**IT FOR SOCIETY, Vol. 09, No. 02**
**ISSN 2503-2224**

# OPTIMIZING NETWORK INTRUSION DETECTION USING A RANDOM FOREST AND XGBOOST CLASSIFIER TO PREDICT ANOMALY ACTIVITY

Amadeuz Ezrafel
*Faculty of Computer Science*
*President University*
Bekasi, West Java
amadeuz@president.ac.id

Hasanul Fahmi
*Faculty of Computer Science*
*President University*
Bekasi, West Java
hasanul.fahmi@president.ac.id

Imron Iswanto
*Faculty of Computer Science*
*President University*
Bekasi, West Java
imron@president.ac.id

***Abstract - Intrusion Detection System (IDS) are intrinsically vital components of a computer network. Using Machine Learning (ML) algorithm to construct an IDS provides advanced defenses against increasingly complex cyberthreats. This research aims to optimize the network intrusion detection system (IDS) using Random Forest and XGBoost algorithms to make the prediction of anomalous activity and produce higher accuracy. In this research, UNSW-NB15 dataset was used to train and test the model. Data preprocessing and feature selection also was carried out before building the model. The Random Forest and XGBoost models are trained and refined individually to achieve parameters that maximize their accuracy in generating the best results. Subsequently, the Stacking technique is employed to merge the two models, with Random Forest as the underlying learner or classifier and XGBoost as the meta or Stacking learner. Model evaluation included the implementation of Cross Validation testing and distinct dataset testing. The results of this research show that combination of these models produce the accuracy performance of Cross Validation by 96.08%, compared to using both models individually, optimized Random Forest 95.99% and optimized XGBoost 95.86%..***

***Keywords— IDS, Random Forest, XGBoost, UNSW-NB15, Stacking***

## I. INTRODUCTION

Cybersecurity risks are evolving and becoming more sophisticated in this highly developed digital age. Organizations worldwide face significant risks from cyberattacks, which can lead to major financial and reputational losses. Increasingly complex cyberattacks, such as data theft, ransomware, and Distributed Denial of Service (DDoS), highlight the vulnerability of modern network systems [1]. According to Cybersecurity Ventures' 2023 report, 60% of companies that experience a cyberattack will go out of business within six months. Hackers' capabilities have evolved rapidly due to advancements in technology like the Internet and Internet-of-Things (IoT) [2]. These fraudsters constantly seek new ways to undermine network security, making Intrusion Detection Systems (IDS) crucial for identifying and responding to threats. An IDS detects suspicious or malicious activity on a network, continuously analyzing traffic for signs of attacks such as data theft and malware. IDSs are divided into network-based (NIDS) and host-based (HIDS) systems, each monitoring different aspects of network activity. They not only aid in threat detection and attack mitigation but also contribute to regulatory compliance and overall network security. Investing in IDS technology is essential for addressing current and future cybersecurity challenges, as they are vital components of a secure network [3].

Intrusion Detection Systems (IDS) are relevant to initiatives promoting the Sustainable Development Goals (SDG), particularly SDG 9: Industry and Innovation, as they help develop cyber-attack-resistant infrastructure [4]. Safe and dependable infrastructure is crucial for driving innovation and sustainable industrial development. Additionally, IDS supports SDG 16: Peace, Justice, and Strong Institutions by preventing cybercrime, which can threaten a country's digital stability and security. By providing secure digital platforms, IDS facilitates the development of transparent and efficient institutions, making research on effective security solutions like IDS essential. One effective method for creating IDS is through machine learning (ML), which does not require explicit programming and learns from experience to generate solutions [5]. Using ML to construct an IDS provides advanced defenses against increasingly complex cyberthreats by analyzing vast quantities of data and spotting unusual patterns. ML techniques like Random Forests, Decision Trees, and Neural Networks enable IDS to swiftly respond to

emerging threats and reduce false positives, a primary issue with IDS implementation [6]. Integrating ML with IDS enhances detection efficiency and accuracy, fortifying overall network security and enabling the system to be more proactive and adaptable to various cyberthreats. Thus, IDS must adapt and recognize previously unknown attack patterns [7].

