electronics

MDPI

Article

Empowering Digital Resilience: Machine Learning-Based Policing Models for Cyber-Attack Detection in Wi-Fi Networks

Suryadi MT [1,2,*], Achmad Eriza Aminanto [2] and Muhamad Erza Aminanto [3]

[1] Department of Mathematics, Universitas Indonesia, Depok 16424, Indonesia
[2] Graduate School of Strategic and Global Studies, Universitas Indonesia, Depok 16424, Indonesia
[3] Cyber Security, Monash University Indonesia, Banten 15345, Indonesia
* Correspondence: yadi.mt@sci.ui.ac.id

Abstract: In the wake of the COVID-19 pandemic, there has been a significant digital transformation. The widespread use of wireless communication in IoT has posed security challenges due to its vulnerability to cybercrime. The Indonesian National Police's Directorate of Cyber Crime is expected to play a preventive role in supervising these attacks, despite lacking a specific cyber-attack prevention function. An Intrusion Detection System (IDS), employing artificial intelligence, can differentiate between cyber-attacks and non-attacks. This study focuses on developing a machine learning-based policing model to detect cyber-attacks on Wi-Fi networks. The model analyzes network data, enabling quick identification of attack indications in the command room. The research involves simulations and analyses of various feature selection methods and classification models using a public dataset of cyber-attacks on Wi-Fi networks. The study identifies mutual information with 20 features such as the optimal feature reduction method and the Neural Network as the best classification method, achieving a 94% F1-Score within 95 s. These results demonstrate the proposed IDS's ability to swiftly detect attacks, aligning with previous research findings.

Keywords: cyber-attack identification; cyber policing; intrusion detection system; machine learning

Citation: MT, S.; Aminanto, A.E.; Aminanto, M.E. Empowering Digital Resilience: Machine Learning-Based Policing Models for Cyber-Attack Detection in Wi-Fi Networks. *Electronics* **2024**, *13*, 2583. https://doi.org/10.3390/electronics13132583

Academic Editor: Andreas Mauthe

Received: 4 May 2024
Revised: 18 June 2024
Accepted: 26 June 2024
Published: 30 June 2024

1. Introduction

In Indonesia, the incidence of cybercrime has been steadily rising, posing significant challenges to law enforcement, especially in the wake of the COVID-19 pandemic. The proliferation of online activities, particularly on social media, has provided cybercriminals with ample opportunities. According to data from the Indonesian National Police, the majority of cybercrimes are fraud-related, leading to substantial financial losses. In 2019, the total losses amounted to 49.5 billion Indonesian Rupiah, with an additional 17.69 billion Rupiah in the first half of 2020. Various mediums such as email, websites, social media, and telecommunications are exploited by fraudsters. Email-based fraud alone accounted for over 144 billion Rupiah in losses, making it the most significant modus operandi in 2019. Websites were the second most common target, with losses exceeding 73 billion Rupiah from 351 reported cases, primarily in ecommerce platforms. The Indonesian National Police, realizing the severity of the situation, has been actively organizing awareness campaigns, such as the content competition highlighted in Attachment 2, to educate the public about cybercrimes.

To combat this rising trend, the Indonesian National Police established the Directorate of Cyber Crime (Dittipidsiber) under Bareskrim Polri in 2017. However, Dittipidsiber currently focuses on cybercrime enforcement and lacks preventive measures. Presently, it relies on reports of cybercrimes, leading to delays in response and allowing cybercriminals to operate unhindered. To address this, there is a need for proactive measures, especially in the realm of the Internet of Things (IoT).

IoT devices, often small and battery-powered, are vulnerable due to their limited computational capabilities. They are susceptible to passive and active attacks, including

Distributed Denial of Service (DDoS) attacks, targeting bandwidth, CPU, or memory of the target system. Additionally, the complexity and volume of data generated by IoT devices, coupled with the widespread use of Wi-Fi networks, present significant challenges in detecting and preventing cyber-attacks. Intrusion Detection Systems (IDS) play a vital role in network security, alerting users or administrators to unauthorized attempts or access [1]. However, developing IDS for unsupervised cyber-attack detection in the IoT environment is challenging.

Several studies have attempted to tackle this challenge, focusing on binary and multiple classifications of Wi-Fi network attack data using datasets like AWID [2]. The integration of machine learning techniques, particularly deep learning, has shown promise in enhancing large-scale IDS on Wi-Fi networks by improving detection rates and reducing processing time [3–7]. To proactively address cybercrime, there is a need for lightweight machine learning techniques with low processing times [8].

The escalating trend of cybercrimes in Indonesia demands proactive measures from law enforcement. Establishing an efficient and timely response system, particularly in the IoT environment, is crucial. Integrating advanced machine learning techniques into IDS can significantly enhance the detection and prevention of cyber-attacks, thereby safeguarding digital assets and ensuring a more secure online environment.

Over the years, considerable research has been devoted to Intrusion Detection Systems (IDS), specifically focusing on the identification and categorization of attacks in wireless networks. This process entails considering a variety of factors, each of which can be seen as distinct attributes for individual data records. In this study, the Wi-Fi intrusion dataset, AWID2, established by Kolias et al. in 2015 [2], was employed. With 154 attributes per record, AWID2 qualifies as a high-dimensional dataset. Previous studies have utilized AWID2 to tackle binary or multiclass classification challenges based on attack labels.

For instance, Thing et al. [9] introduced an IDS incorporating deep learning models, achieving an impressive 98.66% accuracy in a 4-class classification scenario using all 154 attributes. Similarly, Kasongo et al. [10] utilized a Feed Forward Deep Neural Network (FFDNN) model for 4-class classification, achieving a notable accuracy score of 99.77%. However, it is worth noting that Kasongo's FFDNN model only utilized 26 out of the 154 attributes present in the original AWID2 dataset [10].

Furthermore, Kasongo et al. [10] extended their research to transform the AWID2 dataset into a binary classification problem, demonstrating an outstanding accuracy rate of 99.66%. Another significant contribution came from Aminanto et al. [4], who introduced the Deep-Feature Extraction and Selection (D-FES) method. This approach exhibited a remarkable performance, achieving an accuracy of 99.97% and an F1 score of 99.94% in the context of binary classification challenges [4].

