Evaluating the Performance of Machine Learning Classifiers for Network Intrusion Detection: A Comparative Study Using the UNSW-NB15 Dataset

Iwan Handoyo Putro

Electrical Engineering Department, Petra Christian University, Surabaya, East Java, Indonesia
E-mail: iwanhp@petra.ac.id

(Received: 4 Jun 2025, revised: 24 Jun 2025, accepted: 25 Jun 2025)

Abstract

Network security has become a critical concern in digital data transmission. It is because of their growing adoption and complexity of cyber-attacks. Therefore, protecting network infrastructures and identifying malicious behavior becomes a necessity. This paper gives a comparative performance analysis of multiple machine learning (ML) classifiers for intrusion detection systems (IDS) by using the UNSW-NB15 dataset. To gain better insight into the IDS performances, several ML classifiers are being assessed. This includes the Decision Tree (DT), Random Forest (RF), Support Vector Machine (SVM), k-Nearest Neighbors (k-NN), Naïve Bayes (NB), and Gradient Boosting (XGB). The performance matrix in the analysis comprises the training score, accuracy, precision, recall, F1-score, training time, and AUC-ROC. The results reveal that DT and RF scored the highest training marks of 99.77%. Regarding accuracy, the RF model achieves the highest percentage at 95.05%. In terms of the computational time, *k-NN* displayed the lowest training time at 0.01 seconds. These analysis results provide guidance to the selection of appropriate ML-based cyberattack classification. It also provides insights for further research in ML-based cybersecurity systems to support the development of intelligent and efficient IDS solutions.

Keywords: Machine Learning, Intrusion Detection Systems, Feature Selection, UNSW-NB15, Models Comparison

I. INTRODUCTION

The explosive growth of the digital networks poses a threat to internet-connected devices. For industries and organizations that depend on this digital infrastructure, this situation leads to important concerns. Designed to monitor network traffic and find illegal access or malicious activity, intrusion detection systems (IDS) are an essential line of defense. Nonetheless, conventional rule-based IDS systems may lack the flexibility required to identify dynamic changes and recent cyber threats. As such, integrating machine learning (ML) techniques with IDS is necessary to allow more dynamic and intelligent intrusion detection [1].

The 2017 Equifax data hack is an alarming example of the vital part intrusion detection systems (IDS) play in modern cybersecurity incident [2]. Equifax had installed an IDS that had configured properly to identify attempts at exploitation against known vulnerabilities. However, a crucial error an expired SSL certificate have blinded the system to encrypted communication.

This unmonitoring security fault let attackers remain undetectable for more than 76 days and exfiltrate personal data of around 147 million people. This event emphasizes how important constant monitoring, appropriate configuration, and timely maintenance of IDS by itself is not enough.

Moreover, this threat emphasizes the possible benefits of including ML-based IDS solutions. This would have highlighted unusual internal traffic or data exfiltration activities even in the lack of signature-based detection. This Equifax data breach shows that emphasizing the requirement of proactive, smart, and strong IDS strategies are needed. Finally, how the failure of managing conventional IDS may have led to disastrous outcomes.

ML algorithms can improve intrusion detection by making it more intelligent and automated. It operates by automating pattern recognition, anomaly detection, and behavioral classification [3]. By learning from the network traffic, the ML model could identify deviations from normal data transmission that may indicate malicious activities [4]. This automation procedure could potentially reduce the reliance of the security system on defined rules.

However, the performance of ML-based IDS differs significantly depending on elements like the dataset properties, feature representation, and method of algorithm as well as the parameters chosen [5]. In addition, network attacks are

DOI: 10.34148/teknika.v14i2.1276

becoming increasingly sophisticated and intelligent to bypass conventional IDS [6]. Therefore, determining the most efficient ML models for practical intrusion detection uses depends on doing a thorough comparative analysis of many classifiers.

Developed by the Australian Centre for Cyber Security (ACCS), the UNSW-NB15 dataset [7] incorporates a wide range of attack scenarios and more realistic network traffic. This dataset is a useful tool for benchmarking the performance of ML-based IDS since it resembles real-world network security environments.

The aim of this study is to present a comparison analysis to highlight the strengths and weaknesses of ML models in detecting and classifying normal and attack network activities using the UNSW-NB15 dataset. We empirically evaluate five often used ML classifiers: Decision Tree (DT), Random Forest (RF), k-Nearest Neighbors (*k*-NN), Naïve Bayes (NB), and Gradient Boosting (XGB). Training time, accuracy, precision, recall, F1-score, area under the ROC curve (AUC), and training score define numerous performance criteria used in assessment of these classifiers.

The remainder of this work is arranged as follows: Section 2 reviews state-of-the-art research on the domain of ML-based IDS. Section 3 presents the research approach, feature selection process, and role of a confusion matrix for performance matrix evaluation. Section 4 details the experimental setup and libraries used in this work. Section 5 provides a comprehensive performance assessment and discusses the experimental results. Finally, Section 6 concludes this study and outlines potential direction for future work.

II. RELATED WORK

ML-based IDS has gained a lot of interest since it addresses the shortcomings of conventional rule-based methods [8]. This conventional approach often fails to identify zero-day and sophisticated attacks [9]. Using ML algorithms, IDS can automatically learn from network traffic both normal and malicious patterns [10]. As such, it enables more intelligent and immediate threat detection.