Even though IDS technology has developed quickly, there are significant obstacles that need to be addressed to increase its accuracy and effectiveness. Many conventional ML-based IDS still struggle to differentiate between regular and unusual network traffic, leading to numerous false positives and negatives. As a result, real attacks may go unnoticed, or normal activities may be incorrectly flagged as threats, making IDS performance and speed critical in large data networks. Ineffective algorithms can impair network security by delaying attack detection and response times. To improve IDS accuracy beyond several previous studies, this research applies ML algorithms, namely Random Forest and XGBoost classifier. These algorithms were chosen for their high accuracy and ability to handle complex data [8]. Random Forest, an ensemble learning algorithm, consists of multiple independent decision trees built from various data subsets, which vote collectively to make a final decision, reducing overfitting and boosting prediction accuracy [9]. It effectively handles large features and unbalanced data, while XGBoost employs a gradient boosting strategy to enhance model performance, creating new models to correct previous errors for more accurate predictions. Known for its speed and computational efficiency in managing large data (Researcher, Year), this study optimizes network intrusion detection by combining these methods through stacking to better distinguish normal from anomalous activity, aiming for higher classification accuracy.

## II. RESEARCH METHODS

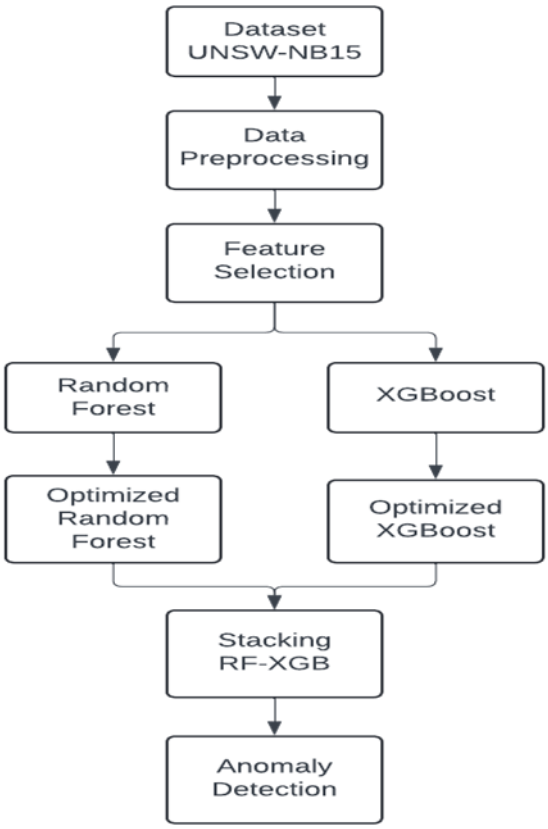



Figure 1. Research Step of NIDS Anomaly Detection

This research focuses on optimizing a Network Intrusion Detection System using Random Forest and XGBoost algorithms to accurately predict anomalous activity by collecting network data, training and combining algorithms, and evaluating and optimizing the model. The process involves stages such as data collection and environment preparation, data preprocessing including cleaning and transformation, feature selection, model training, and evaluation, which are detailed in subsequent sub-chapters and illustrated in Figure 1.

