relevant features needed. The data is then divided into training and testing subsets according to a predetermined ratio, where the error result are displayed.

Table. 1: Data Preparation

No	Attribute Name	Description
1	Duration	The length of the connection period is measured in seconds.
2	Protocol_type	The type of network protocol used in this connection, such as: TCP, UDP, ICMP.
3	Service	Networks accessed through this connection, for example: http, ftp, smtp, telnet, and others.
4	Flag	The status of a TCP connection that shows the final result or condition of the connection, for example, SF (normal connection), REJ (rejected), RSTO and so on.
5	Src_bytes	Total bytes sent from source (client) to destination (server).
6	Dst_bytes	The number of bytes sent from the destination (server) back to the source (client).
7	Land	Indicates whether connections originate and are destined for the same host/IP. 0 = no (normal) 1 = yes (often a sign of a DoS attack)
8	Wrong_fragment	Incorrect or invalid packet fragment count. Incorrect fragments often appear in fragmentation-based attacks.
9	Urgent	The number of packets that have an urgent pointer (a TCP feature for urgent data). Typically 0 for normal connections.
10	Hot	The number of “suspicious” events in a connection, such as excessive system file access, unusual logins, or spesific commands executed. Often used to detect system access-based attacks.

1.2. Data Mining Modeling

The research process includes data preprocessing steps, such as removingg irrelevant attributes, encoding categorical variables, and adjusting numerical features, so that the data is more ready for use in model training. This process includes data cleaning, transformation, and normalization to remove noise, address missing values, and ensure relevant feature are ready for the intrusion classification process (such as distinguishing attack types such as DoS, Probe, or Normal). Categorical Data Encoding IDS datasets often contain categorical attributes such as “protocol_type” (tcp/udp/icmp) or “flag” (SF/S0). The label encoding method converts categories into numbers (tcp=0, udp=1). One Hot Encoding provides a binary column for each category of the tcp, udp, and icmp columns with values 0/1. Normalization of numeric attributes such as src_bytes or dst_bytes has different values ranges, which can affect model performance.

1.2.1. Algorithm 1: Naïve Bayes

Naive Bayes is a simple, probabilistic classification method that calculates a set of probabilities by summing the frequencies and combinations of values from a given dataset. This algorithm uses Bayes’Theorem and assumes all attributes are independent, or independent of each other, based on their values in the variable classes. Naïve Bayes is based on the simplifying assumption that attribute values are conditionally independent when given an output value [5].

1.2.2. Comparison Algorithm: Random Forest

Random Forest is an innovation of the Decision Tree method that combines several Decision Trees, where each Decision Tree is trained using individual samples and each attribute is separated into trees selected from a random subset of attributes. Random Forest has several advantages, including its ability to improve accuracy even when there is missing data [6].

1.3. Evaluation

A system evaluation was conducted to assess how well the network intrusion detection system prototype using Streamlit can efficiently recognize and classify network activity. In this phase, the Naïve Bayes algorithm was applied as a classification method to distinguish between normal network traffic and those showing signs of intrusion. The evaluation procedure involved testing pre-processed network data to measure system performance based on various parameters, such as Route Error, F1 Score, and Accuracy.

1.3.1. Route Error

The error rate, or route error in Machine Learning is an indicator of the number of incorrect predictions compared to the total number of samples analyzed. In other words, it is the opposite of accuracy or route error= = 1 – accuracy, and in research practice, this study relies solely on the route error value [7].

The following is the calculation formula for route error:

$$\text{Error Rate} = \frac{FP + FN}{TP + TN + FP + FN} = 1 - \text{Accuracy}.$$

To measure how big the error is in data prediction, the error rate is calculated by paying attention to the amount of data that is predicted incorrectly, the total of all prediction errors added up and divided by the total number of data will give the error rate value [8].

1.3.2. F1-Score

The F1-Score is a measure frequently used to assess segmentation result. It represents the harmonic average between the precision and recall of the segmentation system being analyzed. The F1-Score is used in this study to provide a more comprehensive measure of model performance [9].

