Optimizing Neural Network Architecture for Detecting DDOS Attacks using ANN and XGBoost in Imbalanced Networks

Rissal Efendi
Department of Information Technology, Satya Wacana Christian University, Salatiga, Indonesia
rissal.efendi@uksw.edu (corresponding author)

Received: 11 December 2024 | Revised: 12 January 2025, 4 February 2025, and 15 February 2025 | Accepted: 21 February 2025

 DOI: https://doi.org/10.48084/etasr.9909

ABSTRACT

The advancement of Internet technology and digital transformation is always followed by increased security concerns in computer networks. Attacks can disrupt services connected to LANs and the Internet, particularly those targeting web-based applications. The most common threat in HTTP is Distributed Denial of Service (DDoS) attacks. Network security is critical to preserve the integrity and availability of services, and it is a critical necessity to have effective methods for detecting and mitigating such attacks to address these risks. Machine learning techniques, particularly ANN and XGBoost, play a key role in enhancing the ability to identify unusual patterns. Despite that, challenges remain in fine-tuning these models for accurate and efficient detection, especially when working with imbalanced data. This study proposes an integrated model that combines ANN with XGBoost to improve detection performance. In the first phase, the ANN architecture is customized to distinguish normal traffic from attacks, while in the second phase, XGBoost is used to refine predictions and improve accuracy. The evaluation results show that the DBSCAN-SMOTE-ANN-XGBoost-PSO model outperforms others, with high accuracy (96.83%), sensitivity (93.23%), and precision (96.13%), demonstrating its effectiveness in detecting DDOS attacks while reducing both false positives and negatives. This integrated approach offers an optimal solution to improve network security and address evolving DDOS attack patterns.

Keywords-DDOS; ANN; XGBoost; anomaly detection; imbalaced networks

I. INTRODUCTION

Digital transformation has encouraged a deeper exchange of data and services that increases the risk of cyberattacks, especially those focused on web-based applications and services. One of the most frequent and severe cyberattacks is via HTTP [1]. Among the most common and high-risk threats are HTTP attacks, such as Distributed Denial of Service (DDoS) [2, 3], SQL injection [4, 5], and Cross-Site Scripting (XSS) [6, 7]. Therefore, it is essential to detect and mitigate HTTP attacks to ensure the security and accessibility of online services [8].

There are various types of research to detect anomalies in network traffic, which are very important to prevent and overcome attacks while maintaining the security of data exchange [9-11]. Intrusion Detection System (IDS) technology detects and blocks behavior that can undermine system security, such as collecting vulnerability data, blocking access, and proactively identifying unusual traffic. These systems continuously monitor and analyze events within a computer system or network to ensure network access control and maintain the reliability and integrity of data. Machine learning is frequently employed in network security systems to learn data exchange patterns and address decision-making challenges such as prediction and classification [12]. This technology employs past data on network traffic to analyze traffic behavior, supporting network administrators in classifying traffic and applying suitable defensive strategies to counter malicious attacks. Rule-driven traffic analysis has drawbacks that can be mitigated by machine learning classification techniques. Many studies have explored machine-learning approaches to detect abnormal network data flows [13-16]. With the fast-paced improvement of network technologies, the growth in network traffic has created challenges for machine learning, particularly in analyzing complex nonlinear data relationships. Traditional methods show many limitations that are becoming increasingly evident [17]. Therefore, with the growing amount of data, learning accurate traffic characteristics to enhance the accuracy of detecting unknown attacks is increasingly difficult [16, 18].

As a solution to this problem, deep learning methods such as Convolutional Neural Networks (CNN) [19], Recurrent Neural Networks (RNN) [20], and Generative Adversarial Networks (GAN) [21] are used to identify anomalies in network traffic. These methods have resulted in efficient and precise attack detection classifiers, which offer more reliable security measures for networks and further improve the precision of network detection. However, certain deep learning methods are overly dependent on data when detecting network

attacks or require the application of various models. This can cause delays in detection, thus reducing the efficiency of IDSs. In addition, there is a need for models that can generalize well across different types of attacks, especially as new attack forms emerge. This requires the exploration of robust architectures that can adapt to evolving attack patterns.

Recent studies have shown that multiple techniques can increase the effectiveness of DDoS detection [22-26]. For example, in [27], Clustering-Based Local Outlier Factor (CBLOF) was combined with eXtreme Gradient Boosting (XGBoost), showing improved results in detecting DDoS attacks than traditional methods. CBLOF was used to detect anomalies in network traffic patterns, while XGBoost was used to classify them into attacks or normal traffic. The combination of these two techniques provides advantages in the context of detection effectiveness and the potential to identify more complex and unexpected attacks. Another noteworthy approach is the use of DBSCAN (Density-Based Spatial Clustering of Applications with Noise) and SMOTE (Synthetic Minority Oversampling Technique) combined with Long Short-Term Memory (LSTM) [28]. This method focuses on managing class imbalance in DDoS detection datasets, which is often a major problem in machine learning models. By applying SMOTE and DBSCAN, the class distribution in the dataset is improved, facilitating the LSTM model to learn better at detecting time-based anomalies that occur during DDoS attacks. This approach further strengthens the model's proficiency in identifying attacks in highly heterogeneous networks.

ANNs have made significant strides in recognizing complex patterns in network traffic data [29]. However, optimizing ANN architectures to detect DDOS attacks remains a matter of improvement, as the preferred selection of hyperparameters, such as the number of layers, neurons, and the activation mechanism, has a major effect on model performance. Therefore, integration of ANN with other models is needed. One model that has the potential to improve detection capabilities is XGBoost [30, 31], which is acknowledged for its effectiveness and accuracy in handling structured data and imbalanced datasets [32]. However, the optimal integration of these models for the detection of DDOS attacks has not been thoroughly explored. Integration of XGBoost with neural network-based models for DDOS attack detection has received less attention. The existing literature often treats these models as standalone solutions, without exploring the synergies that can arise from combining the capabilities of ANNs with the power of XGBoost. As a result, there is a crucial gap in the potential benefits of hybridizing these two approaches to improve both detection accuracy and computational efficiency. This study addresses this gap by proposing a novel hybrid ANN-XGBoost framework, aiming to provide a balanced solution to the challenges of high detection accuracy and computational efficiency in DDOS attack detection. The proposed solution aims to leverage the strengths of both models, where ANN captures complex nonlinear patterns in DDOS traffic and XGBoost fine-tunes the predictions to improve accuracy, especially in scenarios with imbalanced data.