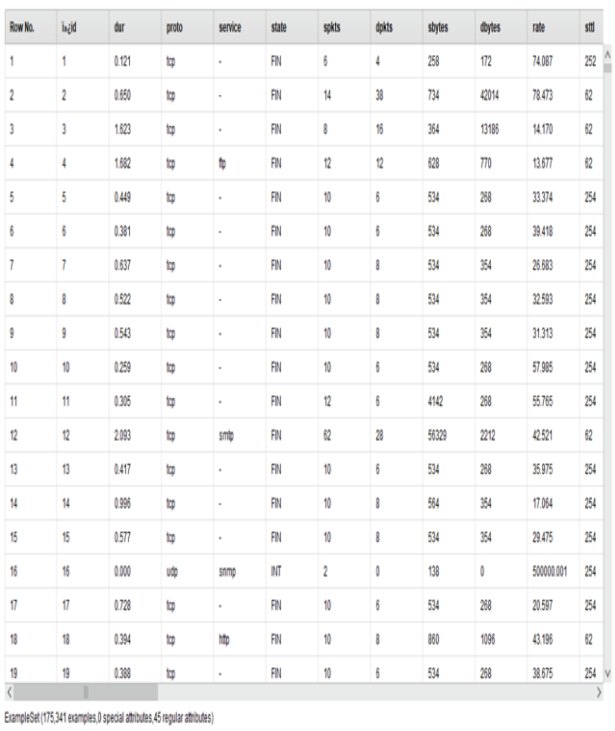
### *A. Data Collection*

Data collection for this research was obtained from open dataset sources available at the UNSW website. The dataset includes 49 separate features, detailed in the UNSW-NB15_features.csv file. It contains a total of 2,540,044 rows of raw data, divided into four CSV files: UNSW-NB15_1.csv, UNSW-NB15_2.csv, UNSW-NB15_3.csv, and UNSW-NB15_4.csv. The dataset's author partitioned the data into two files for this study: UNSW_NB15_training-set.csv with 175,341 lines for training and UNSW_NB15_testing-set.csv with

82,332 lines for testing, eliminating the need for additional data splitting.

### B. *Data Preparation*

At the data preparation stage, the data will be prepared using an application called Altair AI Studio 2024.0.1 (formerly known as RapidMiner) supported by a physical laptop device that has a Core i7-1165G7 @ 2.80GHz CPU specification with 16GB RAM. This was prepared considering the large number of datasets to be processed. At this stage, which is the first step in analyzing and understanding existing data. It involves statistics. The goal is to gain initial insight and determine the necessary preprocessing steps. To determine the quantity of attributes in the training and testing data, each dataset is first imported into the AI Studio dashboard for data pre-processing. The dataset has 45 regular attributes with a variety of data types, as seen in the image below:

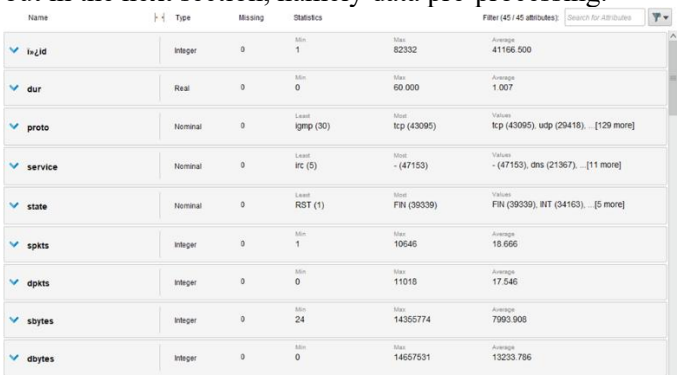


| Row No. | id | dur | proto | service | state | spkts | dpkts | sbytes | dbytes | rate | stt |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 0.121 | tcp | - | FIN | 6 | 4 | 258 | 172 | 74.087 | 252 |
| 2 | 2 | 0.650 | tcp | - | FIN | 14 | 38 | 734 | 42014 | 78.473 | 62 |
| 3 | 3 | 1.623 | tcp | - | FIN | 8 | 16 | 364 | 13186 | 14.170 | 62 |
| 4 | 4 | 1.682 | tcp | ftp | FIN | 12 | 12 | 628 | 770 | 13.677 | 62 |
| 5 | 5 | 0.449 | tcp | - | FIN | 10 | 6 | 534 | 268 | 33.374 | 254 |
| 6 | 6 | 0.381 | tcp | - | FIN | 10 | 6 | 534 | 268 | 39.416 | 254 |
| 7 | 7 | 0.637 | tcp | - | FIN | 10 | 8 | 534 | 354 | 26.683 | 254 |
| 8 | 8 | 0.522 | tcp | - | FIN | 10 | 8 | 534 | 354 | 32.593 | 254 |
| 9 | 9 | 0.543 | tcp | - | FIN | 10 | 8 | 534 | 354 | 31.313 | 254 |
| 10 | 10 | 0.259 | tcp | - | FIN | 10 | 6 | 534 | 268 | 57.985 | 254 |
| 11 | 11 | 0.305 | tcp | - | FIN | 12 | 6 | 4142 | 268 | 55.765 | 254 |
| 12 | 12 | 2.093 | tcp | smtp | FIN | 62 | 28 | 56329 | 2212 | 42.521 | 62 |
| 13 | 13 | 0.417 | tcp | - | FIN | 10 | 6 | 534 | 268 | 35.975 | 254 |
| 14 | 14 | 0.996 | tcp | - | FIN | 10 | 8 | 564 | 354 | 17.064 | 254 |
| 15 | 15 | 0.577 | tcp | - | FIN | 10 | 8 | 534 | 354 | 29.475 | 254 |
| 16 | 16 | 0.000 | udp | snmp | INT | 2 | 0 | 138 | 0 | 500000.001 | 254 |
| 17 | 17 | 0.728 | tcp | - | FIN | 10 | 6 | 534 | 268 | 20.597 | 254 |
| 18 | 18 | 0.394 | tcp | http | FIN | 10 | 8 | 860 | 1096 | 43.196 | 62 |
| 19 | 19 | 0.388 | tcp | - | FIN | 10 | 6 | 534 | 268 | 38.675 | 254 |