The main contributions of this paper are two-fold. First, while previous work has shown good performance in 4-class attack classification using deep learning models, which are computationally heavy, this study examines lightweight machine learning models for the same task. These models, although slightly less accurate, offer a significant advantage in terms of resource efficiency and speed. Second, we explore the feasibility of implementing these lightweight models in real-world police department systems, aiming to enhance their capabilities in detecting cyber-attacks with limited computational resources. This study provides a practical approach that could significantly improve the operational efficiency of police departments in the future.

In summary, these studies highlight the progression of IDS methodologies, particularly within the domain of wireless networks. By employing advanced techniques and carefully selecting attributes from high-dimensional datasets such as AWID2, researchers have made substantial advancements in enhancing the accuracy and efficiency of intrusion detection systems. These developments lay the groundwork for more robust cybersecurity protocols.

2. Materials and Methods

In this study, the AWID2 dataset, sourced from IEEE 802.11 wireless networks and introduced by Kolias in 2015 [2], was selected. This dataset is particularly valuable as it simulates real network conditions with various attacks. The simulation includes a physical setting with ten regular users and one adversarial node, all connected through a Wired Equivalent Privacy (WEP) protected access point.

As outlined in Table 1, the dataset used in our research comprises four distinct classes. The first class, labeled as "normal" (class 0 in our experiment), represents an untouched network and constitutes the majority of the dataset, outnumbering all other classes combined at a ratio of 10:1. The remaining classes, denoted as "impersonation", "injection", and "flooding", correspond to different attack types and were designated as class 1, class 2, and class 3, respectively, during our experiments.

Table 1. Dataset class distribution among four classes.

Class	Training	Testing
Benign	1,633,190	530,785
Impersonation	48,522	20,079
Injection	65,379	16,682
Flooding	48,484	8097
Total Original Dataset	1,795,575	575,643

The Impersonation class includes four attacks: Caffe Latte [11], Hirte [12], Evil Twin [13], and Rogue Access Point (AP) [14]. Remote Caffe Latte and Hirte attacks involve obtaining the WEP key. Typically, clients store networks they have connected to for future use. The attacker seizes the WEP key during the greeting protocol. Hirte and Caffe Latte differ in strategy, with the latter using a fragmentation attack. Meanwhile, Evil Twin and Rogue AP attacks exploit victims' tendency to search for nearby networks by name. The attacker impersonates the victim's official AP.

The Injection category encompasses three attacks: ARP Injection [15], Chop-Chop [16], and Fragmentation [17]. ARP Injection manipulates the network to generate numerous Initialization Vectors (IV) for key cracking. Chop-Chop exposes a portion of the keystream's final bytes, allowing the attacker to deduce the cipher without the key. Fragmentation exploits the 802.11 protocol, requiring packets exceeding the maximum length to be fragmented and delivered independently.

The Flooding attacks include at least 13 distinct types, such as Beacon Flooding [18], Request to Send (RTS) Flooding [19], and Fake Power Saving [20]. Beacon Flooding transmits false beacons with spoofed or nonexistent identifiers, causing client overruns. RTS Flooding targets the RTS/CTS (Request to Send/Clear to Send) protocol by sending numerous RTS frames with a lengthy duration window to keep clients occupied. The Fake Power Saving attack exploits the power-saving protocol by forcing clients into sleep mode and then ignoring them, rendering them ineffective.

We explain the process of data preparation and data pre-processing. Then the process is followed by three feature selection models and three classifiers.

The proposed method could be defined as sequences as shown in Figure 1. The first sequence is data pre-processing, in which we process the normalized AWID2 datasets. Next, the dataset is sampled to maintain training scalability, followed by examining the best feature selection models among mutual information, random forest, and AdaBoost. Last, we also chose the best classifier from three models: logistic regression, XG-Boost, and Neural Network.

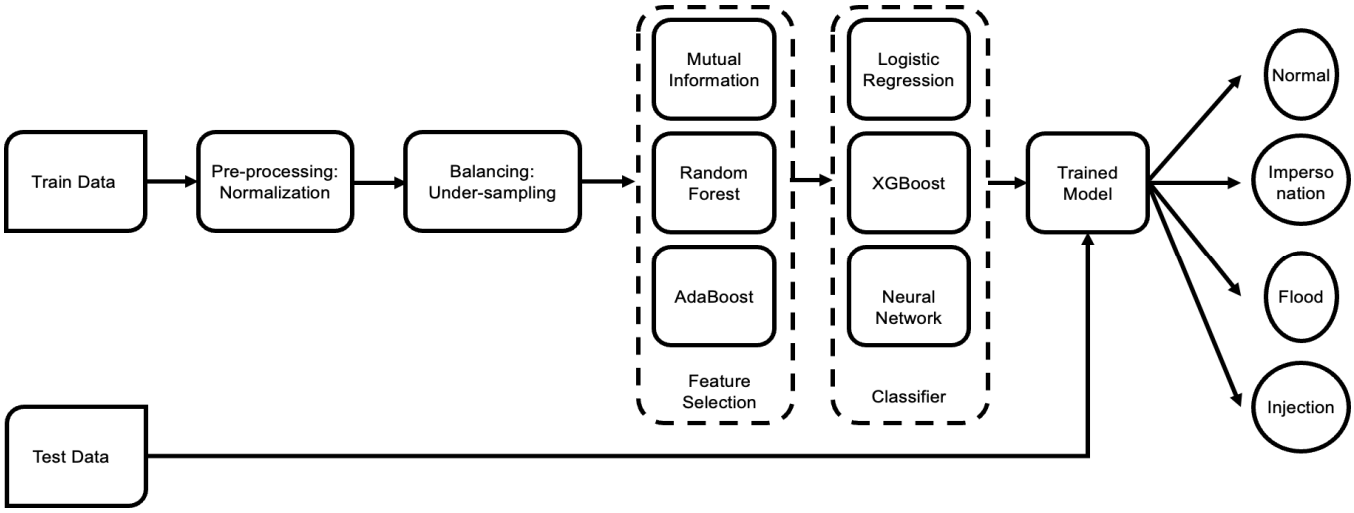

Figure 1. The proposed method with three feature selection models and three classifiers.