In the first phase, the goal is to fine-tune the ANN architecture to effectively distinguish normal HTTP traffic from attacks. The process involves adjusting key parameters such as hidden layers, neurons, activation functions, and learning rates. Techniques such as Bayesian optimization or grid search can be utilized to find the best settings, ensuring that the model balances accuracy with performance. In the second phase, the output of the optimized ANN is passed to the XGBoost model, which helps refine the predictions by leveraging its strength in handling imbalanced data. XGBoost fine-tunes the final classification by focusing on misclassified examples from the ANN stage, thereby reducing false negatives and improving overall detection accuracy. This combined model is expected to provide more reliable results than using independently each model. By optimizing ANN to capture complex patterns in DDOS traffic and using XGBoost to fine-tune the output, this solution aims to achieve high detection rates with low false positives and negatives.

The DBSCAN algorithm is employed to identify anomalies in traffic data. DBSCAN helps distinguish outliers that might suggest DDoS attacks, enhancing overall detection accuracy. SMOTE is used to address the issue of class imbalance in the dataset. By generating synthetic samples for the minority class, SMOTE improves the model's ability to detect rare DDoS attack instances, thus improving detection performance. A method is proposed to optimize the ANN architecture to effectively distinguish between normal HTTP traffic and DDoS attacks, with a focus on hyperparameter tuning. The output of the optimized ANN is integrated into XGBoost to refine the detection process, particularly in the handling of imbalanced datasets. Finally, the effectiveness of the developed integrated model was evaluated and contrasted with conventional methods, demonstrating its superior accuracy and efficiency.

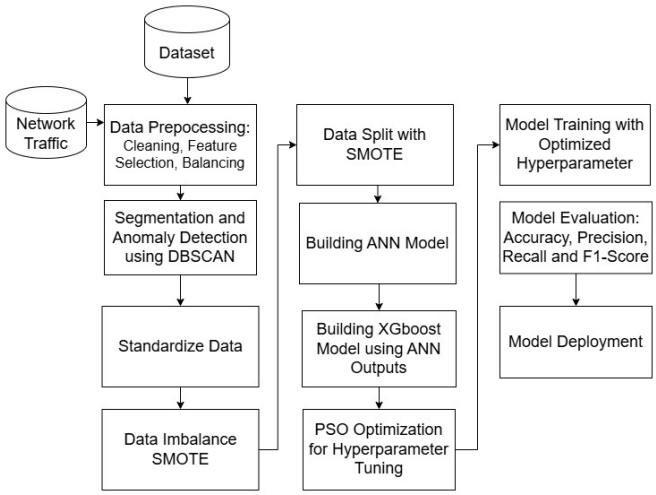

Fig. 1. Research method.

II. RESEARCH METHOD

Figure 1 shows a structured pipeline designed to process network traffic data, detect anomalies, and perform classification using a machine learning model. A detailed description of each stage of the flowchart follows.

A. Traffic Input and Dataset Collection

The process begins with the input of raw network traffic data into the system. The data are stored in a dataset that went through a series of transformations to make it ready for use by machine learning models. Network traffic data typically includes various metrics and characteristics, such as packet size, duration, and origin/destination information. Table II provides a summary of the key variables utilized in this study to describe network traffic during a communication session. Each variable plays a critical role in identifying patterns related to DDOS attacks, especially in the context of packet-level data. Table II represents the details of these variables, such as the total number of packets, byte size, protocol type, and connection status, which are critical to analyzing and detecting potential security threats.

This dataset was collected using Wireshark during a penetration test conducted at the Department of Information Technology, Satya Wacana Christian University, Indonesia, with a DDoS attack using the HTTP protocol launched through Metasploit. The entire penetration test lasted 4 days, with the DDoS attack only taking two hours to generate packet data. A total of 1,048,598 packets were collected, consisting of 978,279 benign packets (93.3%) and 70,319 malicious packets (6.7%). This dataset is imbalanced, with benign packets far outnumbering malicious packets, providing a realistic picture of real-world conditions.

TABLE I. PACKET FEATURES DETAILS

Variables	Variable details
pkts	Number of packets sent during a communication session or data stream.
byte	Number of bytes (data size) transferred in a communication session or data stream.
rate (packet rate)	Number of packets sent per second in a session or data stream.
proto (Protocol)	Type of network protocol used in communication, for example, TCP, UDP, or ICMP.
flgs (Flags)	An indicator or bit in the TCP protocol that indicates the status or control of the packet, such as SYN, ACK, RST, FIN.
state (Connection State)	The status or condition of the connection at a certain time, for example, "ESTABLISHED", "CLOSED", or "SYN_SENT"
saddr (Source Address)	The source or origin IP address from which the packet is sent
daddr (Destination Address)	The destination IP address to which the packet is sent
dport (Destination Port)	The destination port on the receiving device used in communication, for example, port 80 for HTTP or 443 for HTTPS.

B. Preprocessing

During the preprocessing phase, raw traffic data were processed and organized. This includes tasks such as handling missing values, removing irrelevant information, converting data formats, and eliminating unnecessary or redundant data. The objective of this phase was to ensure that the data is consistent and reliable for the next stages.

C. DBSCAN Anomaly Detection

After the preprocessing stage, the data was processed using an anomaly detection algorithm based on DBSCAN. DBSCAN is an unsupervised clustering method that is often used to identify outliers or anomalies in data [33]. By clustering data based on density, DBSCAN helps isolate suspicious or unusual traffic patterns that may indicate attacks or abnormal activity. Figure 2 shows the process of the DBSCAN algorithm to detect clusters and anomalies in the network traffic data.