ExampleSet (175,341 examples, 0 special attributes, 45 regular attributes)

Figure 2. Training data UNSW-NB15

After viewing the dataset, a statistical look at the data rows is carried out to see whether there are missing values for each attribute. In Figure 3.2, it can be seen that there is data that functions as an identity such as the attributes: srcip, sport, dstip, dsport. Then attributes with category data types such as: proto, state, service, and attack_cat. These attributes will be selected to discard these features sufficiently because they do not have important data, considering that this research only discusses anomalous activity. Feature selection will be carried out in the next section, namely data pre-processing.

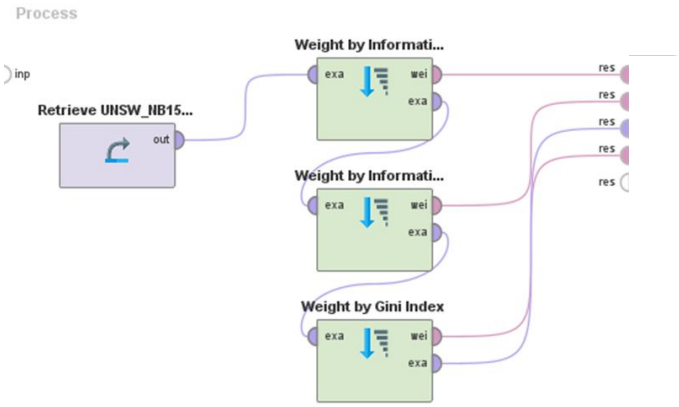


| Name | Type | Missing | Statistics | | | Filter (45 / 45 attributes): |
|---|---|---|---|---|---|---|
| id | Integer | 0 | Min: 1 | Max: 82332 | Average: 41166.500 | |
| dur | Real | 0 | Min: 0 | Max: 60.000 | Average: 1.007 | |
| proto | Nominal | 0 | Least: igmp (30) | Most: tcp (43095) | Values: tcp (43095), udp (29418), ... [129 more] | |
| service | Nominal | 0 | Least: irc (5) | Most: - (47153) | Values: - (47153), dns (21367), ... [11 more] | |
| state | Nominal | 0 | Least: RST (1) | Most: FIN (39339) | Values: FIN (39339), INT (34163), ... [5 more] | |
| spkts | Integer | 0 | Min: 1 | Max: 10646 | Average: 18.666 | |
| dpkts | Integer | 0 | Min: 0 | Max: 11018 | Average: 17.546 | |
| sbytes | Integer | 0 | Min: 24 | Max: 14355774 | Average: 7993.908 | |
| dbytes | Integer | 0 | Min: 0 | Max: 14657531 | Average: 13233.786 | |