2.1. Pre-Processing

The initial step involves data separation, dividing it into training and testing sets. The training data serves as the input for machine learning, enabling the creation of a predictive model. In this study, the dataset used is referred to as AWID2 [2], containing internet traffic on Wi-Fi networks, encompassing three distinct attack types: flooding, injection, and impersonation. This dataset comprises over 2 million records classified into 4 classes. Class 0 represents the 'Normal' category, which is the most abundant among all classes. Impersonation is labeled as class 1, injection as class 2, and flooding as class 3. The first task involves normalizing the entire dataset and scaling the data from 0 to 1. The dataset utilized consists of a total of 2,371,218 Wi-Fi data points, divided into training and testing data with a 75:25 ratio. We amalgamated all these data into one large dataset to ensure consistency.

In this research, only 0.5% of the dataset was used to reduce computational demands while maintaining the original data ratio. Despite the reduction, the dataset's distribution was preserved. The total reduction resulted in 11,856 data points, further divided into training and testing data with a 75:25 ratio. Balancing the number of samples between normal and attack classes is crucial, as machine learning learns more effectively when the ratio of each class is equal.

The next step involves data pre-processing, specifically through a process called normalization. Normalization standardizes data so that each attribute has the same minimum and maximum values, typically ranging from 0 to 1. We merged the normalized training and testing data, incorporating four different class labels, to reduce biases. In this study, we only utilized approximately 0.5% of the dataset due to its extensive size. This precaution was necessary to prevent computer memory overload during the training process. The dataset was split into two subsets: one for training and another for testing. Often, variables or features in a dataset have varying scales. For instance, one variable might have values in the millions, while another is in the hundreds. Normalization is essential to standardize the range of values across features in the data. This method adjusts values within each feature to a consistent scale across all features in the dataset.

During the data analysis process, it is common to encounter imbalanced datasets, where the proportions between different groups are uneven. The class with fewer instances is often termed the minority class, while the more dominant class is referred to as the majority class. Conducting classification analysis directly on imbalanced data typically leads to less accurate results. Hence, a data sampling process is employed, involving drawing samples from the existing data. This process can be carried out in two different ways: undersampling and oversampling. Undersampling involves selecting samples in a way that reduces the ratio of the majority class to the minority class. On the contrary, oversampling aims to increase the ratio of the minority class to the majority class. For

instance, in the dataset used for this study, there were initially 37,817,835 instances, with 36,732,463 being normal and 1,085,372 being attacks, resulting in a ratio of 1:33 between attacks and non-attacks. This ratio is significantly large and requires techniques to address it. One approach is undersampling, which is necessary because the number of normal instances is exceedingly high and needs to be reduced to approach the attack instances. After applying undersampling, the observations were reduced to 325,704 instances, with 163,319 being normal and 162,385 being attacks, thus achieving a balanced 1:1 ratio. This balanced ratio is essential for enhancing the accuracy of machine learning analysis.

2.2. Feature Selection

Feature selection has been extensively studied and utilized in machine learning and data mining. In this context, features are also referred to as attributes or variables, representing values derived from a process or system measured or constructed from input variables. The general objective of feature selection is to choose the smallest subset of features that can represent the overall features [21]. The purposes of feature selection are as follows:

a. Enhancing Generalization Performance: Selecting features that can generalize as effectively as using all features.
b. Providing Robust Feature Selection: Enabling the selection of features that are robust and facilitate faster or lighter advanced analysis tasks.
c. Facilitating Better and Easier Analysis: Obtaining a more efficient and straightforward analysis process from the chosen feature subset.

Several methods are available for feature selection. In this study, three feature selection methods were employed: mutual information (MI), random forest, and AdaBoost.

Mutual information was chosen due to its measurement based on statistical independence, possessing two properties: it can measure relationships between random variables, including nonlinear relationships, and it is invariant under invertible and differentiable feature space transformations. Mutual information quantifies the amount of information shared by one random variable with another. This definition is valuable in the context of feature selection because it provides a method to quantify the relevance of feature subsets.

Different features inherently provide distinct roles or information in the classification process. Additionally, there are features that have a low impact (insignificant influence on the classification process) and can be eliminated, meaning they are not used for classification. The Random Forest method proves beneficial in this context. Its distinctive features include:

a. Efficient feature selection so that it can perform feature selection on a large dataset with high dimensions swiftly.
b. Random selection that can enhance classification performance.
c. Unbiased estimators, during forest construction, the method can create unbiased estimators.

AdaBoost, short for Adaptive Boosting, is one of the most promising methods in machine learning. It converges quickly and is easy to implement [22]. The main idea behind AdaBoost is to maintain the distribution or set of weights on the training data. The weights on the distribution of training data at round t are denoted as $Dt(i)$. Initially, all weights have the same value. Then, in each round, these weights are adjusted according to the weights Dt.

2.3. Classifier

After feature selection, the next step is classification analysis, a process of assigning labels (classes) to previously unlabeled data based on the learned features or variables from machine learning. This research employs lightweight machine learning classification methods. While deep learning methods are powerful and have been used in cyber-attack detection, the aim of this paper is to develop lightweight models suitable for real-world deployment in police departments, which often have limited computational resources. Models such as logistic regression, neural network, and XG-Boost were chosen because they

require less computational power and memory, making them more practical for real-time applications. Despite their simplicity, these models achieved high accuracy, indicating that complex deep-learning models may not be necessary for this task. Additionally, lightweight models are easier to implement, maintain, and update, facilitating faster deployment and quicker responses to evolving cyber threats.

One of the lightweight methods used is logistic regression. Linear models like logistic regression can be relatively simple and cost-effective. Logistic regression is a specific form of linear regression. In this application, the response variable represents class labels, and the independent variables represent the features or columns in the dataset. Using logistic regression in this data analysis, the response variable represents binary labels, which violate ordinary linear regression assumptions. Logistic regression models provide an advantage in estimating the classification probabilities of binary response variables [23]. Its limitation lies in dealing with nonlinear problems and interactions between independent variables. Maximum likelihood ratio is employed to determine the significance of independent variables in the logistic regression equation. Logistic regression is valuable when attempting to predict the influence of independent variables, similar to linear regression, but it is specifically suited for binary dependent variables.

XG-Boost, short for eXtreme Gradient Boosting, was created by Chen and Guestrin in 2016 to enhance tree-based learning performance [24]. XG-Boost is a classification method that is lightweight as it is constructed from a set of decision trees similar to Random Forest. XG-Boost is also commonly referred to as Gradient Boosting, as seen in the Python library Sklearn. Generally, XG-Boost is a reliable classification method due to its high classification accuracy, fast processing time, low computational complexity, and ease of use [25].