Research in this ML-based IDS domain covers a range of learning paradigms, including supervised, unsupervised, and hybrid approaches [11]. Each has unique benefits in terms of detection capabilities and computational cost. Intrusion detection studies have made substantial use of supervised learning methods including the DT, RF, Support Vector Machine (SVM), and *k*-NN. Moustafa and Slay [12] utilized five algorithms: DT, LR, NB, ANN and EM in their study to evaluate network anomaly detection. The reason behind selecting these ML models is its various characteristics to learn and evaluate the dataset. The results of their study demonstrates an enhance performance of the DT models in identifying attack patterns on the UNSW-NB15 dataset.

Kasongo and Sun [13] examined the IDS using a feature selection approach on the UNSW-NB15 dataset. Their study used feature priority ranking to reduce the dimensionality of the dataset. This process resulted in the selection of 19 attributes for the next step of the experiment. In this study, they implemented the SVM, DT, Artificial Neural network (ANN), *k*-NN, and Logistic Regression (LR) algorithms. The SVM was choosen as its it tends to perform effectively for high dimensional input spaces. The DT model adopted in their study because this specific algorithm provides less model complexity to interpret by the user. The *k*-NN method included in this IDS study because of the ability of this method to operate in supervised manner. Their study results show an increase in the model accuracy after conducting a feature selection stage.

Using UNSW-NB15, Ahmad et al. ran IDS with consideration for the MQTT, Flow, and TCP categories of the dataset [14]. They adopted supervised ML techniques including the ANN, SVM, and RF models. These ML models where choosen based on their capability to operate in supervised learning. Moreover, these three models can be implemented in both binary and multi-class classification. As far as the results are concerned, the RF model achieves the highest percentages of binary and multi-class classification, with 98.67% and 97.54%, respectively.

Disha and Waheed [15] applied the Chi-square test methods for selecting features prior to ML training. By dropping features with p-value higher than 0.05, seven features were removed from this dataset. After that, several ML and DL methods were employed (DT, AdaBoost, GradientBoosting, Multilayer Perceptron, Long Short-term Memory, and Gated Recurrent Unit). The reason for selecting these methods was to enable a comparative analysis across multiple machine an deep learning algorithms. This comparison was also conducted with the objective of evaluating their effectiveness in detecting intrusions on the characteristic of UNSW-NB15 as an imbalanced dataset. Further training using various ML and DL algorithms shows that the DT model outperforms other algorithms with an accuracy of 93.01% using selected features.

These previous studies have utilized the UNSW-NB15 dataset to evaluate the effectiveness of machine learning approaches in IDS study. It recognizes its relevance in simulating realistic and contemporary network traffic as well as inclusion of diverse attack types.

While many of these studies focused on a limited subset of classifiers, our study expands upon this foundation by employing a broader range of machine learning algorithms. These approach could represent various ML algorithms that available for cybersecurity study. This includes statistical methods (e.g., NB), tree-based models (e.g., DT, RF), ensemble techniques (e.g., XGB), and lazy learning method (e.g., *k*-NN).

By adopting these wide spectrum of learning paradigms, our study provides a more comprehensive evaluation of IDS performance and generalization across different types of detection strategies. This broader experimental scope contributes to a deeper understanding of model suitability for real-world IDS deployment. It also reinforces the importance of using a modern datasets like UNSW-NB15 for meaningful performance benchmarking.

DOI: 10.34148/teknika.v14i2.1276

III. METHODOLOGY

A. Dataset and Preprocessing Stage

This IDS experiment utilized the UNSW-NB15 dataset. It was coosen as its suitable to modern real-world network traffic patterns. Unlike previous databases like KDD99 or NSL-KDD, which are today regarded as obsolete and excessively processed.

UNSW-NB15 combines a varied mix of modern attack forms and legal traffic flows generated with modern tools. This makes it a more realistic benchmark for measuring the performance of ML based IDS. Therefore, allowing researchers to evaluate model effectiveness under situations more closely reflecting pragmatic deployment environments.

Before conducting experiment, the dataset need to be prepared. This is important as most dataset often incomplete or inconsistent. As such, we to clean the dataset to make sure that our models does not lead to bias. The original dataset contains 44 features and 1 target class. The target class, called label, indicating that the network traffic classified as noral or malicious behaviour. It contains numerical figure as 0 and 1. Zero indicates normal traffic, while 1 indicates attack attempts.

First, non-essential features such as id and timestamps were removed, as they do not contribute to the predictive task and may introduce noise. Features that having object data type, such as protocol type (e.g. tcp, udp, arp, etc), service (e.g. smtp, ftp, http, ssh, etc), and state (i.e. FIN, CON, or INT), were transfored into numerical format using one-hot encoding technique. This is to maintain compatibility with the ML algorithms used in this study. While the dataset dimensionalty remains unchanged, we can preserve the categorical distinctions without introducing ordinal bias.

After that, we are handling the missing value in the dataset. The incomplete value in the dataset might be happened due to sensor failure when producing data. This lack of information can also be produced when errors occuring at data entry stage.

As the UNSW-NB15 dataset is relatively clean, no imputation was necessary. Numerical features were standardized using StandardScaler fro scikit-learn library. This is to ensure all attributes had zero mean and unit variance and contriibute eqaually to the models. In addition, this step was particularly important for distance-based classifiers such as *k*-NN and for gradient-based optimization methods used in algorithms like SVM and XGB to avoid bias towards high-scales features [16].

B. Proposed Experiment

Figure 1 describes the flow of proposed experiment in this IDS study. This study utilized the UNSW-NB15 dataset. The malicious behaviour that includes in this dataset are DoS, Exploits, Fuzzers, Reconnaissance, Shellcode, Fuzzers, Analysis, Generics, and Worms. For making the ML evaluation easier, in this IDS dataset these attacks have labeled into the normal and attack attempts using categorical value.