Figure 3. UNSW-NB15 Data Attributes

### C. *Data Preprocessing*

Data preprocessing is a crucial step in preparing data for machine learning algorithms, focusing on transforming and manipulating data. It involves cleaning data by handling missing values and removing duplicates to ensure data quality. Since no missing values were found in the previous step, the preprocessing focused on eliminating duplicate data. Initial feature selection was performed using the "select attributes" operator to reduce the number of features and improve model efficiency. The label attribute, originally with values "1" and "0", was transformed into nominal data labeled as Anomaly and Normal using the "numerical to polynomial" and "Map" operators. The "set role" operator was then used to assign the label attribute as a special attribute. Further feature selection aimed to reduce computing time and overfitting, enhancing model accuracy and generalization. This research used references from previous studies that utilized 23 attributes for advanced feature selection. The "Select Attributes" operator was applied to both training and testing data for this purpose. Additionally, the feature selection results from previous research were compared with other techniques based on weighting and the original 39 features. The accuracy of these selections was tested using the Random Forest algorithm, employing a model design based on average Information Gain, Gain Ratio, and Gini Index criteria.



*D. Model Training*

The model training stage is where the algorithm learns from data divided into training sets to make predictions or decisions. In this research, Random Forest (RF) and XGBoost (XGB) were chosen as the machine learning algorithms due to their suitability for the problem type. An example design in AI Studio for modeling uses the RF algorithm, and for XGBoost, the steps are similar, with the addition of XGBoost model operators. However, since XGBoost is not available by default in AI Studio, it requires additional installation via the Marketplace updates and extensions menu. Cross-validation is conducted during the model design process for each model to provide a more precise assessment of performance compared to basic data splitting techniques like train-test split. This is because cross-validation uses the entire dataset for both training and validation alternately, reducing overfitting. The research aims to enhance prediction accuracy by combining Random Forest and XGBoost. In AI Studio, this combination is achieved by adding the "Stacking" operator, which integrates both Random Forest and XGBoost models into an ensemble.

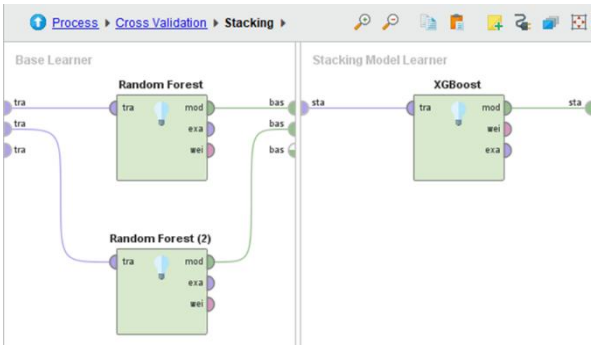



Figure 5. Stacking Model on AI Studio

*E. Model Evaluation*

The model evaluation stage occurs after training each model and the combined model, using the test set to assess performance. In AI Studio, the "Apply Model and Performance" operator calculates evaluation metrics like Accuracy, Kappa, Precision, Recall, and F1-Score. Visualization operators, such as ROC Curves, are added to visualize model performance. If test results are not optimal, parameter optimization is necessary. This research uses the Grid Optimize Parameters technique for model optimization. The process involves rebuilding the model with optimized parameters, followed by retesting and reevaluating. The Receiver Operating Characteristic (ROC) curve is then examined to assess the binary classification model's effectiveness. The "compare ROC" operator in AI Studio allows for comparing multiple models, with better models having ROC curves closer to the upper left corner.

## III. RESULTS AND DISCUSSION

*A. Experimental Result*

a. Feature Selection

The initial feature selection process involved removing identity data attributes and categorical data types such as id, state, service, proto, and attack_cat. This filtering reduced the number of attributes from 45 to 40, consisting of 39 regular attributes and one special attribute, label. This step was crucial to streamline the dataset for further analysis and model training. Following this, a more detailed feature selection was conducted by comparing the accuracy results of datasets with selected features from previous research by Husain (2019) [10] using the XGBoost algorithm and a new selection based on average attribute weighting techniques. The previous research identified 23 attributes, while the weighting method resulted in 14 attributes. This comparison aimed to determine the most effective feature set for improving model accuracy.