Neural Network (NN) is a commonly used machine learning algorithm for classification purposes [26]. NN is an information processing model that resembles the structure of the human brain, consisting of biological neural systems. Meanwhile, according to Haykin [27], NN can be described as a machine that produces a system resembling the human brain, trained to understand a specific task.

NN offers advantages over other statistical models because it does not rely on assumptions about the properties or distribution of the data it learns. Hence, NN is highly beneficial for practical applications and does not require hypothesis testing like other statistical models. NN is incredibly flexible and capable of reducing data complexity, modeling nonlinear regression, and discrimination. It also exhibits a higher error tolerance compared to other models such as Support Vector Machine (SVM) and Decision Tree because NN can handle incomplete data, noisy data, and nonlinear problems. Finally, NN is considered more stable and faster [28].

2.4. Experimental Scenarios

This study was conducted across nine scenarios, each representing a combination of different feature selection and classification methods. The first three scenarios utilized mutual information for feature selection and employed diverse classification methods: logistic regression, XG-Boost, and Neural Network. Following that, the subsequent three scenarios used Random Forest for feature selection with different classification methods. The final three scenarios involved a Neural Network for feature selection, coupled with various classification methods. Each scenario was repeated three times, corresponding to the number of feature sets used in the feature selection analysis: 20, 50, and 100 features. The Neural Network classifier used in this study is a Python library called Sklearn MLPClassifier. While the logistic regression and the XGBoost used LogisticRegression and XGBClassifier, respectively. The architecture of NN is as follows: number of hidden layer = 1; Number of neurons in that hidden layer = 50; activation unit = rectified linear unit; optimizer = adam, which is a stochastic gradient-based optimizer; regularizer = L2 regularization. Table 2 shows the details of each of the nine scenarios.

Table 2. Nine Scenarios of Experiments.

Scenario	Feature Selection	Classifier
1	Mutual Information	Logistic Regression
2		XG-Boost
3		Neural Network
4	Random Forest	Logistic Regression
5		XG-Boost
6		Neural Network
7	AdaBoost	Logistic Regression
8		XG-Boost
9		Neural Network

3. Results

We categorized the extensive AWID2 dataset, comprising over two million Wi-Fi network data records, into four distinct classes. Utilizing square values to filter the attributes, we devised eight unique configurations. The sequences of pipeline implementation remained consistent across all configurations. The experiments were conducted using Google Colab, a cloud-based platform that provides access to GPU and TPU resources. Google Colab allows for the execution of Python code in a Jupyter notebook environment, facilitating the implementation and testing of machine learning models. This platform supports various machine learning libraries such as TensorFlow, Keras, and Scikit-learn, which were utilized in our experiments. While the specific hardware configurations are managed by Google, the platform ensures that adequate computational resources are available for the tasks. This setup was chosen for its ease of use, accessibility, and ability to handle the computational demands of our experiments.

3.1. Evaluation Metrics

To ensure a fair evaluation of each model's performance, dealing with the imbalanced data distribution in multiclass datasets is crucial. We considered Accuracy (Acc) as an additional measure in our evaluation, providing a general overview of correct predictions. However, in our case, due to the highly imbalanced testing dataset, accuracy alone is insufficient to assess the model's performance. Accuracy tends to favor the dominant class and does not effectively account for minority class prediction.

In light of this challenge, we emphasized the F1 score (F1) as our main metric. F1 score provides a more precise understanding of how well the model's predictions align with both precision and recall. Precision signifies the ratio of true positive predictions to the total positive predictions (True Positive + False Positive), while Recall, or Detection Rate, compares true positive predictions to the actual positive count (True Positive plus False Negative). These four metrics are expressed, respectively, in the form of the following equation:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

$$F1 = \frac{2TP}{2TP + FP + FN} \tag{2}$$

$$Precission = \frac{TP}{TP + FP} \tag{3}$$

$$Recall = \frac{TP}{TP + FN} \tag{4}$$

where:

TP: True Positive
TN: True Negative
FP: False Positive

FN: False Negative

3.2. Experimental Results

Our experiment was evaluated based on each scenario, one to nine. As there were nine scenarios (see Table 2), each scenario was repeated three times for different selected feature subsets: 20, 50, and 100 features.

a. Scenario 1

In this scenario, we leveraged Mutual Information as the feature selection and Logistic Regression as the classifier. After calculating the feature importance using MI, we define three subsets of features with 20, 50, and 100 set features. Table 3 shows the performance measurements of the scenario 1. In that Table we can see that modeling using 50 features provides the best accuracy and recall. However, this is a disguise, since the best performer is the modeling with 100 features, proved by the best F1 score, which is 92.26%. In this situation, we should be careful when looking at the performance metrics since the dataset is imbalanced. Then, by looking at the F1 score, we can identify the best performer.

Table 3. Performance Measurements on Scenario 1 (Mutual Information and Logistic Regression).

∑ Features	AUC (%)	Acc (%)	F1 (%)	Prec (%)	Recall (%)
20	82.70	90.05	91.60	93.71	90.05
50	77.52	92.03	91.99	92.22	92.03
100	81.31	90.95	**92.26**	94.22	90.95

b. Scenario 2

In the second scenario, we employed Mutual Information for feature selection and XG-Boost as the classifier. After evaluating feature importance through Mutual Information, we segmented the features into subsets of 20, 50, and 100 features. The performance metrics in Table 4 depict a compelling story. Surprisingly, the model utilizing 50 features outperformed others in terms of accuracy and recall. However, a deeper analysis revealed that the model with 100 features excelled, boasting the highest F1 score of 93.70%. This makes more sense since the more features used, the better the performance. This is also a similar pattern to Scenario 1 since the dataset subsets used are the same.

Table 4. Performance Measurements on Scenario 2 (Mutual Information and XG-Boost).