Here's the F1-Score formula calculation:

*F*1,-. /0=21/0-232. 4×70-8991/0-232. 4:70-899

1.3.3. Accuracy

Accuracy is the simplest and most frequently used assessment measure in Machine Learning classification, measuring the number of correct predictions by a model compared to the total data tested. It is calculated by dividing the number of correct predictions (True Positive and True Negative) by the total number of predictions (TP + TN + FP + FN) [10].

The following is the Accuracy formula:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

1.4. Streamlit Application Design and Construction

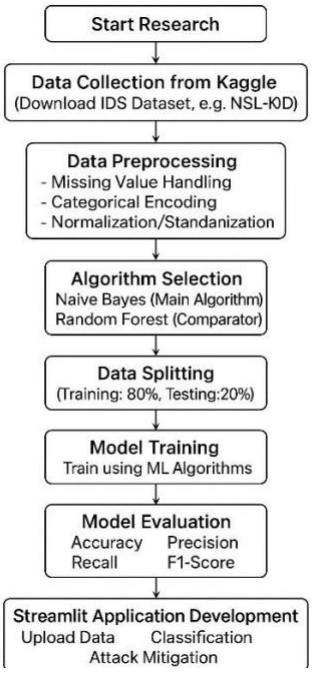

Fig.1: Research Stage Flowchart

1.4.1. System Architecture

A network security method that aims to create an integrated security system architecture between an Intrusion Detection System (IDS), Firewall System, Database System, and Monitoring System linked to a mobile agent review. This security system aims to protect the network with the ability to respond according to security policies. The resulting architecture of a computer network intrusion detection

system that has the ability to detect suspicious network activity, and take further countermeasures against attacks based on mobile agents [11].
In systems designed to detect intrusions, the initial stage involves a script run on platforms like Google Colab, where Machine Learning models are developed using datasets like Kaggle, including preprocessing steps, feature selection, and testing algorithm like Naïve Bayes and Random Forest. After finding the most effective model, it is saved in the .pkl file format (using the pickle library) so that the entire training process does not need to be repeated during inference. At this stage, a Streamlit-based application functions as both a user interface and a live inference engine. When a user uploads network connection data through the Streamlit interface, the application loads the saved pickle model. Next, the application calls a prediction function like a model. Predict to determine whether the incoming data falls into the normal connection category pr an attack. The classification results are then displayed interactively in the Streamlit interface, where users can view the results in the form of statuses such as Attack/Normal, probability percentages, and appropriate mitigation recommendations based on the type of attack successfully detected.

2. Results and Discussion

This section present the evaluation results of a prototype system for detecting network intrusions created using the Naïve Bayes and Random Forest algorithms. The test results are presented through analysis of error rates, model performance, and data visualization to observe the effectiveness of each algorithm in classifying network activity as normal or potentially an attack. The presented findings serve as the basis for a discussion to assess how well the model can recognize intrusion patterns, while also evaluating the role of the Streamlit application as an interface to present prediction results interactively and easily understood by users.

2.1. Data Mining Model Analysis Results

Based on the results of trials using the Naïve Bayes and Random Forest algorithms, a comparison of model performance was obtained based on the assessment matrix, namely Accuracy, Error Rate, and F1-Score. The analysis showed that the Naïve Bayes model performed better than Random Forest.

Fig. 2: Key Model/Data Visualization

Table. 2: Visualization of Error Rate Result from Both Algorithms

Model	Accuracy	Error Rate	F1-Score (Attack)
Naive Bayes	0.8000	0.2000	0.7273
Random Forest	0.7333	0.2667	0.3333

The following results show a comparison of the performance of the two Naïve Bayes and Random Forest algorithms, based on three evaluation metrics: Accuracy, Error Rate, and F1-Score. The table shows that Naïve Bayes achieved the highest accuracy value, which is 0.8000, indicating that this model can make correct predictions up to 80% of all test data. The error rate on Naïve Bayes was recorded at 0.2000, indicating that only 20% were listed as incorrect predictions. In addition, the F1-Score value of Naïve Bayes reached 0.7273, making this model quite effective in detecting attack classes. Meanwhile, Random Forest recorded an accuracy value of 0.7333, which is equivalent to approximately 73,33%, and an error rate of 0.2667, meaning there was a prediction error of almost 26.67%. the F1-Score value for Random Forest only reached 0.3333, indicating that this model is less effective in distinguishing the positive class.