The datasets resulting from feature selection were then tested for accuracy using the Random Forest model with criteria such as Information Gain, Gini Index, and Gain Ratio. The results indicated that the feature selection using the XGBoost algorithm from previous research yielded higher accuracy compared to the weighting techniques and the dataset without feature selection. Specifically, the XGBoost-based selection showed superior accuracy in Information Gain and Gini Index criteria. Based on this accuracy comparison, the study decided to use the attributes identified by the XGBoost-based feature selection from previous research, which included 23 attributes. This decision was made because this feature set consistently provided the best accuracy results, demonstrating its effectiveness in enhancing the model's predictive performance.

b. Random Forest

The Random Forest model is employed as a foundational tool in this research to predict anomalous activities in network data traffic. Utilizing the UNSW-NB15 training dataset, which has undergone prior feature selection and preparation, the model was trained using AI Studio tools with cross-validation involving 10 folds and shuffled sampling. Initially, the model was configured with default parameters,

including 100 trees, Gain Ratio criteria, and a maximum tree depth of 10. The initial cross-validation results showed an accuracy of 92.91%, but this dropped to 79.13% when tested with new data, highlighting the need for parameter optimization to improve performance.

To address the decrease in accuracy, the Grid technique was employed to optimize the Random Forest parameters. Various values were tested, such as the number of trees (100, 150, 200), maximum depth (10, 15, 20), and criteria (information_gain and gini_index), resulting in 36 parameter combinations. The best parameters were selected for retraining the model, which included 150 trees, a maximum depth of 20, information_gain as the criterion, and apply_prepruning set to false. After retraining with these optimized parameters, the model's performance improved significantly, achieving a cross-validation accuracy of 95.99%, a Kappa value of 0.906, and an AUC of 0.994. When tested with new data, the model reached an accuracy of 87.05%, a notable improvement from the initial 79.13%. This optimization process demonstrated the effectiveness of parameter tuning in enhancing the model's predictive capabilities, underscoring the importance of such adjustments in improving machine learning models for intrusion detection systems.

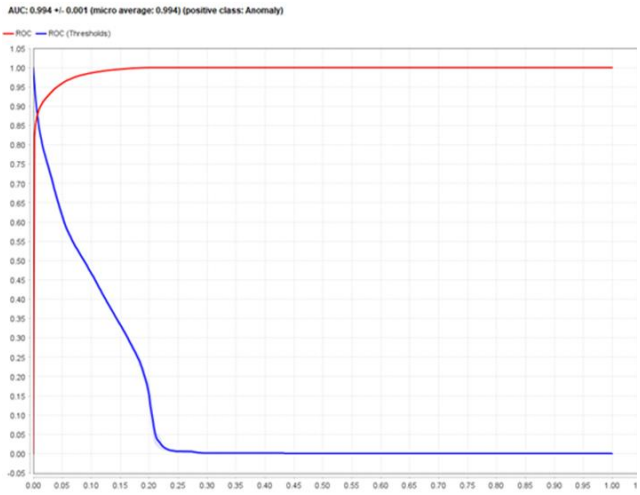



Figure 6. ROC on Cross Validation Random Forest

Based on the results of Cross Validation and the ROC curve, indicates a rise in outcomes within the Random Forest model following parameter tuning, specifically achieving an accuracy value of 95.99%. Subsequently, the trained model is once again addressed with test data, producing the following outcomes.

accuracy: 87.05%

| | true Normal | true Anomaly | class precision |
|---|---|---|---|
| pred. Normal | 26895 | 557 | 97.97% |
| pred. Anomaly | 10105 | 44775 | 81.59% |
| class recall | 72.69% | 98.77% | |

```
accuracy: 87.05%
kappa: 0.732
AUC: 0.983 (positive class: Anomaly)
f_measure: 89.36% (positive class: Anomaly)
```

Figure 7. Results from Testing Data on Optimized Random Forest

Based on the results of the image above, it shows that when the Random Forest model has optimized its parameters, there is an increase from the previous 79.13% to 87.05%. Likewise with the results of the Kappa, AUC and F_measure values.

c. XGBoost Classifier

The XGBoost classifier is utilized as an independent model in this research, which later transitions into a meta learner when combined with Random Forest. The dataset used is the UNSW-NB15 training dataset, which has been preprocessed and undergone feature selection. Training the XGBoost model required additional extensions in AI Studio tools, as the default setup does not include this algorithm. Cross-validation was performed with 10 folds and random sampling, using default parameters such as rounds=25, learning_rate=0.3, min_split_loss=0, max_depth=6, lambda=1, and alpha=0. The initial results showed an accuracy of 95.15% from cross-validation, which decreased to 86.76% when tested with new data, indicating a need for parameter optimization to enhance the model's performance.