∑ Features	AUC (%)	Acc (%)	F1 (%)	Prec (%)	Recall (%)
20	74.41	94.21	92.89	92.26	94.21
50	72.63	94.95	93.56	93.89	94.95
100	78.73	94.12	**93.70**	95.45	94.12

c. Scenario 3

In the third scenario, we employed Mutual Information for feature selection and Neural Network as the classifier. Post Mutual Information analysis, we once again organized the feature into subsets: 20, 50, and 100 features. The metrics displayed in Table 5 offer intriguing insights. Initially, the model utilizing 20 features showcased superior accuracy and recall. However, a closer examination unveiled the supremacy of the 100-feature model, attaining a remarkable F1 score of 92.17%. This nuanced outcome underscores the significance of a meticulous evaluation, especially given the dataset's imbalance. Relying on the F1 score facilitated a robust comparison of the models' performance.

Table 5. Performance Measurements on Scenario 3 (Mutual Information and Neural Network).

∑ Features	AUC (%)	Acc (%)	F1 (%)	Prec (%)	Recall (%)
20	78.25	95.17	94.12	93.36	95.17
50	77.35	92.12	92.03	92.15	92.12
100	76.87	91.59	**92.17**	93.11	91.59

d. Scenario 4

In the fourth scenario, we leveraged Random Forest for feature selection and Logistic Regression as the classifier. After the feature selection process, creating subsets of 20, 50, and 100 features, we delved into the performance metrics outlined in Table 6. In this scenario, we found the expected pattern, where the performance quality gradually increased as the number of features increased. A deeper analysis revealed the prominence of the 100-feature model, showcasing an impressive F1 score of 93.36%.

Table 6. Performance Measurements on Scenario 4 (Random Forest and Logistic Regression).

∑ Features	AUC (%)	Acc (%)	F1 (%)	Prec (%)	Recall (%)
20	82.70	90.05	91.60	93.71	90.05
50	77.52	92.03	91.99	92.22	92.03
100	82.05	92.93	**93.36**	94.34	92.93

e. Scenario 5

In the fifth scenario, we utilized Random Forest for feature selection and XG-Boost as the classifier. Post feature selection, where subsets of 20, 50, and 100 features were identified, the performance metrics in Table 7 unfolded a compelling narrative. Intriguingly, the model incorporating 20 features demonstrated superior accuracy and recall. Yet, a meticulous analysis revealed the dominance of the 100-feature model, achieving a stellar F1 score of 93.61%. Given the dataset's imbalance, interpreting these results cautiously was imperative. Relying on the F1 score ensured a reliable evaluation of the model's efficacy.

Table 7. Performance Measurements on Scenario 5 (Random Forest and XG-Boost).

∑ Features	AUC (%)	Acc (%)	F1 (%)	Prec (%)	Recall (%)
20	74.41	94.21	92.89	92.26	94.21
50	72.63	94.95	93.56	93.89	94.95
100	78.59	94.00	**93.61**	94.90	94.00

f. Scenario 6

In the sixth scenario, we adopted Random Forest for feature selection and Neural Network as the classifier. Following the feature selection process, creating subsets with 20, 50, and 100 features, we analyzed the performance metrics detailed in Table 8. Surprisingly, the best performer in this Scenario is the model with 20 features. This shows that the selected 20 features are best suited to Neural Network learning and show good learning efficiency.

Table 8. Performance Measurements on Scenario 6 (Random Forest and Neural Network).

∑ Features	AUC (%)	Acc (%)	F1 (%)	Prec (%)	Recall (%)
20	77.43	94.15	**93.48**	92.87	94.15
50	76.82	93.77	93.06	92.55	93.77
100	76.68	93.50	93.16	93.26	93.50

g. Scenario 7

In the seventh scenario, we utilized AdaBoost for feature selection and Logistic Regression as the classifier. Post feature selection, organizing subsets with 20, 50, and 100 features, the performance metrics illustrated in Table 9 unfolded intriguing patterns. Unlike, previous scenarios, in this scenario, the model with 100 features showed the best accuracy, F1 score, and recall. This pattern is expected since the more features included, the better the learning itself.

Table 9. Performance Measurements on Scenario 7 (AdaBoost and Logistic Regression).

∑ Features	AUC (%)	Acc (%)	F1 (%)	Prec (%)	Recall (%)
20	82.70	90.05	91.60	93.71	90.05
50	77.52	92.03	91.99	92.22	92.03
100	82.05	92.93	**93.36**	94.34	92.93

h. Scenario 8

In the eighth scenario, we applied AdaBoost for feature selection and XG-Boost as the classifier. Following the feature selection process, generating subsets of 20, 50, and 100 features, the performance metrics outlined in Table 10 revealed intriguing findings. Initially, the model incorporating 50 features demonstrated superior accuracy and recall. However, a meticulous analysis uncovered the supremacy of the 100-feature model, achieving an exceptional F1 score of 93.61%. Due to the dataset's imbalance, careful evaluation was crucial. This scenario's pattern shows a similar pattern to Scenario 1.

Table 10. Performance Measurements on Scenario 8 (AdaBoost and XG-Boost).

∑ Features	AUC (%)	Acc (%)	F1 (%)	Prec (%)	Recall (%)
20	74.41	94.21	92.89	92.26	94.21
50	72.63	94.95	93.56	93.89	94.95
100	78.59	94.00	**93.61**	94.90	94.00

i. Scenario 9

In the ninth scenario, we leveraged AdaBoost for feature selection and Neural Network as the classifier. Post feature selection, creating subsets of 20, 50, and 100 features, the performance metrics in Table 11 revealed intriguing insights. Surprisingly, the best performer in this scenario is the model with 20 features. It shows a similar pattern to Scenario 6. Both Scenario 6 and 9 are using Neural Network for their classifier. It shows that the NN classifier can efficiently learn the smaller number of features that make the overall model lightweight.

Table 11. Performance Measurements on Scenario 9 (AdaBoost and Neural Network).

∑ Features	AUC (%)	Acc (%)	F1 (%)	Prec (%)	Recall (%)
20	77.43	94.15	**93.48**	92.87	94.15
50	76.82	93.77	93.06	92.55	93.77
100	76.68	93.50	93.16	93.26	93.50

4. Discussion

In this section, we further analyze the experimental results. We look deeper into each feature subset: 20, 50, and 100. Figure 2 shows the F1 score comparison on the models with 20 features. We can see a consistent pattern in which Neural Network always shows the best performance regardless of the feature selection method. While the Logistic Regression classifier is the worst performer for each feature selection method. Figure 3 indicates the

interesting fact that XG-Boost outperforms Neural Network at a 50-feature configuration. A similar pattern is shown in Figure 4 as well. XG-Boost is the winner for models with 100 features. In general, we can see that XG-Boost outperformed other models with more than 50 features. However, for fewer features, the Neural Network shows better F1 scores. If we compare all the scenarios, the best F1 score was achieved by Scenario 3, which leveraged Mutual Information as the feature selection method and Neural Network as the classifier. This shows that a Neural Network can be used for lightweight models since a small number of features are sufficient.