The data preprocessing stage, as mentioned earlier, involved several sub process. This includes sub process, such as: cleaning null or irrelevant values and encoding categorical attributes using one-hot label encoding. As a final step, we conduct numerical features normalization to ensure consistent input for the ML training.

The normalization process has been proven as beneficial to decreased the models' error rate. Karatas [17] in his study of IoT attack detection implementing normalization and standardization process to improve ML performances. The results indicate that by conducting these process in this IDS study, the error rate of LR and SGD models can be reduced.

Other important aspect of ML workflow is to conduct one hot encoding. This is important because feature in the dataset might introduce ordinal relationship that possibly mislead the ML models. Moreover, most ML models (for example: LR, SVM, *k*-NN, and NB), could only process numeric inputs as such this one hot encoding become beneficial in order to make these ML models operate successfully.

Hussein, Falcarin, and Sadiq [18] in their study using IoTID20 dataset utilizing one hot encoding technique particularly in RF model. The results of their study shows that by implementing one hot encoding techniques the accuracy of RF model can be improved.

The feature selection stage was performed using Pearson correlation coefficient (PCC) analysis to evaluate linear relationships between features. A correlation threshold of 0.98 was applied to eliminate highly redundant features, and Recursive Feature Elimination (RFE) was used to refine the subset. This process resulted in 19 relevance features that contribute significantly to classification performance.

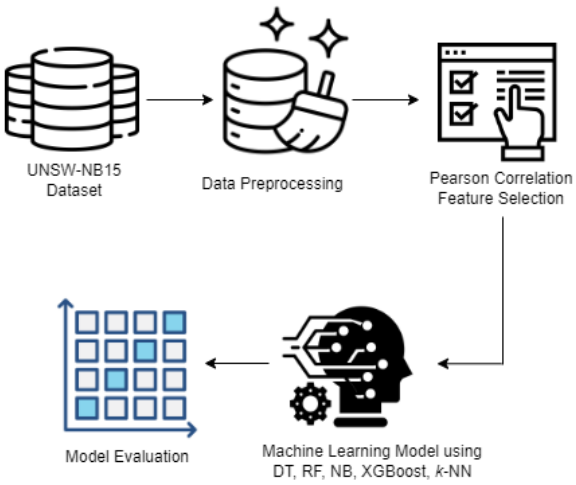

Figure 1. The Proposed Experiment

Five ML classifiers were selected for this study: DT, RF, *k*-NN, NB, and XGB. The reason of choosing these models does not relate to the dataset characteristics but merely to their widely used in the ML based IDS studies.

Moreover, these models were chosen to represent a diverse set of learning algorithms. This allows for a comprehensive evaluation of IDS performance across different learning paradigms.

DT and RF are tree-based models known for their interpretability and robustness [19]. In particular, RF model capable to address overfitting issues by aggregating the

DOI: 10.34148/teknika.v14i2.1276

predictions of multiple randomized trees. *k*-NN, as an example of lazy learner algorithm, capable to classify observations based on similarity [20]. It makes this specific model being effective for detecting localized patterns in network traffic.

NB, a statistical model based on Bayes' theorem, offers computational efficiency [21]. It has a strong baseline performance while working towards high-dimensional feature spaces. The last model, XGB, shows a promising ensemble method based on gradient boosting [22]. It is included for its proven ability to deliver high predictive accuracy and handle complex, non-linear feature interactions.

The inclusion of these various classifiers ensures a balanced comparison between simple, interpretable models and more advanced methods. This will be a standard requirement for ML algorithms commonly used in contemporary IDS research.

To assess models' performances, the calculation of accuracy, precision, recall, F1-score is done. In order to gain better insight into the classfication models, we also measure the area under the ROC Curve (AUC) and comparing training time. All of these calculation were derived from the confusion matrix of each classifier. This models evaluation aims to identify the algorithm that efficiently distinguish between normal and attack attempts.

In this experiment we are not conducting any hyperparameters tuning to these ML models. In other words, each of the models using their default paramaters. The reason for not run this tuning process is because we do not comparing the results with any state of the art IDS system that utilize hyperparameters tuning. Furthermore, we aim to focus on the strength and limitation of the ML models being applied.

C. Pearson Correlation Feature Selection

For the purposes of this study, the PCC and RFE are combined sequentially to perform feature selection. This approach was taken to identify the most relevant features from the UNSW-NB15 dataset.

The PCC measures the linear relationship between pairs of variables, producing values between -1 and 1, where values near ±1 indicate a strong linear correlation. In order to address multicollinearity, a conservative threshold of 0.98 was applied. Following this procedure, the RFE, a wrapper-based feature selection method, was employed to iteratively select features that contribute most to model performance.

By combining PCC with RFE, the feature set was efficiently reduced from the original 44 attributes to a subset of 19 features. These selected features keeping high relevance to the target class. This potentially gives the ML models sufficient information to achieve high accuracy and generalizability. In addition, this new set of features balances dimensionality reduction with the preservation of critical information to enhance model interpretability and reduce computational cost.

Overall, this hybrid feature selection optimizes the trade-off between eliminating redundant data and preserving essential information. Therefore, it helps to facilitate efficient and effective intrusion detection in complex network environments.

D. Confusion Matrix

In order to perform an assessment of the ML model performances, we need to use a confusion matrix. This assessment matrix serves as a visual representation of the ML models' prediction. This matrix consists of four components: True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN). Through these four components, further calculations can be done to provide a better understanding of the performances of ML models.