To address this, the Grid technique was employed to optimize the XGBoost parameters. Various values were tested, including rounds (150, 175, 200), learning_rate (0.3, 0.4, 0.5, 0.6), min_split_loss (0, 2), max_depth (5, 6), and lambda (0, 1), generating 96 parameter combinations. The best set was selected for retraining the model, which included rounds=150, learning_rate=0.3, min_split_loss=0, max_depth=5, and lambda=1. After retraining with these optimized parameters, the XGBoost model showed improved performance, achieving a cross-validation accuracy of 95.86%, a Kappa value of 0.904, and an AUC of 0.994. When tested with new data, the model reached an accuracy of 87.19%, a slight improvement from the initial 86.76%. This optimization process demonstrated the effectiveness of parameter tuning in enhancing the model's predictive capabilities, underscoring

the importance of such adjustments in improving machine learning models for intrusion detection systems.

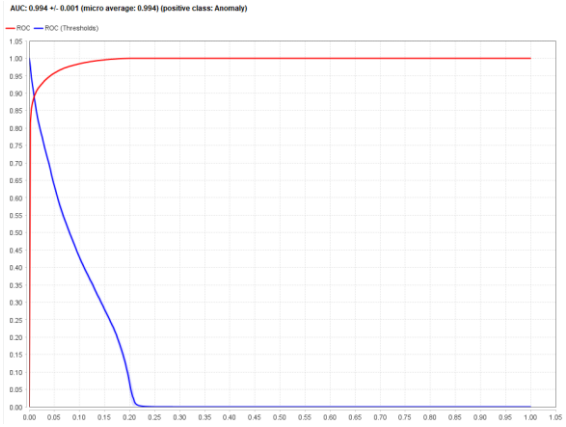



Figure 8. ROC on Cross Validation XGBoost

Based on the results of Cross Validation and the ROC curve, it shows that there is an increase in results in the XGBoost model after parameter optimization, achieving an accuracy value of 95.86%. Next, the model that has been trained is again faced with test data and the results are as follows.

accuracy: 87.19%

| | true Normal | true Anomaly | class precision |
|---|---|---|---|
| pred. Normal | 27310 | 856 | 96.96% |
| pred. Anomaly | 9690 | 44476 | 82.11% |
| class recall | 73.81% | 98.11% | |

```
accuracy: 87.19%
Kappa: 0.737
AUC: 0.982 (positive class: Anomaly)
f_measure: 89.43% (positive class: Anomaly)
```

Figure 9. Results from Testing Data on Optimized XGBoost

The image above results demonstrate that there is an increase from 86.76% to 87.19% after the XGBoost model's parameters have been improved. In the same way, the f measure, AUC, and Kappa values.

d. Combination of Random Forest and XGBoost

The research explores the integration of Random Forest and XGBoost using the Stacking method, a technique in ensemble learning that involves using multiple models to improve prediction accuracy. In this setup, Random Forest models serve as base learners, while XGBoost acts as the meta learner. The Stacking method allows for the combination of the strengths of both algorithms, aiming to enhance the overall performance of the intrusion detection system. This approach leverages the diverse capabilities of each model, with Random Forest providing robust decision-making through its ensemble of trees and XGBoost offering precise adjustments through gradient boosting.

The configuration of the base and meta learners involves two Random Forest models, RF1 and RF2, optimized for parameter settings. RF1 is configured with parameters such as 150 trees, a maximum depth of 20, and information gain as the criterion, while RF2 uses 100 trees with similar depth and criterion settings. These configurations were chosen based on their high accuracy in previous tests. XGBoost, as the meta learner, uses optimized parameters including 150 rounds, a learning rate of 0.3, and a maximum depth of 5, which were determined to be effective in previous optimization steps. The combination of Random Forest and XGBoost through Stacking yielded an accuracy of 96.08% in cross-validation, with a Kappa score of 0.909 and an F-measure of 97.15%. When tested with new data, the model achieved an accuracy of 87.13%, demonstrating robust performance. This combination approach shows promise for enhancing intrusion detection systems by leveraging the complementary capabilities of both algorithms, resulting in improved accuracy and stability in predictions.