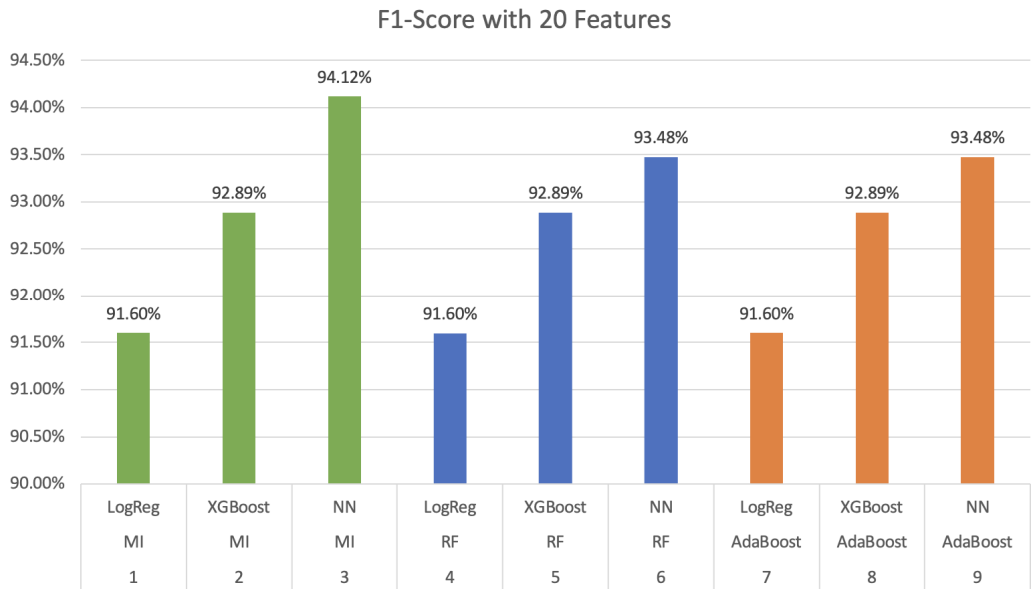

Figure 2. F1 Scores with 20 Selected Features.

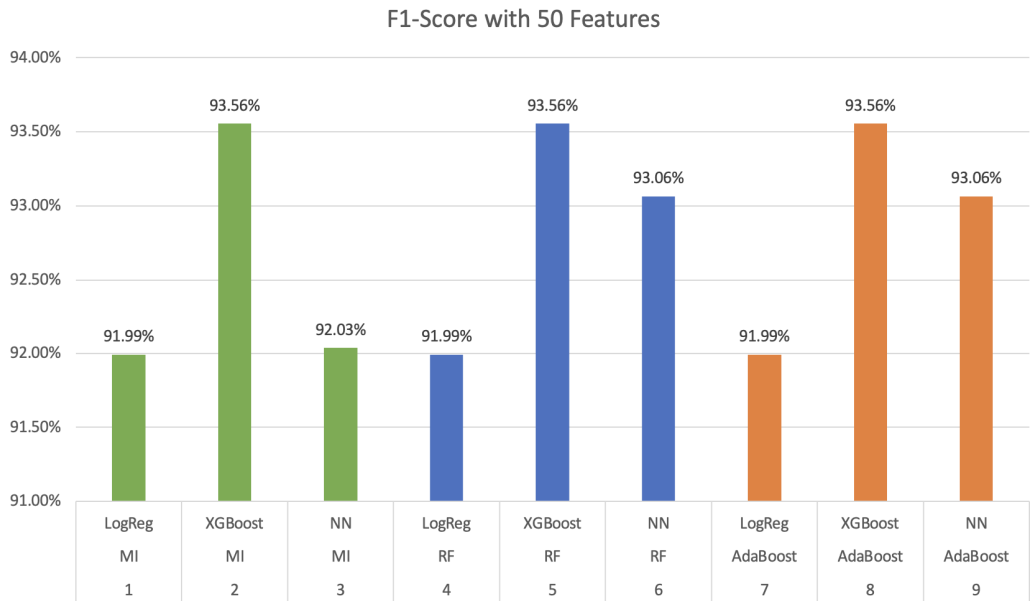

Figure 3. F1 Scores with 50 Selected Features.

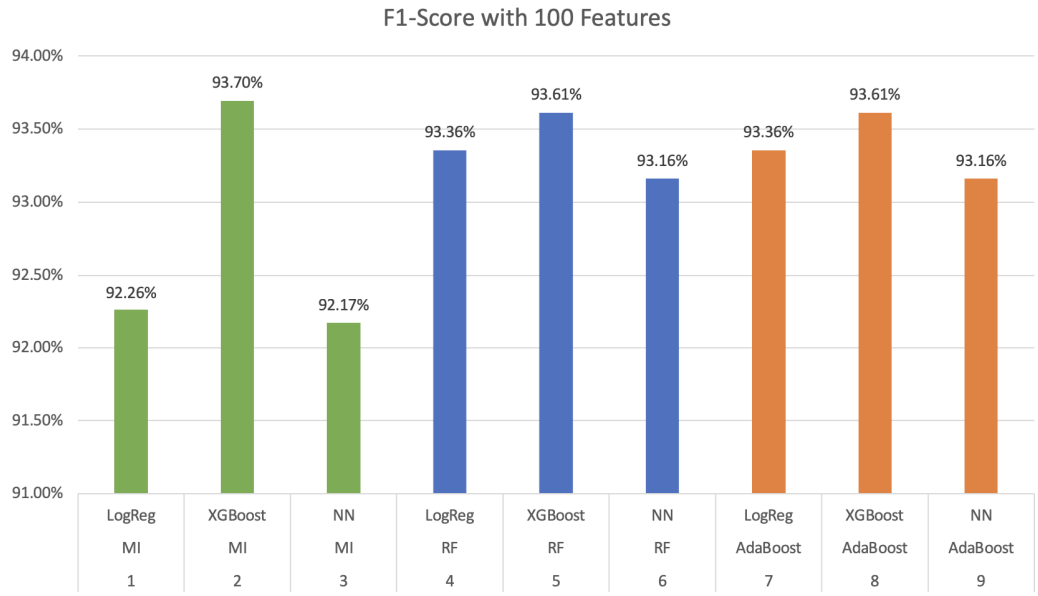

Figure 4. F1 Scores with 100 Selected Features.