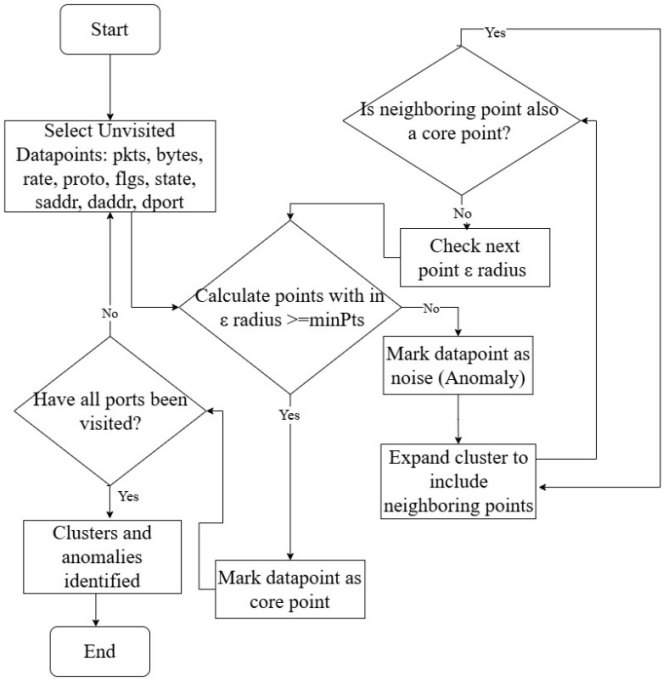

Fig. 2. DBSCAN clustering workflow with anomaly detection using network traffic variables.

The process begins by selecting an unvisited data point that includes attributes such as packets, bytes, rate, protocol, flags, status, source address, destination address, and destination port. The algorithm then calculates the number of neighboring points within a certain radius (ε). If the number of neighboring points meets or exceeds the minimum value (*minPts*), then the data point is considered a core point. This core point is then expanded by adding its neighboring points to the cluster. If a point does not have enough neighbors, it is marked as noise or anomaly. This process continues until all data points have been visited. The final result is the identification of clusters for similar data and the marking of anomalies that potentially indicate unusual or suspicious patterns.

D. Data Standardization

Once anomalies are detected, the data go through a standardization process as a preparation step for further analysis. The data standardization process generally involves transforming features to have a mean of 0 and a standard deviation of 1. This process can be performed with:

$$Z = \frac{X - \mu}{\sigma} \quad (1)$$

where X is the initial value, μ is the average value of the feature, and σ is the standard deviation. This equation ensures that all features have a balanced contribution to the analysis, which is very useful in clustering, as it prevents features with larger ranges from dominating the distance calculation.

E. Dataset Split

Before building and testing a model, it is important to split the dataset carefully so that the training and evaluation processes can be carried out properly. A 70:30 was chosen to divide the dataset between training and testing to ensure a balance between sufficient data for accurate model training and evaluation. 70% of the training data allows the model to learn existing patterns, while 30% of the testing data ensures a valid evaluation with previously unseen data. This ratio avoids overfitting and ensures that the model can generalize well to new and representative data.

F. Handling Data Imbalance with SMOTE

The dataset contains two main groups, benign and malicious. However, there was an imbalance, with far fewer malicious samples compared to benign ones. SMOTE was used to address this, which helps create new malicious samples by generating variations of existing ones. For example, SMOTE can modify features such as the number of HTTP packets, HTTP ports, and external IP addresses to generate new examples. This helps balance the dataset, ensuring that the model can learn effectively from both classes. As a result, the model becomes better at detecting and identifying malicious activities or unusual behavior in network security. To deal with the imbalance, two key steps were followed: First, the minority class (malicious) was identified, and then samples from this class were selected to generate new synthetic data using SMOTE. SMOTE aims to address data imbalance by adding synthetic data to the minority class (e.g., HTTP attack data). These synthetic data are created by interpolating existing minority data points and their neighbors using the k-Nearest Neighbors (k-NN) algorithm [34]. Synthetic data are created using:

$$x_{new} = x_i + \left(x_{neighbor} - x_i\right) * \delta \quad (2)$$

where X_i denotes the original sample from the minority class (e.g., selected malicious examples), $X_{neighbor}$ denotes the selected neighbors from the minority class, and δ is a number between 0 and 1 to set how far a new synthetic sample will be created from the original example.

Equation (2) describes how to update the value of X_i towards the value of $X_{neighbor}$ based on the parameter δ that controls how much of a change is made. Here, X_i is the initial value, $X_{neighbor}$ is the neighbor or reference value to be approximated, and δ is a multiplier that determines how much the difference between x_i and $X_{neighbor}$ contributes to the update of x_{new}. The larger δ is, the larger the change made to X_i to approach $X_{neighbor}$.

G. Building an ANN Model

ANNs are powerful tools for classification built on the concept of artificial neurons. These neurons are designed to mimic the way biological neurons work and serve as the basic elements of the network. In this study, the ANN architecture created consists of 11 input units and a single hidden layer containing 10 neurons. The number of inputs and the number of neurons in the hidden layer can change depending on the feature selection method used in the model. This ANN model learns from the features and patterns in the training data to classify between normal and anomalous traffic. The proposed ANN architecture comprises three layers: an input layer, a single hidden layer, and an output layer. The input layer contains 11 units, each representing different network traffic features such as byte, rate, proto, flgs, state, saddr, daddr, dport, and other relevant variables. These input units are fully connected to the hidden layer, which consists of 10 neurons. The hidden layer processes the input data and is then connected to the output layer, which has a single unit representing the result, either "Attack" or "No Attack." This architecture forms a basic structure for detecting DDOS attacks, where each input is passed through the hidden layer, and the output is classified based on the learned weights of the neural network.

H. Building an XGBoost Model Using ANN Output

After training the ANN model, the output of the ANN is input to the XGBoost model. XGBoost is a gradient-boosting algorithm that is well known for its speed and high performance, especially in tabular data. XGBoost is used to further refine the predictions by exploiting the patterns identified by the ANN model. The combination of ANN and XGBoost can create a more robust classifier with higher predictive power.

The XGBoost classifier model was trained using the core parameters that define its structure and behavior. With 10 decision trees (n_estimators = 10), XGBoost builds a series of trees iteratively, where each tree aims to reduce the prediction error of the previous. The maximum tree depth was set to 3 (max_depth = 3), limiting the number of splits to keep the model complexity low and reduce the risk of overfitting. The learning rate was set at 0.1 (learning_rate = 0.1), which controls how much each tree contributes to the final prediction, allowing the model to learn more gradually and achieve better optimization.