2.2. Model Visualization and Key Features

At this stage, visual modeling is applied to show the model's structure and the key elements that play a significant role in the classification process. The image below shows a visualization of the error comparison during classification.

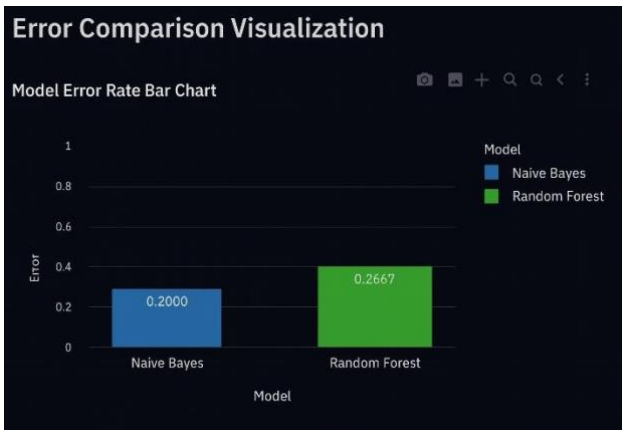

Fig. 3: Key Model/Data Visualization

Presents a comparison of error rates between two classification approaches, Naïve Bayes and Random Forest. This chart supports the findings outlined previously, showing that the Naïve Bayes method has the lowest error rate with a result of 0.2000. This lower error rate strengthens the decision to choose Naïve Bayes as the most efficient model for the intrusion detection task, as it consistently produces more accurate predictions than Random Forest which has a value of 0.2667.

2.3. Streamlit Application Implementation

The Stramlit application functions as an interactive platform that assists users in directly performing classification. The image below shows how the model is implemented in the Streamlit application.

Fig. 4: Streamlit Application Input Interface (UI) Display

Table. 3: Input Connection Parameters

Input Parameters	Explanation
Protocol Type	Displays the various types of communication protocols used in a network, such as TCP, UDP, or ICMP. These protocols govern the methods of sending, receiving, and managing information between devices on the network.
Service	It presents various types of network services used in the connection, such as HTTP, FTP, SMTP, or courier. Each service has a specific function in the process of sending information over the network.
Flag	These are signals or states within data packets that are intended to control the flow of communication. For example, SYN, a request to join, ACK, a signal that information has been received, FIN, a signal to close the connection, and REJ, a signal to reject the connection request.

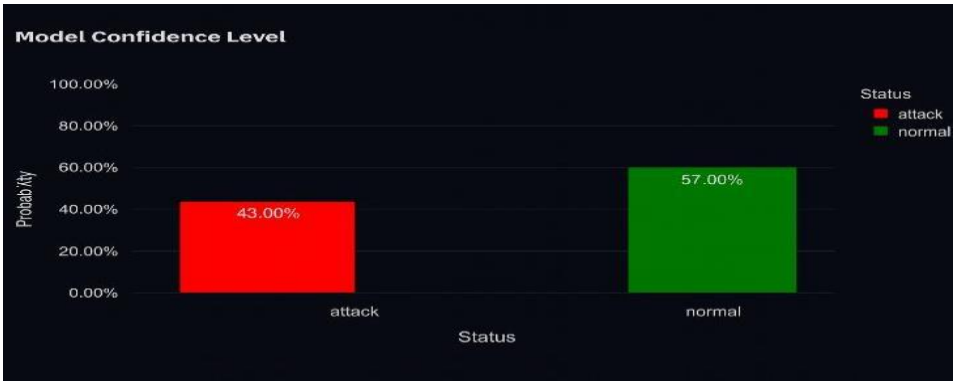

Fig. 5: Display of Prediction/Analysis Result in the Streamlit Application

This view illustrates the results of the new data grouping. The Naïve Bayes model assesses the correlation as normal with a confidence level or probability of 57.00%, slightly higher than the probability og being considered an attack, which is only 43.00%. The bar chart shows the preponderance of the normal option in green, confirming the categorization or separation results by type, as seen in the success notification at the bottom of the chart.