The experimental outcomes demonstrate that machine learning-based IDS analysis can effectively identify cyber-attacks, potentially reducing risks and losses associated with cybercrimes. This aligns with the preventive responsibilities of the Indonesian National Police. Hence, the researchers propose the implementation of a policing model in the form of a command center containing machine learning-based IDS. While often regarded as a military facility, command centers have versatile applications in various governmental or business contexts. The term "Command Center" is also frequently used in politics to refer to a communication team monitoring and listening to media and the public, responding to queries, and synthesizing opinions to determine the best course of action. In some local governments, the term "Command Center" is replaced with City Operation Center, Control Room, and other similar names. Therefore, it is advisable to establish the Command Center around the leader's workspace. Conceptually, a Command Center serves as a source of leadership and guidance to ensure that services and order are maintained. It is not merely an information center or a help desk. Its tasks are accomplished by monitoring the environment and responding to incidents, ranging from relatively harmless situations to major crises, using predefined procedures.

The command center comprises several crucial components, with the Intrusion Detection System (IDS) being one of the most vital. The primary function of IDS is to monitor internet network traffic. To obtain this traffic data, specialized IP addresses are used as sensors to monitor internet traffic. These sensors are intentionally designed to attract attackers, and their security protocols are intentionally made weak or vulnerable to entice attackers to try to infiltrate the sensor system. The traffic recordings from these sensors are captured and analyzed, a concept known as a honeypot. This dataset can be used to train machine learning models to identify cyber-attacks. Consequently, the model can predict which instances constitute attacks and which ones do not. Once the model is trained, it can be used to detect attacks in real-time sensor data. Therefore, when an attack is identified, swift actions can be taken to mitigate potential losses and reduce risks.

The proposed machine learning model for cyber-attack detection is primarily focused on Wi-Fi network attacks, utilizing a dataset consisting of three different attack classes. This narrow scope presents certain limitations, as real-world scenarios encompass a wide variety of attack forms beyond Wi-Fi networks. Consequently, the model's applicability may be constrained when faced with diverse and evolving cyber threats. Moreover, this study serves as a proof of concept, indicating that while the results are promising, a comprehensive and robust IDS would require extensive datasets that cover a broader spectrum of attack types.

Future work should therefore aim to collect and incorporate more diverse attack samples to enhance the model's effectiveness and reliability in real-world applications.

The display in the command room can take various forms. It can represent real-time traffic on a map or focus on specific ports under attack. The features recorded during data collection from the sensors are crucial for facilitating rapid responses. Essential features to collect include IP addresses, ports, time stamps, Wi-Fi protocols, domains, packet sizes, and more. For instance, if the IDS in the command center indicates a high volume of attacks on a specific port, the operators in the command room can promptly close that port to minimize the impact, aligning with the theory of reducing rewards for criminals.

5. Conclusions

In conclusion, this research highlights the current capabilities and limitations of the Cyber Department in Indonesia Police in cyber threat prevention, emphasizing their existing cyber patrols and live threat mapping efforts but the absence of specific cyber-attack identification measures. The study demonstrates that attribute processing techniques, utilizing machine learning models and feature selection, can effectively classify Wi-Fi network attacks and predict the types of attacks in previously unidentified data. Importantly, the implementation of a machine learning-based policing model for attack identification within an Intrusion Detection System (IDS) in the command center proves beneficial. This approach enables swift responses to potential cyber-attacks, providing valuable tools to the Cyber Department of the Indonesian police for future cyber security efforts.

In future work, we plan to conduct a detailed analysis of the computational complexity, scalability, resource requirements, and potential bottlenecks of the proposed machine learning models to ensure their practical applicability and efficiency in real-time cyber-attack detection scenarios. Additionally, we intend to perform extensive experiments, including cross-validation on multiple datasets and testing on unseen data from different sources, to assess the robustness and generalization capabilities of the models. These efforts will provide comprehensive insights into the models' performance across diverse scenarios and validate their reliability in real-world applications. Furthermore, we recommend conducting an in-depth analysis of modeling within the Directorate of Intelligence and Security, considering their role in early detection and early warning as part of operational intelligence and security activities, and exploring the implementation of deep learning models to enhance classification performance.

Author Contributions: Conceptualization, S.M.; Methodology, S.M. and M.E.A.; Software, A.E.A.; Validation, M.E.A.; Formal analysis, M.E.A.; Investigation, A.E.A.; Resources, M.E.A.; Data curation, A.E.A.; Writing—original draft, M.E.A.; Writing—review and editing, S.M. and A.E.A.; Visualization, A.E.A.; Supervision, S.M.; Project administration, S.M.; Funding acquisition, S.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Directorate of Research and Development Universitas Indonesia through the Hibah Publikasi Terindeks Internasional (PUTI) Q2 Scheme under Contract NKB-771/UN2.RST/HKP.05.00/2023.

Data Availability Statement: In this study, all data were drawn from publicly available datasets, called the AWID2 dataset, sourced from IEEE 802.11 wireless networks and introduced by Kolias [2].