I. PSO Optimization

PSO was applied to optimize the performance of the ANN and XGBoost models. PSO was used to tune the hyperparameters of the ANN and XGBoost models, finding optimal settings that maximize the accuracy and performance of the model on the dataset. Every particle within the PSO model represents a potential solution consisting of the ANN and XGBoost parameters, such as weights and biases. This process begins by initializing the particles randomly in the specified search space. This random initialization allows PSO to explore various possible XGBoost parameter configurations in an attempt to find the optimal convergence to the expected solution. PSO involved the following steps.

1) Velocity Update

Equation (3) is used in PSO to update the position and velocity of particles in the search space. In this context, this equation is used to update the parameters or weights of a neural network model (such as ANN or XGBoost) based on the

particle's personal best position $p_{i,best}$ and the global best position g_{best}. $v_i(t)$ is the particle velocity at iteration t, while $x_i(t)$ is the position or value of the parameter being searched for. Factors c_1 and c_2 are coefficients that control how much influence the personal and global best positions have on the update, and r_i and r_2 are random numbers to enhance exploration. Using this equation, PSO can find the optimal architecture or parameters for detecting DDOS attacks, even though facing class imbalance problems.

$$v_i(t+1) = w.v_i(t) + c_1.r_i\left(p_{i,best} - x_i(t)\right) + \\ +c_2.r_2(g_{best} - x_i(t)) \quad (3)$$

where w is the inertia weight, c_1 and c_2 are cognitive and social coefficients, r_i and r_2 are random numbers between 0 and 1, $p_{i,best}$ is the best position found by the particle i, and g_{best} is the best position found by the entire swarm.

2) Position Update

After the particle velocity $v_i(t+1)$ is updated using (3), the new position of the particle $x_i(t+1)$ is calculated by adding the old position $x_i(t)$ with the new velocity $v_i(t+1)$. Mathematically, this is expressed as $x_i(t+1) = x_i(t) + v_i(t+1)$, where $x_i(t)$ are the weights or parameters of the model at iteration t and $v_i(t+1)$ are the changes applied to those weights. This position update will move the model parameters closer to the optimal solution which can enhance the model's ability to detect DDOS attacks. This update process will continue in each iteration, with the particle trying to explore the solution space by moving closer to the personal best position $p_{i,best}$ and global g_{best}. This is performed until PSO finds the most effective combination of weights or parameters, maximizing the detection accuracy even in situations with imbalanced data, where the amount of normal data is much larger compared to the attack data. In this way, PSO iteratively updates the weights or parameters of the model (both in ANN and XGBoost) to optimize the detection of DDOS attacks, so that the model can recognize rare but dangerous attack patterns in the network data stream.

J. Model Evaluation

In model evaluation, a confusion matrix was used to determine the model performance. Some elements in the confusion matrix include False Positive (FP), False Negative (FN), True Positive (TP), and True Negative (TN). This confusion matrix is used to describe the extent of the classification effectiveness. The confusion matrix can be used to determine whether the predictions produced are accurate or not. In addition, the proposed model was evaluated using metrics commonly used in DDOS detection.

Accuracy reflects the model's prediction performance, measuring the total percentage of correctly detected results, both normal and anomalous. Accuracy was calculated using:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (4)$$

Precision is the ratio of true detected attacks compared to the total number of detected attacks. Precision shows how many positive DDOS detections were correctly predicted.

$$Precision\ (P) = \frac{TP}{FP+TP} \quad (5)$$

Recall, also known as Detection Rate (DR), often referred to as true positive rate (TPR), is the ratio of success in identifying attacks compared to their total amount.

$$Recall\ (r) = \frac{TP}{FN+TP} \quad (6)$$

F-score or F1 score considers both false positives and false negatives. F-score is very useful, especially in cases where the distribution of class labels is imbalanced, and is the harmonic balance between recall and precision.

$$F_{score} = 2 \times \frac{Precision \times Recall}{Precision+Recall} \quad (7)$$

III. RESULTS

Table II summarizes the various experimental scenarios conducted, along with the models used and the input variables applied to each scenario. Each scenario was designed to assess the performance of different machine learning algorithms for comparison, including ANN, XGboost, SMOTE PSO LSTM-GRU, Random Forest, as well as hybrid models such as PSO-ANN-XGBoost and SMOTE-PSO-ANN-XGBoost. Table II lists the specific input features used in each model, including network traffic attributes such as TCP packets, UDP packets, external IP, and DNS query time, among other variables. These input variables were selected based on their relevance in detecting DDOS attacks on imbalanced networks and were used to train and test the models in each experimental scenario. These experimental scenarios provide a comprehensive overview of how various machine-learning approaches and input configurations were tested for their effectiveness in detecting DDOS attacks on networks.

TABLE II. DETAILS ON EXPERIMENTS

Scenario	Model	Input variables
S1	DBSCAN-SMOTE-ANN	pkts, byte, rate, proto, flgs, state, saddr, daddr, dport
S2	DBSCAN-SMOTE-XGBOOST	pkts, byte, rate, proto, flgs, state, saddr, daddr, dport
S3	DBSCAN-SMOTE-ANN-XGBoost	pkts, byte, rate, proto, flgs, state, saddr, daddr, dport
S4	Random Forest	pkts, byte, rate, proto, flgs, state, saddr, daddr, dport
S5	SMOTE PSO LSTM-GRU	pkts, byte, rate, proto, flgs, state, saddr, daddr, dport
S6	DBSCAN-SMOTE-ANN-XGBoost-PSO	pkts, byte, rate, proto, flgs, state, saddr, daddr, dport

Figure 3 and Table III show the model performance metrics in six different scenarios (S1 to S6). The results show that model performance varies across scenarios. Overall, scenario S6 performed the best on all metrics, with accuracy, precision and F1 score values approaching 96%. In contrast, scenario S3 performed lowest across almost all metrics, especially in Sensitivity and F1-score. This analysis shows that model effectiveness is highly dependent on the conditions or parameters applied in each scenario, providing important insights for further optimization.