3. Conclusion

This research successfully created a prototype of a Streamlit-based network intrusion detection system with the application of Machine Learning algorithms to improve computer network security. Through a series of experiments conducted using public datasets regarding network traffic, this system can automatically detect suspicious activity. The test results identified that the Naïve Bayes algorithm has a lower error rate when compared to the Random Forest algorithm, so it can be concluded that this algorithm is more suitable for use in classifying network activity in the developed system. The use of Streamlit as a web interface makes is easy for users to monitor and understand the detection results without the need for in-depth technical knowledge. Overall, the system developed can help in making quick decisions to reduce the risk of network attacks. For further development, this research can focus on the application of Deep Learning models and real-time testing so that the system can adapt to new attack patterns and provide a better level of accuracy.

Acknowledgement

The author would like to express his gratitude to Bina Sarana Informatika University for providing the facilities and opportunity to conduct this study, he also appreciates his supervisors who provided valuable guidance, direction, and input throughout the development of this research. He also thanks his colleagues who assisted in data collection and system testing, enable the successful completion of this research. The support and cooperation of various parties significantly contibuted to the success of this research.

References

[1] A. Saleh, "Implementasi Metode Klasifikasi Naïve Bayes Dalam Memprediksi Besarnya Penggunaan Listrik Rumah Tangga," *Creat. Inf. Technol. J.*, vol. 2, no. 3, pp. 207–217, 2015.

[2] A. D. Afifaturahman and F. MSN, "Perbandingan Algoritma K-Nearest Neighbour (KNN) dan Naive Bayes pada Intrusion Detection System (IDS)," *Innov. Res. Informatics*, vol. 3, no. 1, pp. 17–25, 2021, doi: 10.37058/innovatics.v3i1.2852.

[3] F. Ardiyansyah, K. Setiawan, and N. Sutisna, "Implementation of IDS on Computer Networks Using Snort Based on Telegram Chatbot Implementasi IDS pada Jaringan Komputer Menggunakan Snort Berbasis Chatbot Telegram," *MALCOM Indones. J. Mach. Learn. Comput. Sci.*, vol. 4, no. October, pp. 1614–1623, 2024.

[4] S. Anwar, F. Septian, and R. D. Septiana, "Klasifikasi Anomali Intrusion Detection System (IDS) Menggunakan Algoritma Naïve Bayes Classifier dan Correlation-Based Feature Selection," vol. 2, no. 4, pp. 135–140, 2019.

[5] D. D. Putri, G. F. Nama, and W. E. Sulistiono, "ANALISIS SENTIMEN KINERJA DEWAN PERWAKILAN RAKYAT (DPR) PADA TWITTER MENGGUNAKAN METODE NAIVE BAYES CLASSIFIER," vol. 10, no. 1, pp. 34–40, 2022.

[6] D. A. Larasati and M. Hemansyah, "Deteksi Anomali dalam Sistem Keamanan Jaringan Menggunakan Teknik Supervised Machine Learning," vol. 9, no. 1, pp. 65–69, 2025, doi: 10.55886/infokom.v9i1.971.

[7] D. I. Kabupaten and D. Tahun, "1 , 2 , 3," vol. 3, pp. 831–838, 2014.

[8] Y. Ilanda, D. Vionanda, Y. Kurniawati, and D. Fitria, "Comparison of Error Rate Prediction Methods of C4 . 5 Algorithm for Imbalanced Data," vol. 1, no. 2006, pp. 240–247, 2023.

[9] A. Bachtiar, R. Firliana, and D. Harini, "Rancang Bangun Platform E – Commerce Pada TUTUS BETTA FARM," vol. 7, no. 2, pp. 96–102, 2024.

[10] A. Lowell, A. Lowell, K. Candra, and E. Indra, "Perbandingan Metode Support Vector Machine (SVM) Dan Naive Bayes Pada Analisis Sentimen Ulasan Aplikasi OVO JURNAL MEDIA INFORMATIKA [JUMIN]," vol. 6, no. 2, pp. 896–905, 2025.

[11] E. Manalu, F. A. Sianturi, and M. R. Manalu, "PENERAPAN ALGORITMA NAIVE BAYES UNTUK MEMPREDIKSI JUMLAH PRODUKSI BARANG BERDASARKAN DATA PERSEDIAAN DAN JUMLAH PEMESANAN PADA CV . PAPADAN MAMA PASTRIES," vol. 1, no. 2, 2017.