The first evaluation is accuracy, which provides us with the proportion of correct predictions from the model. This particular assessment is beneficial when the dataset is balanced. To quantify the accuracy value, we need to calculate the ratio of all correct predictions and then divide it by the total number of cases. This can be done in Equation 1.

$$Accuracy = \frac{(TP+TN)}{(TP+TN+FP+FN)} \quad (1)$$

The second evaluation in this confusion matrix is the Precision. It is used as an indicator of how well the model is able to identify positive identifications that were actually correct. This Precision value reflects the model's ability to avoid false alarms. Equation 2 is used to calculate Precision value.

$$Precision = \frac{TP}{(TP+FP)} \quad (2)$$

Recall, also known as sensitivity or true positive rate, give information on the ratio of actual positive cases that the model correctly identifies. This evaluation is considered important in IDS study as getting all relevant cases is crucial. Equation 3 is used to calculate a Recall value.

$$Recall = \frac{TP}{(TP+FN)} \quad (3)$$

The F1 score is helpful when dealing with an imbalanced dataset. It gives the harmonic mean of precision and recall and provides a balanced metric that considers both false positives and false negatives. Equation 4 is formulated to measure the F1-score.

$$F1 - score = 2\,x\,\frac{(Precision\ x\ Recall)}{(Precision+Recall)} \quad (4)$$

These four metrics offer an in-depth evaluation of ML classifier performance. This is particularly essential in ML-based IDS, where minimizing both false alarms and missed detections is critical. These four metrics derived from the confusion matrix were calculated for all classifiers used in this experiment to provide objective performance comp-arison.

IV. EXPERIMENT ENVIRONMENT

This study utilizes the UNSW-NB15 dataset, supplied by the Intelligent Security Group at UNSW Canberra, Australia. The UNSW-NB15 dataset possesses the advantages of being comparatively compact and exhibiting minimal data

DOI: 10.34148/teknika.v14i2.1276

redundancy. These benefits render the data pre-processing phase more viable. The UNSW-NB15 dataset is a widely recognized benchmark dataset designed for network intrusion detection research. It was generated in a realistic testbed environment using the IXIA PerfectStorm tool to simulate a mix of normal and malicious network traffic.

The dataset provided was already divided into training and testing subsets. The number of training set is 175.341, while the testing set is 82,332. The training set was used for model learning, while the testing set was allocated for ML performance evaluation. This dataset split was conducted in order to ensure a fair and consistent comparison of classifier generalizability and robustness to unseen data.

The UNSW-NB15 dataset consists of nine types of network attacks. This includes DoS, Exploits, Reconnaissance, Backdoors, Analysis, Worms, Shellcode, Fuzzers, and Generic attacks. This comprehensive malicious behavior provides realistic conditions for the network. In addition, the labels classify each record as either benign or one of the attack types, enabling multi-class classification and detailed performance analysis.

For this experiment, we run Python scripts on the Windows 11-based PC. To perform ML-based analysis, we implemented ML-based libraries for Python environments, such as Pandas, Numpy, Matplotlib, Seaborn, and Sklearn. As far as the hardware specification is concerned, we conducted the experiment on PC with CPU Intel I5-13500, 32 GB DDR5 memory, 1 TB NVme, and Nvidia GPU RTX 3060.

V. RESULTS AND DISCUSSION

This result and discussion section presents the experimental outcomes from evaluating various ML classifiers using the UNSW-NB15 dataset. The aim of this analysis is to compare the strengths and limitations of each ML classifiers. The analysis results are interpreted in the context of both the statistical performance indicators and the security implications they suggest. This is to refine our understanding to the underlying detection behaviors exhibited by each classifier.

The PCC heatmap, as shown in Figure 2, offers a quantitative summary of the linear relationships among the various features in the UNSW-NB15 dataset. This first analysis conducted to explote the features electio step in this IDS work. The heatmap reveals that the majority of features reveal weak linear correlations with the target label. This is align to our expectation that in complex network datasets the relationships between features and outcomes are often non-linear.

However, several features from the dataset, such as ct_dst_ltm, ct_srv_src, and ct_dst_sport_ltm, demonstrate relatively higher correlation with the target label. These features are associated with the frequency and distribution of connections. It suggests that behavioral characteristics of traffic may lead to anomalous activity than merely plain flow-level statistics.

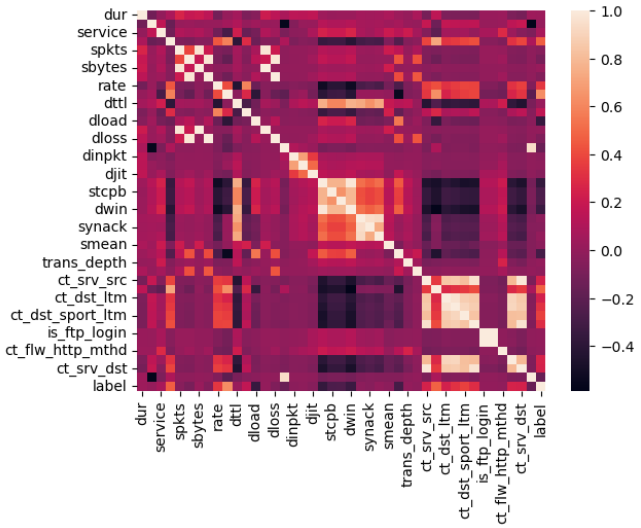

Figure 2. The Pearson Correlation Result

While much focus is placed on positively correlated features, the presence of negative correlations also provides useful insights. Features such as is_ftp_login and rate, or ct_flw_http_mthd and service, exhibit moderate negative correlations. These inverse relationships suggest that certain protocol behaviors may counter-indicate others. For instance, an increased rate of packets may be less likely when specific FTP login attempts occur, reflecting distinct behavioral patterns across traffic classes. Incorporating this understanding into model design, such as through engineered interaction terms or hybrid rule-learning techniques, can enhance classifier sensitivity.