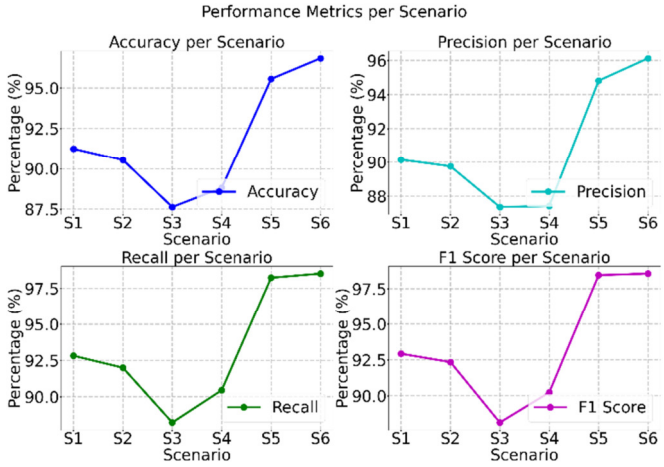

Fig. 3. Performance metrics per scenario.

TABLE III. PERFORMANCE METRICS PER SCENARIO

Scenario	Model	Accuracy	Precision	Recall	F1 score
S1	DBSCAN-SMOTE-ANN	91.25%	90.15%	92.82%	92.93%
S2	DBSCAN-SMOTE-XGBOOST	90.54%	89.76%	92.00%	92.35%
S3	DBSCAN-SMOTE-ANN-XGBoost	87.63%	87.36%	88,23%	88.15%
S4	Random Forest	88.81%	87.42%	90.45%	90.24%
S5	SMOTE PSO LSTM-GRU	95.56%	94.82%	98.24%	98.45%
S6	DBSCAN-SMOTE-ANN-XGBoost-PSO	96.83%	96.13%	98.53%	98.56%

To further validate the performance of the DBSCAN-SMOTE-ANN-XGBoost-PSO model, Table IV presents the confusion matrix of the DBSCAN-SMOTE-ANN-XGBoost-PSO model. The high TP (69,296) and TN (955,000) values indicate the effectiveness of the model in detecting DDoS attacks, while the relatively low number of FP (23,279) and FN (1,023) indicate minimal misclassification rates.

TABLE IV. CONFUSION MATRIX FOR DBSCAN-SMOTE-ANN-XGBOOST-PSO

Actual \ Predicted	No Attack (0)	Attack (1)	Total
No Attack (0)	955,000 (TN)	23,279 (FP)	978,279
Attack (1)	1,023 (FN)	69,296 (TP)	70,319
Total	956,023	92,575	1,048,598

TABLE V. TRAINING AND VALIDATION RESULTS

Model	Training loss	Validation loss	Training accuracy (%)	Validation accuracy (%)
Random Forest	0.32	0.25	91.25	88.81
SMOTE PSO LSTM-GRU	0.2	0.15	96	95.56
DBSCAN SMOTE-ANN	0.28	0.21	92.8	91.25
DBSCAN SMOTE-XGBOOST	0.3	0.22	92.1	90.54
DBSCAN SMOTE-ANN-XGBOOST	0.4	0.35	89	87.63
DBSCAN SMOTE-ANN-XGBOOST-PSO	0.18	0.12	97.5	96.83

Table V presents a performance comparison of several machine learning models, summarizing their results. SMOTE PSO LSTM-GRU and DBSCAN-SMOTE-ANN-XGBoost-PSO showed superior results compared to other models in terms of accuracy and F1 score. This table provides an overview of the effectiveness of various models on the same dataset, allowing the selection of the optimal model for a specific purpose.

IV. DISCUSSION

The DBSCAN-SMOTE-ANN-XGBoost-PSO (S6) model stood out as the best in almost all metrics tested. This model had the highest accuracy of 96.83%, indicating that it successfully predicted classes with the lowest error rate. In addition, S6 also excelled in terms of sensitivity (93.23%) and precision (96.13%), which means that this model is very good at detecting positive classes (avoiding FN) and producing accurate positive predictions (avoiding FP). In addition, this model also shows the highest specificity (94.27%), indicating that it is very good at avoiding prediction errors in the negative class. Its F1-score is also very high (98.56%), reflecting a very good balance between precision and recall.

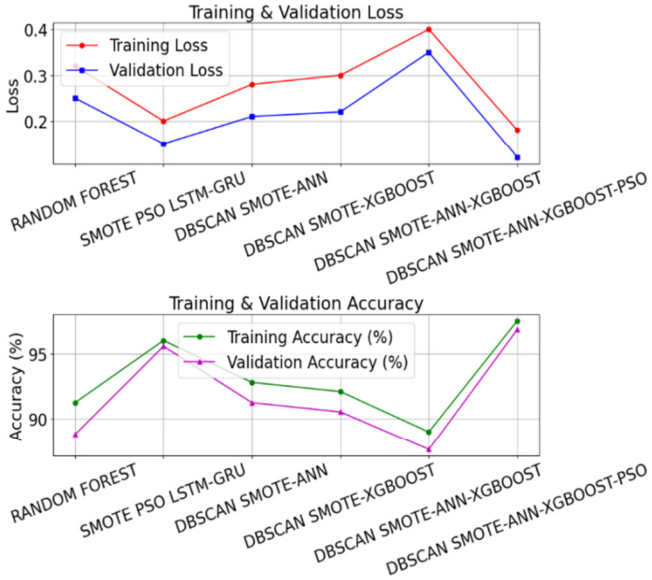

Fig. 4. Training validation loss and accuracy.

The SMOTE PSO LSTM-GRU model (S5) also performed very well, although it was slightly behind S6. It achieved an accuracy of 95.56%, which was very good, and an F1-score that was almost on par with S6 (98.45%). Although slightly lower in terms of sensitivity (92.36%) and specificity (91.49%) than S6, SMOTE PSO LSTM-GRU was still a solid model, with an excellent ability to detect the positive class and produce accurate predictions. This suggests that SMOTE PSO LSTM-GRU is a good choice if high F1-score performance is a priority, although S6 still excels.

The DBSCAN-SMOTE-ANN model (S1) showed the highest sensitivity among all models (93.29%) but had lower accuracy (91.25%) and F1-score (92.93%) compared to

SMOTE PSO LSTM-GRU and S6. Although these models are quite good at detecting positive classes, S1 is not as good as S5 and S6 in terms of metric balance. In addition, the DBSCAN-SMOTE-ANN-XGBoost (S3) and Random Forest (S4) models showed lower performance, with accuracies of 87.63% and 88.81% respectively, meaning they are less effective in making correct predictions compared to the other models. Both also had lower F1-score values, indicating that they are not as optimal in terms of the balance between precision and recall.