Acknowledgments: Many thanks for the opportunity and support to the Directorate of Research and Community Engagements Universitas Indonesia in the Q2 International Indexed Publication grant program for 2023–2024.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Kolias, C.; Kambourakis, G.; Maragoudakis, M. Swarm intelligence in intrusion detection: A survey. *Comput. Secur.* **2011**, *30*, 625–642. [CrossRef]
2. Kolias, C.; Kambourakis, G.; Stavrou, A.; Gritzalis, S. Intrusion detection in 802.11 networks: Empirical evaluation of threats and a public dataset. *IEEE Commun. Surv. Tutor.* **2015**, *18*, 184–208. [CrossRef]
3. Parker, L.R.; Yoo, P.D.; Asyhari, T.A.; Chermak, L.; Jhi, Y.; Taha, K. Demise: Interpretable deep extraction and mutual information selection techniques for IOT intrusion detection. In Proceedings of the 14th International Conference on Availability, Reliability and Security, Canterbury, UK, 2–29 August 2019; pp. 1–10.
4. Aminanto, M.E.; Choi, R.; Tanuwidjaja, H.C.; Yoo, P.D.; Kim, K. Deep abstraction, and weighted feature selection for wi-fi impersonation detection. *IEEE Trans. Inf. Forensics Secur.* **2017**, *13*, 621–636. [CrossRef]
5. Vaca, F.D.; Niyaz, Q. An ensemble learning based wi-fi network intrusion detection system (wnids). In Proceedings of the 2018 IEEE 17th International Symposium on Network Computing and Applications (NCA), Cambridge, MA, USA, 1–3 November 2018; pp. 1–5.
6. Wang, D.; He, H.; Liu, D. Adaptive critic nonlinear robust control: A survey. *IEEE Trans. Cybern.* **2017**, *47*, 3429–3451. [CrossRef] [PubMed]
7. Ran, J.; Ji, Y.; Tang, B. A semi-supervised learning approach to IEEE 802.11 network anomaly detection. In Proceedings of the 2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring), Kuala Lumpur, Malaysia, 28 April–1 May 2019; pp. 1–5.
8. Sommer, R.; Paxson, V. Outside the closed world: On using machine learning for network intrusion detection. In Proceedings of the 2010 IEEE Symposium on Security and Privacy, Berleley/Oakland, CA, USA, 16–19 May 2010; pp. 305–316.
9. Thing, V.L. IEEE 802.11 network anomaly detection and attack classification: A deep learning approach. In Proceedings of the 2017 IEEE Wireless Communications and Networking Conference (WCNC), San Francisco, CA, USA, 19–22 March 2017; pp. 1–6.
10. Kasongo, S.M.; Sun, Y. A deep learning method with wrapper-based feature extraction for wireless intrusion detection systems. *Comput. Secur.* **2020**, *92*, 101752. [CrossRef]
11. Ahmad, M.; Ramachandran, V. Cafe latte with a free topping of cracked wep retrieving wep keys from road warriors. In Proceedings of the Conference ToorCon, San Diego, CA, USA, 11–15 June 2007.
12. *airbase-ng*, 2018. Available online: https://www.aircrack-ng.org/doku.php?id=airbase-ng#how_does_the_hirte_attack_work (accessed on 9 February 2022).
13. Song, Y.; Yang, C.; Gu, G. Who is peeping at your passwords at starbucks?—To catch an evil twin access point. In Proceedings of the 2010 IEEE/IFIP International Conference on Dependable Systems & Networks (DSN), Chicago, IL, USA, 28 June–1 July 2010; pp. 323–332.
14. Beyah, R.; Venkataraman, A. Rogue-access-point detection: Challenges, solutions, and future directions. *IEEE Secur. Priv.* **2011**, *9*, 56–61. [CrossRef]
15. Tews, E.; Weinmann, R.-P.; Pyshkin, A. Breaking 104bit WEP in less than 60 seconds. In Proceedings of the International Workshop on Information Security Applications, Jeju Island, Korea, 27–29 August 2007; pp. 188–202.
16. Tews, E.; Beck, M. Practical attacks against WEP and WPA. In Proceedings of the Second ACM Conference on Wireless Network Security, Zurich, Switzerland, 16–19 March 2009; pp. 79–86.
17. Bittau, A. The fragmentation attack in practice. In Proceedings of the IEEE Symposium on Security and Privacy, IEEE Computer Society, Oakland, CA, USA, 8–11 May 2005.
18. Martínez, A.; Zurutuza, U.; Uribeetxeberria, R.; Fernández, M.; Lizarraga, J.; Serna, A.; Vélez, I. Beacon frame spoofing attack detection in IEEE 802.11 networks. In Proceedings of the 2008 Third International Conference on Availability, Reliability and Security, Barcelona, Spain, 4–7 March 2008; pp. 520–525.
19. Sawwashere, S.S.; Nimbhorkar, S.U. Survey of RTS-CTS attacks in wireless network. In Proceedings of the 2014 Fourth International Conference on Communication Systems and Network Technologies, Bhopal, India, 7–9 April 2014; pp. 752–755.
20. Meiners, L.F. But... My Station Is Awake! Power Save Denial of Service in 802.11 Networks. 2009. Available online: https://www.coresecurity.com/sites/default/files/private-files/publications/2016/05/PS_DoS.pdf (accessed on 9 February 2022).
21. Vergara, J.R.; Estévez, P.A. A review of feature selection methods based on mutual information. *Neural Comput. Appl.* **2014**, *24*, 175–186. [CrossRef]
22. Wang, R. Adaboost for feature selection, classification and its relation with SVM, a review. *Phys. Procedia* **2012**, *25*, 800–807. [CrossRef]
23. Khemphila, A.; Boonjing, V. comparing performances of logistic regression, decision trees, and neural networks for classifying heart disease patients. In Proceedings of the 2010 International conference on computer information systems and industrial management applications (CISIM), Krakow, Poland, 8–10 October 2010; pp. 193–198.
24. Chen, T.; Guestrin, C. XGboost: A scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 785–794.
25. Parsa, A.B.; Movahedi, A.; Taghipour, H.; Derrible, S.; Mohammadian, A.K. Toward safer highways, application of xgboost and shap for real-time accident detection and feature analysis. *Accid. Anal. Prev.* **2020**, *136*, 105405. [CrossRef] [PubMed]
26. Dave, V.S.; Dutta, K. Neural network-based models for software effort estimation: A review. *Artif. Intell. Rev.* **2014**, *42*, 295–307. [CrossRef]

27. Haykin, S. Neural Networks and Learning Machines. 3/E, Pearson Education India. 2009. Available online: https://books.google.co.id/books/about/Neural_Networks_and_Learning_Machines.html?id=KCwWOAAACAAJ&redir_esc=y (accessed on 9 February 2022).
28. Abiodun, O.I.; Jantan, A.; Omolara, A.E.; Dada, K.V.; Mohamed, N.A.; Arshad, H. State-of-the-art in artificial neural network applications: A survey. *Heliyon* **2018**, *4*, 11. [CrossRef] [PubMed]