Lastly, feature grouping based on internal correlations and their relationships with the label suggests the presence of semantically meaningful domains within the dataset. Features related to connection duration, packet size, and connection counts often form clusters that are collectively important for differentiating between normal and malicious traffic. Binary features such as is_ftp_login and is_sm_ips_ports also show strong associations with attack labels, indicating their utility in detecting specific types of intrusions. Overall, the correlation analysis highlights not only the most relevant features but also the structural interdependencies that can inform effective feature engineering and model development in the IDS.

Table 1 The ML Classifiers Results

ML Models	Training Score	Accuracy	Precision	Recall	F11-score	Training Time
	%	%	%	%	%	seconds
DT	99.77	93.65	95.06	95.01	95.03	4.39
RF	99.77	95.05	96.33	95.98	96.15	73.06
NB	83.42	83.52	84.06	91.61	87.67	0.13
XGB	93.34	93.44	94.54	95.25	94.89	156.09
k-NN	94.50	91.57	93.60	93.19	93.39	0.01

The second analysis is comparing through the training score, testing scores and training time across all models provides important insights into their generalization capabilities. Table 1 outlines the results from all of these ML models. The ML models that demonstrate a high training score

DOI: 10.34148/teknika.v14i2.1276

alongside a notably lower test accuracy often exhibit symptoms of overfitting. This pattern is evident in the performance of the DT and RF classifiers. Both achieved near-perfect training scores, yet their test accuracies were comparatively reduced. Such a discrepancy suggests that these models may have captured noise or specific patterns unique to the training data, leading to decreased effectiveness when exposed to unseen instances. In intrusion detection systems (IDS), overfitting can be particularly problematic [23], as it reduces a model's adaptability to evolving threats and real-world variability. Therefore, despite their strong initial learning capacity, overly complex models may require regularization or pruning techniques to enhance generalizability.

Conversely, the NB classifier exhibited relatively low scores on both training and testing datasets, indicating potential underfitting. Underfitting arises when a model is too simplistic to capture the underlying structure of the data. In the case of NB, its assumption of feature independence [24] may not suit the intricate and correlated nature of the features within the UNSW-NB15 dataset. This results in the model failing to effectively differentiate between normal and malicious traffic patterns. In cybersecurity contexts, such underperformance can lead to higher false negatives, allowing intrusions to go undetected. Thus, models with limited representational power may not be appropriate for complex, high-stakes environments like IoT-based networks.

In contrast, models such as *k*-NN and XGB displayed a balanced relationship between training score and test accuracy, suggesting a strong generalization ability. These classifiers managed to learn significant patterns during training without overfitting, which translated into stable and consistent performance during evaluation. Good generalization is crucial in IDS applications [25], where models must accurately identify both known and novel attack behaviors across diverse traffic conditions. The minimal gap between training and testing performance in these models indicates their suitability for deployment in dynamic and unpredictable environments. This insight reinforces the value of evaluating both training and test metrics holistically when selecting classifiers for security-critical systems.

The comparative evaluation of machine learning classifiers on the UNSW-NB15 dataset demonstrates significant variation in predictive performance and computational efficiency. The RF classifier achieves the highest accuracy of 95.05%, precision of 96.33%, recall of 95.98%, and an area under the ROC curve (AUC) of 0.99. These results indicate that RF is highly effective at distinguishing between benign and malicious network traffic, making it well-suited for practical IDS implementations.

Similarly, the XGB model exhibits competitive performance with an accuracy of 93.44% and an identical AUC of 0.99. However, the training time of 156.09 seconds is considerably longer compared to RF and other classifiers, reflecting the computational overhead of gradient boosting techniques. Despite this, XGB's robust precision and recall metrics underscore its potential value in scenarios where detection accuracy is prioritized over training speed.

The DT classifier achieves a high training score of 99.77% and an accuracy of 93.65%, with a relatively low training time of 4.39 seconds. This suggests a favorable balance between predictive capability and computational demand, making the DT a practical option for environments requiring efficient model retraining without significant compromise on accuracy.

Among the classifiers evaluated, *k*-NN offers the fastest training time of 0.01 seconds. While its accuracy (91.57%) and AUC (0.97) are marginally lower than those of RF and XGB, *k*-NN remains a viable choice for applications necessitating rapid model updates or real-time responsiveness, where computational resources or latency constraints are critical.

In contrast, the NB classifier demonstrates the lowest performance, with an accuracy of 83.52% and precision of 84.06%. Although NB benefits from a brief training time of 0.13 seconds, its comparatively lower precision and accuracy suggest limitations in capturing the complex, nonlinear patterns characteristic of network traffic data. Consequently, the NB may be less suitable for IDS applications requiring high detection reliability and low false alarm rates.

Overall, the results indicate that ensemble-based classifiers such as RF and XGB provide superior detection performance but incur greater computational costs. In comparison, DT and *k*-NN classifiers offer faster training times with a modest reduction in accuracy, enabling flexible model selection tailored to specific operational requirements and resource constraints.

The third discussion is by analyzing the AUC curve. The significant disparity in AUC values between the RF model and the NB classifier provides important insights into their relative suitability for intrusion detection in complex network environments.