Overall, S6 (DBSCAN-SMOTE-ANN-XGBoost-PSO) is superior, showing the best ability to detect positive classes, avoid errors in negative classes, and produce very accurate predictions, with a very good balance between precision and recall. S5 (SMOTE PSO LSTM-GRU) is a very good second choice but lags slightly behind in some metrics such as sensitivity and specificity. S1 (DBSCAN-SMOTE-ANN) and S3 (DBSCAN-SMOTE-ANN-XGBoost), although good in some aspects, cannot outperform S5 and S6 in overall performance. Random Forest (S4) also performs less than the best models.

In the performance analysis of several DDoS attack detection models, Table VI provides a comparison of the accuracy of previously various deep learning-based approaches and other techniques. Existing research shows that several models show excellent results in detecting DDoS attacks.

TABLE VI. COMPARISON WITH PREVIOUS STUDIES

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
DBSCAN-SMOTE-LSTM [28]	96.12	93.6	96.2	98.3
WF-HDL [35]	99.69	99.03	99.07	99.05
DDAD-SOEL [36]	99.81	99.6	99.61	99.59
CNN+BiLSTM [37]	94.52	93.44	92.04	94.74
AE-MLP [38]	98.34	98.18	98.48	97.91
PB-DID [39]	96.3	N/A	N/A	N/A
DBSCAN-SMOTE-ANN-XGBoost-PSO (Proposed model)	96.83	96.13	98.53	98.56

One of the models that stands out is the WF-HDL (Wrapper Feature Selection-Based Hybrid Deep Learning Model) [35], which achieved a very high accuracy of 99.69%. This model shows high effectiveness in detecting DDoS by utilizing wrapper-based feature selection and the application of deep learning techniques. This excellent accuracy shows the importance of properly selecting features to improve the performance of attack detection models. Another model that has high performance is DDAD-SOEL (Snake Optimizer with Ensemble Learning) [36], which achieved excellent results in all metrics with an accuracy of 99.81%. The use of Snake Optimizer to guide the training process and ensemble techniques to improve accuracy have proven effective in detecting DDoS attacks in the IoT environment. In addition, the DBSCAN-SMOTE-LSTM model [28] showed solid performance with an accuracy of 96.12%. This approach utilizes a combination of DBSCAN and SMOTE techniques to handle imbalanced class problems and LSTM for sequential data analysis. This shows how the combination of multiple techniques can improve DDoS detection capabilities in networks with data imbalance. CNN+BiLSTM (Convolutional Neural Network + Bidirectional Long Short-Term Memory) [37] showed good results with an accuracy of 94.52%, where, although slightly lower than other models, the combination of CNN and BiLSTM is effective in processing time-based data and attack detection.

The proposed DBSCAN-SMOTE-ANN-XGBoost-PSO model showed competitive performance in detecting DDoS attacks. Integration of DBSCAN for anomaly detection, SMOTE for class balancing, and the hybrid ANN-XGBoost approach significantly improved detection accuracy. In addition, the PSO algorithm is used to optimize the hyperparameters, thereby improving its efficiency. In general, the results indicate that deep learning-based hybrid models, such as AE-MLP [38], PB-DID [39], and others, with optimization and ensemble techniques, such as DBSCAN-SMOTE-ANN-XGBoost-PSO, have advantages in detecting DDoS attacks. This confirms that the combination of optimization techniques, feature selection, and the use of deep learning algorithms can significantly improve performance in detecting increasingly complex cyber threats. These results show that the combination of deep learning-based methods and other techniques, such as SMOTE and optimization, can improve detection results and overcome various challenges in detecting DDoS attacks on highly dynamic and large networks.

This study makes an important contribution by showing that the hybrid ANN-XGBoost approach combined with optimal data balancing techniques can significantly improve the accuracy and sensitivity of the model in detecting attacks in real network environments. The implications of these results indicate the relevance of the proposed method for application in modern network security systems. Future studies can integrate advanced ensemble techniques or explore more complex architectures to improve generalization across different types of attacks in dynamic networks.

V. CONCLUSION

Based on the results of the experimental comparison, it can be concluded that the DBSCAN-SMOTE-ANN-XGBoost-PSO model shows excellent performance among all tested scenarios, with accuracy of 96.83%, precision of 96.13%, recall of 98.53%, and F1-score of 98.56%. These results outperform other models such as SMOTE PSO LSTM-GRU (95.56% accuracy) and DBSCAN-SMOTE-ANN-XGBoost (87.63% accuracy). Meanwhile, the SMOTE PSO LSTM-GRU model has the best performance in terms of recall and F1-score, but DBSCAN-SMOTE-ANN-XGBoost-PSO provides a better balance between all evaluation metrics. Compared to previous studies, the DBSCAN-SMOTE-ANN-XGBoost-PSO model shows quite competitive performance. Although it does not achieve the highest performance, this model still has excellent performance, superior to other models, showing great potential for classification tasks.

Future directions can involve further exploration of hyperparameter tuning to improve model performance, as well as the application of more sophisticated ensemble learning or deep learning techniques to find more optimal model combinations. In addition, experiments with larger and more

varied data can help test the reliability of the model under more complex conditions. Techniques such as regularization or transfer learning can also be explored to improve the generalization of the model, which can make significant contributions to its application in various domains.

ACKNOWLEDGMENT

The author would like to thank the Directorate of Infrastructure and Digitalization (DID) of Satya Wacana Christian University, Indonesia, for allowing the collection of all the important data needed for the implementation of the penetration test in the network to carry out this research. They also provided the resources and support needed to complete the study. The author greatly appreciates their commitment to advancing cybersecurity research. This research would not have been possible without their sincere contributions and collaboration.

REFERENCES

[1] M. B. Muzammil, M. Bilal, S. Ajmal, S. C. Shongwe, and Y. Y. Ghadi, "Unveiling Vulnerabilities of Web Attacks Considering Man in the Middle Attack and Session Hijacking," *IEEE Access*, vol. 12, pp. 6365–6375, 2024, https://doi.org/10.1109/ACCESS.2024.3350444.

[2] R. R. Brooks, L. Yu, I. Ozcelik, J. Oakley, and N. Tusing, "Distributed Denial of Service (DDoS): A History," *IEEE Annals of the History of Computing*, vol. 44, no. 2, pp. 44–54, Apr. 2022, https://doi.org/10.1109/MAHC.2021.3072582.

[3] S. Kumar, M. Dwivedi, M. Kumar, and S. S. Gill, "A comprehensive review of vulnerabilities and AI-enabled defense against DDoS attacks for securing cloud services," *Computer Science Review*, vol. 53, Aug. 2024, Art. no. 100661, https://doi.org/10.1016/j.cosrev.2024.100661.

[4] I. Tasevski and K. Jakimoski, "Overview of SQL Injection Defense Mechanisms," in *2020 28th Telecommunications Forum (TELFOR)*, Belgrade, Serbia, Nov. 2020, pp. 1–4, https://doi.org/10.1109/TELFOR51502.2020.9306676.

[5] M. Alghawazi, D. Alghazzawi, and S. Alarifi, "Detection of SQL Injection Attack Using Machine Learning Techniques: A Systematic Literature Review," *Journal of Cybersecurity and Privacy*, vol. 2, no. 4, pp. 764–777, Sep. 2022, https://doi.org/10.3390/jcp2040039.

[6] G. E. Rodríguez, J. G. Torres, P. Flores, and D. E. Benavides, "Cross-site scripting (XSS) attacks and mitigation: A survey," *Computer Networks*, vol. 166, Jan. 2020, Art. no. 106960, https://doi.org/10.1016/j.comnet.2019.106960.

[7] G. Rodríguez-Galán and J. Torres, "Personal data filtering: a systematic literature review comparing the effectiveness of XSS attacks in web applications vs cookie stealing," *Annals of Telecommunications*, vol. 79, no. 11–12, pp. 763–802, Dec. 2024, https://doi.org/10.1007/s12243-024-01022-8.

[8] A. Fadlil, I. Riadi, and M. A. Mu'min, "Mitigation from SQL Injection Attacks on Web Server using Open Web Application Security Project Framework," *International Journal of Engineering*, vol. 37, no. 4, pp. 635–645, 2024, https://doi.org/10.5829/IJE.2024.37.04A.06.

[9] R. K. Dwivedi, R. Kumar, and R. Buyya, "Gaussian Distribution-Based Machine Learning Scheme for Anomaly Detection in Healthcare Sensor Cloud:," *International Journal of Cloud Applications and Computing*, vol. 11, no. 1, pp. 52–72, Jan. 2021, https://doi.org/10.4018/IJCAC.2021010103.

[10] P. Gulihar and B. B. Gupta, "Anomaly based Mitigation of Volumetric DDoS Attack Using Client Puzzle as Proof-of-Work," in *2018 3rd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*, Bangalore, India, May 2018, pp. 2475–2479, https://doi.org/10.1109/RTEICT42901.2018.9012127.

[11] K. Ren, S. Yuan, C. Zhang, Y. Shi, and Z. Huang, "CANET: A hierarchical CNN-Attention model for Network Intrusion Detection," *Computer Communications*, vol. 205, pp. 170–181, May 2023, https://doi.org/10.1016/j.comcom.2023.04.018.

[12] L. Zhang, K. Liu, X. Xie, W. Bai, B. Wu, and P. Dong, "A data-driven network intrusion detection system using feature selection and deep learning," *Journal of Information Security and Applications*, vol. 78, Nov. 2023, Art. no. 103606, https://doi.org/10.1016/j.jisa.2023.103606.

[13] H. Lin, Q. Xue, J. Feng, and D. Bai, "Internet of things intrusion detection model and algorithm based on cloud computing and multi-feature extraction extreme learning machine," *Digital Communications and Networks*, vol. 9, no. 1, pp. 111–124, Feb. 2023, https://doi.org/10.1016/j.dcan.2022.09.021.

[14] M. Jain, G. Kaur, and V. Saxena, "A K-Means clustering and SVM based hybrid concept drift detection technique for network anomaly detection," *Expert Systems with Applications*, vol. 193, May 2022, Art. no. 116510, https://doi.org/10.1016/j.eswa.2022.116510.

[15] C. Ieracitano, A. Adeel, F. C. Morabito, and A. Hussain, "A novel statistical analysis and autoencoder driven intelligent intrusion detection approach," *Neurocomputing*, vol. 387, pp. 51–62, Apr. 2020, https://doi.org/10.1016/j.neucom.2019.11.016.

[16] J. Chen, X. Qi, L. Chen, F. Chen, and G. Cheng, "Quantum-inspired ant lion optimized hybrid k-means for cluster analysis and intrusion detection," *Knowledge-Based Systems*, vol. 203, Sep. 2020, Art. no. 106167, https://doi.org/10.1016/j.knosys.2020.106167.

[17] A. Sanmorino, L. Marnisah, and H. D. Kesuma, "Detection of DDoS Attacks using Fine-Tuned Multi-Layer Perceptron Models," *Engineering, Technology & Applied Science Research*, vol. 14, no. 5, pp. 16444–16449, Oct. 2024, https://doi.org/10.48084/etasr.8362.

[18] D. Wu, Y. Deng, and M. Li, "FL-MGVN: Federated learning for anomaly detection using mixed gaussian variational self-encoding network," *Information Processing & Management*, vol. 59, no. 2, Mar. 2022, Art. no. 102839, https://doi.org/10.1016/j.ipm.2021.102839.

[19] M. Kurni, M. S. Md, B. B. Yannam, and A. S. T, "MRPO-Deep maxout: Manta ray political optimization based Deep maxout network for big data intrusion detection using spark architecture," *Advances in Engineering Software*, vol. 174, Dec. 2022, Art. no. 103324, https://doi.org/10.1016/j.advengsoft.2022.103324.

[20] P. B. Udas, Md. E. Karim, and K. S. Roy, "SPIDER: A shallow PCA based network intrusion detection system with enhanced recurrent neural networks," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 10, pp. 10246–10272, Nov. 2022, https://doi.org/10.1016/j.jksuci.2022.10.019.

[21] M. P. Novaes, L. F. Carvalho, J. Lloret, and M. L. Proença, "Adversarial Deep Learning approach detection and defense against DDoS attacks in SDN environments," *Future Generation Computer Systems*, vol. 125, pp. 156–167, Dec. 2021, https://doi.org/10.1016/j.future.2021.06.047.

[22] D. Alghazzawi, O. Bamasag, H. Ullah, and M. Z. Asghar, "Efficient Detection of DDoS Attacks Using a Hybrid Deep Learning Model with Improved Feature Selection," *Applied Sciences*, vol. 11, no. 24, Dec. 2021, Art. no. 11634, https://doi.org/10.3390/app112411634.