The RF model attains an outstanding AUC of 0.99 as displays in Figure 3. This demonstrates its exceptional capability to distinguish between benign and malicious traffic across a wide range of classification thresholds. This high discriminative power is largely due to the RF's ensemble learning approach, which combines multiple decision trees to capture intricate, nonlinear relationships and feature interactions present in network traffic data. Such complexity is typical in intrusion detection datasets like UNSW-NB15, where attacks may exhibit subtle and varied patterns.

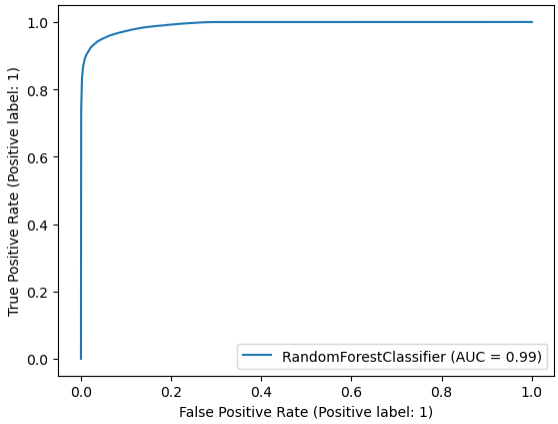

Figure 3. The AUC Plot of the RF Model

DOI: 10.34148/teknika.v14i2.1276

In contrast, the NB classifier records a considerably lower AUC of 0.89 as shown in Figure 4. This reduction in performance can be attributed to NB's foundational assumptions of feature independence and Gaussian-distributed inputs, which often do not hold true for real-world network traffic. These simplifying assumptions limit NB's ability to effectively model the data's underlying structure, resulting in reduced classification precision and recall.

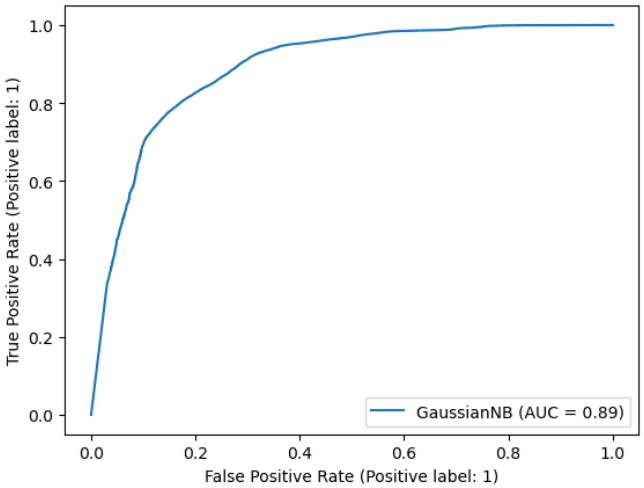

Figure 4. The AUC Plot of the NB Model

Moreover, the lower AUC for NB implies a greater likelihood of misclassifying network events, either by generating false alarms or by failing to detect actual intrusions. In operational settings where minimizing false positives and false negatives is critical, such limitations could hinder the effectiveness of an IDS and increase the burden on security analysts.

Conversely, the RF's superior AUC reflects not only its accuracy but also its robustness across various decision thresholds, making it a reliable choice for detecting a broad spectrum of attack types with minimal error. While NB offers advantages in terms of computational simplicity and faster training times, these benefits are outweighed by its reduced predictive performance in complex and dynamic network environments.

This comparative analysis highlights the trade-offs inherent in selecting ML-based IDS. Ensemble methods like RF, although computationally more intensive, deliver enhanced detection capabilities by accommodating the complex, multi-dimensional nature of network traffic.

Nonetheless, simpler probabilistic models such as NB, despite their speed and ease of implementation, may be inadequate for capturing the nuanced behaviors exhibited by modern cyberattacks. Therefore, the findings emphasize the necessity of balancing detection performance with operational constraints when deploying IDS solutions, especially in high-stakes contexts such as IoT networks where accurate and timely threat identification is paramount.

To sum up, the marked difference in AUC values underscores the critical role that model selection plays in developing effective intrusion detection systems. Future research should continue to explore hybrid approaches that combine the robustness of ensemble models with the efficiency of simpler classifiers, as well as investigate advanced feature engineering and data preprocessing techniques to further improve detection accuracy without prohibitive computational costs. This will be essential to addressing evolving cyber threats in increasingly complex network environments.

The last analysis is by confirming the result from confusion matrix. This analysis provides a comprehensive evaluation of the RF and NB classifiers' performance in detecting intrusions within the UNSW-NB15 dataset.

The RF model correctly identifies 26,085 TN instances (as shown in Figure 5), substantially outperforming the NB model, which correctly identifies only 19,221 benign instances. This superior ability to recognize legitimate traffic translates into a significantly lower FP rate for RF, with just 1,791 benign samples misclassified as malicious, compared to 8,655 for NB.

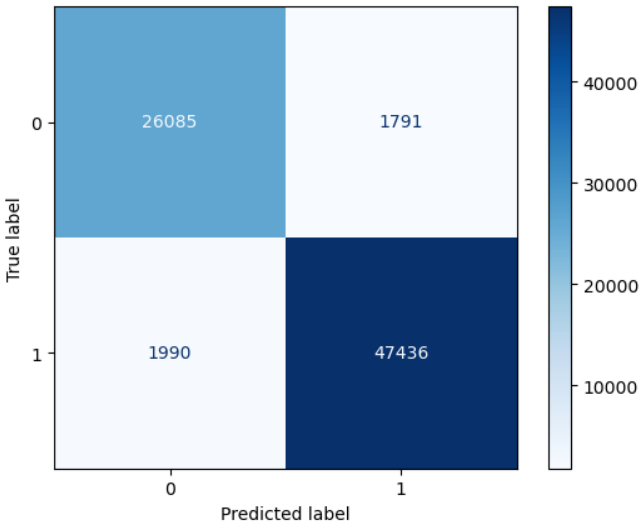

Figure 5. Confusion Matrix of the RF Model

Minimizing FP is essential in IDS, as excessive false alarms can overwhelm security personnel, reduce trust in automated alerts, and increase operational costs. Therefore, the RF's marked reduction in false alarms underscores its practical advantage in real-world deployments.