[23] S. Nandi, S. Phadikar, and K. Majumder, "Detection of DDoS Attack and Classification Using a Hybrid Approach," in *2020 Third ISEA Conference on Security and Privacy (ISEA-ISAP)*, Guwahati, India, Feb. 2020, pp. 41–47, https://doi.org/10.1109/ISEA-ISAP49340.2020.234999.

[24] X. H. Nguyen and K. H. Le, "Robust detection of unknown DoS/DDoS attacks in IoT networks using a hybrid learning model," *Internet of Things*, vol. 23, Oct. 2023, Art. no. 100851, https://doi.org/10.1016/j.iot.2023.100851.

[25] S. Haider *et al.*, "A Deep CNN Ensemble Framework for Efficient DDoS Attack Detection in Software Defined Networks," *IEEE Access*, vol. 8, pp. 53972–53983, 2020, https://doi.org/10.1109/ACCESS.2020.2976908.

[26] M. P. Novaes, L. F. Carvalho, J. Lloret, and M. L. Proenca, "Long Short-Term Memory and Fuzzy Logic for Anomaly Detection and Mitigation in Software-Defined Network Environment," *IEEE Access*, vol. 8, pp. 83765–83781, 2020, https://doi.org/10.1109/ACCESS.2020.2992044.

[27] Z. S. Dhahir, "A Hybrid Approach for Efficient DDoS Detection in Network Traffic Using CBLOF-Based Feature Engineering and XGBoost," *Journal of Future Artificial Intelligence and Technologies*,

vol. 1, no. 2, pp. 174–190, Sep. 2024, https://doi.org/10.62411/faith. 2024-33.

[28] R. Efendi, T. Wahyono, and I. R. Widiasari, "DBSCAN SMOTE LSTM: Effective Strategies for Distributed Denial of Service Detection in Imbalanced Network Environments," *Big Data and Cognitive Computing*, vol. 8, no. 9, Sep. 2024, Art. no. 118, https://doi.org/10.3390/bdcc8090118.

[29] S. Muruganandam, R. Joshi, P. Suresh, N. Balakrishna, K. H. Kishore, and S. V. Manikanthan, "A deep learning based feed forward artificial neural network to predict the K-barriers for intrusion detection using a wireless sensor network," *Measurement: Sensors*, vol. 25, Feb. 2023, Art. no. 100613, https://doi.org/10.1016/j.measen.2022.100613.

[30] J. Al Amien, H. Ab Ghani, N. I. Md Saleh, E. Ismanto, and R. Gunawan, "Intrusion detection system for imbalance ratio class using weighted XGBoost classifier," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 21, no. 5, Oct. 2023, Art. no. 1102, https://doi.org/10.12928/telkomnika.v21i5.24735.

[31] G. Mohiuddin *et al.*, "Intrusion Detection using hybridized Meta-heuristic techniques with Weighted XGBoost Classifier," *Expert Systems with Applications*, vol. 232, Dec. 2023, Art. no. 120596, https://doi.org/10.1016/j.eswa.2023.120596.

[32] S. S. Dhaliwal, A. A. Nahid, and R. Abbas, "Effective Intrusion Detection System Using XGBoost," *Information*, vol. 9, no. 7, Jun. 2018, Art. no. 149, https://doi.org/10.3390/info9070149.

[33] M. Hajihosseinlou, A. Maghsoudi, and R. Ghezelbash, "Intelligent mapping of geochemical anomalies: Adaptation of DBSCAN and mean-shift clustering approaches," *Journal of Geochemical Exploration*, vol. 258, Mar. 2024, Art. no. 107393, https://doi.org/10.1016/j.gexplo.2024. 107393.

[34] S. Mayabadi and H. Saadatfar, "Two density-based sampling approaches for imbalanced and overlapping data," *Knowledge-Based Systems*, vol. 241, Apr. 2022, Art. no. 108217, https://doi.org/10.1016/j.knosys.2022. 108217.

[35] G. N. Tikhe and P. S. Patheja, "A Wrapper Feature Selection Based Hybrid Deep Learning Model for DDoS Detection in a Network with NFV Behaviors," *Wireless Personal Communications*, vol. 133, no. 1, pp. 481–506, Nov. 2023, https://doi.org/10.1007/s11277-023-10775-9.

[36] M. Aljebreen, H. A. Mengash, M. A. Arasi, S. S. Aljameel, A. S. Salama, and M. A. Hamza, "Enhancing DDoS Attack Detection Using Snake Optimizer With Ensemble Learning on Internet of Things Environment," *IEEE Access*, vol. 11, pp. 104745–104753, 2023, https://doi.org/10.1109/ACCESS.2023.3318316.

[37] D. Alghazzawi, O. Bamasag, H. Ullah, and M. Z. Asghar, "Efficient Detection of DDoS Attacks Using a Hybrid Deep Learning Model with Improved Feature Selection," *Applied Sciences*, vol. 11, no. 24, Dec. 2021, Art. no. 11634, https://doi.org/10.3390/app112411634.

[38] Y. Wei, J. Jang-Jaccard, F. Sabrina, A. Singh, W. Xu, and S. Camtepe, "AE-MLP: A Hybrid Deep Learning Approach for DDoS Detection and Classification," *IEEE Access*, vol. 9, pp. 146810–146821, 2021, https://doi.org/10.1109/ACCESS.2021.3123791.

[39] M. Zeeshan *et al.*, "Protocol-Based Deep Intrusion Detection for DoS and DDoS Attacks Using UNSW-NB15 and Bot-IoT Data-Sets," *IEEE Access*, vol. 10, pp. 2269–2283, 2022, https://doi.org/10.1109/ ACCESS.2021.3137201.

AUTHORS PROFILE

Rissal Efendi earned his Master's in Information Systems from Diponegoro University. Since 2018, he has been a lecturer in the Pedagogy of Informatics Engineering and Computer Science Study Program at Satya Wacana Christian University. He has working experience as a computer network engineer in some private companies. His research interests are in the field of computer networks, cyber security, machine learning, deep learning, and pervasive computing. Now, he focuses on conducting IoT-based Flood Disaster Prediction Research in Semarang, Central Java, Indonesia.