In terms of detecting actual attacks, the RF classifier demonstrates enhanced sensitivity, correctly detecting 47,436 TP instances, compared to 45,544 by NB (as seen in Figure 6). Additionally, the RF model exhibits a substantially lower FN count of 1,990, nearly half of NB's 3,882 missed attack instances. FN are especially critical in cybersecurity contexts, as undetected malicious activity can lead to severe breaches and system compromises. Consequently, the RF's ability to minimize FN while maintaining high true positive rates highlights its robustness and reliability in identifying diverse and sophisticated attack types represented in UNSW-NB15.

This improved balance between sensitivity (true positive rate) and specificity (true negative rate) observed in the RF model is indicative of its capability to handle the complex,

DOI: 10.34148/teknika.v14i2.1276

nonlinear feature interactions present in modern network traffic. In contrast, the NB model's assumptions of feature independence and Gaussian-distributed features limit its effectiveness, as reflected in its higher rates of both false positives and false negatives. These limitations reduce NB's practical utility for IDS, particularly in environments requiring stringent detection accuracy and minimal operational disruptions.

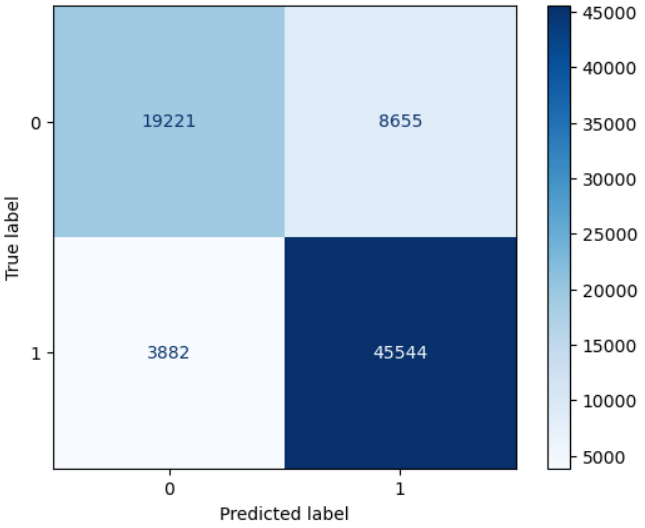

Figure 6. Confusion Matrix of The NB Model

Integrating these findings with the overall model evaluation, it is evident that RF's ensemble learning approach provides significant advantages in managing the trade-off between detection accuracy and false alarm reduction. While NB offers computational efficiency and simplicity, these benefits are overshadowed by its comparatively lower classification performance. Therefore, the RF emerges as a more suitable candidate for deployment in security-sensitive environments such as IoT networks and enterprise systems, where the cost of missed detections and false alerts can be substantial.

In summary, the confusion matrix results reinforce the conclusion that selecting robust ML classifiers like the RF is critical for effective intrusion detection in contemporary network settings. Future work should consider hybrid and ensemble-based approaches to further enhance detection capabilities while optimizing computational efficiency, ensuring scalable and resilient IDS solutions capable of addressing evolving cyber threats.

VI. CONCLUSION AND FUTURE WORK

This IDS study investigated the ML-based performances on the UNSW-NB15 dataset. The performance matrix measures parameters, such as accuracy, precision, recall, F1-score, AUC-ROC, and confusion matrix analysis. The comparison results indicate that the RF model outperforms other ML algorithms in this experiment. The RF model benefits from their ensemble algorithm to accurately identify between normal and attack patterns. Moreover, it shows strong and balanced performances across all assessment criteria.

Simpler models like NB revealed restrictions in handling complicated traffic patterns. Although it is easier to train, it performs poorly among other ML classifiers. When computational resources are not a significant concern, the *k-NN* model may be suitable for this situation due to its fast training time.

These results highlight the need of using strong classifiers and applying a comprehensive set of evaluation measures to guarantee correct and reliable intrusion detection in contemporary network environments. For future work, we are planning to utilize broader range of IDS datasets to gain better insight into the capabilities and limitations of ML-based IDS.

REFERENCES

[1] G. Kocher and G. Kumar, "Machine learning and deep learning methods for intrusion detection systems: recent developments and challenges," *Soft Computing,* vol. 25, no. 15, pp. 9731-9763, 2021.

[2] I. Kabanov and S. E. Madnick, "A Systematic Study of The Control Failures in the Equifax Cybersecurity Incident," *SSRN,* vol. 2020-19, pp. 1-25, 2020.

[3] A. B. Nassif, M. A. Talib, Q. Nasir, and F. M. Dakalbab, "Machine Learning for Anomaly Detection: A Systematic Review," *IEEE Access,* vol. 9, pp. 78658-78700, 2021.

[4] A. Alshammari and A. Aldribi, "Apply machine learning techniques to detect malicious network traffic in cloud computing," *Journal of Big Data,* vol. 8, no. 1, 2021.

[5] A. Thakkar and R. Lohiya, "A survey on intrusion detection system: feature selection, model, performance measures, application perspective, challenges, and future research directions," *Artificial Intelligence Review,* vol. 55, no. 1, pp. 453-563, 2022.

[6] A. Heidari and M. A. Jabraeil Jamali, "Internet of Things intrusion detection systems: a comprehensive review and future directions," *Cluster Computing,* vol. 26, no. 6, pp. 3753-3780, 2023/12/01 2023.

[7] N. Moustafa and J. Slay, "UNSW-NB15: A Comprehensive Data set for Network Intrusion Detection systems (UNSW-NB15 Network Data Set)," in *Military Communications and Information Systems Conference (MilCIS)*, Canberra, ACT, Australia, 2015, pp. 1-6.

[8] Q. Liu, V. Hagenmeyer, and H. B. Keller, "A Review of Rule Learning-Based Intrusion Detection Systems and Their Prospects in Smart Grids," *IEEE Access,* vol. 9, pp. 57542-57564, 2021.

[9] R. Ahmad, I. Alsmadi, W. Alhamdani, and L. a. Tawalbeh, "Zero-day attack detection: a systematic literature review," *Artificial Intelligence Review,* vol. 56, no. 10, pp. 10733-10811, 2023/10/01 2023.

[10] S. Varalakshmi, S. Kottur, R. Sasikumar, K. Saravanan, T. Kanungo, and B. Annamalai, "AI-Powered Anomaly Detection to Strengthen Internet of Things Security and

Forestall Cyber Attacks in Networked Device Environments," presented at the 2025 3rd International Conference on Advancement in Computation & Computer Technologies (InCACCT), Punjab, India, 17-18 April 2025, 2025.

[11] B. F. Azevedo, A. M. A. C. Rocha, and A. I. Pereira, "Hybrid approaches to optimization and machine learning methods: a systematic literature review," *Machine Learning,* vol. 113, no. 7, pp. 4055-4097, 2024.

[12] N. Moustafa and J. Slay, "The evaluation of Network Anomaly Detection Systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set," *Information Security Journal: A Global Perspective,* vol. 25, no. 1-3, pp. 18-31, 2016.

[13] S. M. Kasongo and Y. Sun, "Performance Analysis of Intrusion Detection Systems Using a Feature Selection Method on the UNSW-NB15 Dataset," *Journal of Big Data,* vol. 7, no. 1, 2020.

[14] M. Ahmad, Q. Riaz, M. Zeeshan, H. Tahir, S. A. Haider, and M. S. Khan, "Intrusion detection in internet of things using supervised machine learning based on application and transport layer features using UNSW-NB15 data-set," *EURASIP Journal on Wireless Communications and Networking,* vol. 2021, no. 10, pp. 1-23, 2021.

[15] R. A. Disha and S. Waheed, "Performance analysis of machine learning models for intrusion detection system using Gini Impurity-based Weighted Random Forest (GIWRF) feature selection technique," *Cybersecurity,* vol. 5, no. 1, 2022.

[16] M. M. Ahsan, M. A. P. Mahmud, P. K. Saha, K. D. Gupta, and Z. Siddique, "Effect of Data Scaling Methods on Machine Learning Algorithms and Model Performance," *Technologies,* vol. 9, no. 52, pp. 1-17, 2021.

[17] G. Karatas, "The Effects of Normalization and Standardization an Internet of Things Attack Detection," *European Journal of Science and Technology,* vol. 29, pp. 187-12, 201.

[18] A. Y. Hussein, P. Falcarin, and A. T. Sadiq, "Enhancement performance of random forest algorithm via one hot encoding for IoT IDS," *Periodicals of Engineering and Natural Sciences,* vol. 9, no. 3, pp. 579-571, 2021.

[19] H. Chen, H. Zhang, S. Si, Y. Li, D. Boning, and C.-J. Hsieh, "Robustness verification of tree-based models," in *The 33rd International Conference on Neural Information Processing Systems*, Vancouver, Canada, 2019, pp. 12327 - 12337: Curran Associates Inc.

[20] P. Cunningham and S. J. Delany, "k-Nearest Neighbour Classifiers - A Tutorial," *ACM Computing Surveys,* vol. 54, no. 6, pp. 1-25, 2021.

[21] R. Blanquero, E. Carrizosa, P. Ramírez-Cobo, and M. R. Sillero-Denamiel, "Variable selection for Naïve Bayes classification," *Computers and Operations Research,* vol. 135, 2021.

[22] B. S. Bhati, G. Chugh, F. Al-Turjman, and N. S. Bhati, "An improved ensemble based intrusion detection technique using XGBoost," *Trans Emerging Tel Tech.,* vol. 32, no. 6, 2020.

[23] M. B. Musthafa, S. Huda, M. A. Ali, Y. Kodera, and Y. Nogami, "Evaluation of IDS model by improving accuracy and reducing overfitting using stacking LSTM," presented at the 2024 IEEE International Conference on Consumer Electronics (ICCE), 2024.

[24] T. Kim and J.-S. Lee, "Exponential Loss Minimization for Learning Weighted Naive Bayes Classifiers," *IEEE Access,* vol. 10, pp. 22724-22736, 2022.

[25] M. Verkerken, L. D'hooge, T. Wauters, B. Volckaert, and F. De Turck, "Towards Model Generalization for Intrusion Detection: Unsupervised Machine Learning Techniques," *Journal of Network and Systems Management,* vol. 30, no. 1, p. 12, 2021/10/17 2021.

DOI: 10.34148/teknika.v14i2.1276