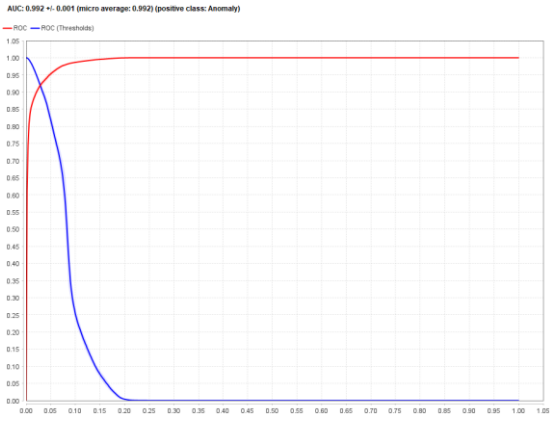



Figure 10. ROC-AUC on Stacking of Random Forest and XGBoost

According to the findings of Cross Validation and the ROC curve, merging the Random Forest and XGBoost models using the Stacking technique yields an accuracy of 96.08%. The curve, which approaches the top left corner, indicates successful combination of the models. Subsequently, the trained model is once again presented with test data, producing the following outcomes.

accuracy: 87.13%

| | true Normal | true Anomaly | class precision |
|---|---|---|---|
| pred. Normal | 26948 | 544 | 98.02% |
| pred. Anomaly | 10052 | 44788 | 81.67% |
| class recall | 72.83% | 98.80% | |

```
accuracy: 87.13%
kappa: 0.734
AUC: 0.976 (positive class: Anomaly)
f_measure: 89.42% (positive class: Anomaly)
```

Figure 11. Results from Testing Data on Stacking of Random Forest and XGBoost

The results indicate that the combining of Random Forest and XGBoost using the Stacking technique, with a distinct test dataset, achieves an accuracy of 87.13%. The comparison of accuracy results between the test data and cross validation results indicates a slight drop, but it is not statistically significant. Therefore, it can be concluded that the model does not exhibit excessive overfitting.

## B. *Result Evaluation*

Table 1. Evaluation Score

| Optimized Model | Accuracy | Precision | Recall | F-Score | AUC | Kappa |
|---|---|---|---|---|---|---|
| Random Forest | 95.99% | 95.99% | 98.20% | 97.08% | 0.994 | 0.906 |
| XGBoost | 95.86% | 96.30% | 97.67% | 96.98% | 0.994 | 0.904 |
| Random Forest + XGBoost | 96.08% | 96.18% | 98.13% | 97.15% | 0.992 | 0.909 |

The assessment score table indicates that the combination of Random Forest and XGBoost models achieves the highest accuracy value of 96.08%, the highest F-Score value of 97.15%, and the highest Kappa score of 0.909. The F-Score is particularly important as it measures the balance between precision and recall, making it valuable for optimizing the detection rate of intrusions while minimizing false alarms. A F-Score of 97.15% demonstrates the system's effectiveness in accurately identifying intrusions without being overly influenced by false positives. Therefore, the high F-Score achieved by the Stacking method of Random Forest and XGBoost serves as a strong foundation for overall evaluation. Additionally, accuracy metrics offer a general perspective on how effectively the IDS distinguishes between normal and anomalous activities. With the highest accuracy score of 96.08%, it acts as a quick indicator of the model's proper functioning. Thus, both F-Score and accuracy metrics are crucial in this research.

The optimized Random Forest model achieved a Recall value of 98.20% and an AUC value of 0.994, which is the highest among all models. In comparison, the optimized XGBoost model reached a Precision value of 96.30% and an AUC value of 0.994, identical to the Random Forest model. These metrics highlight the strengths of each model in different areas of evaluation. Overall, the three models—Random Forest, XGBoost, and their combination—demonstrate excellent performance, achieving high marks across all categories. This highlights their effectiveness and reliability in intrusion detection, underscoring the success of the optimization and combination strategies employed in this research.

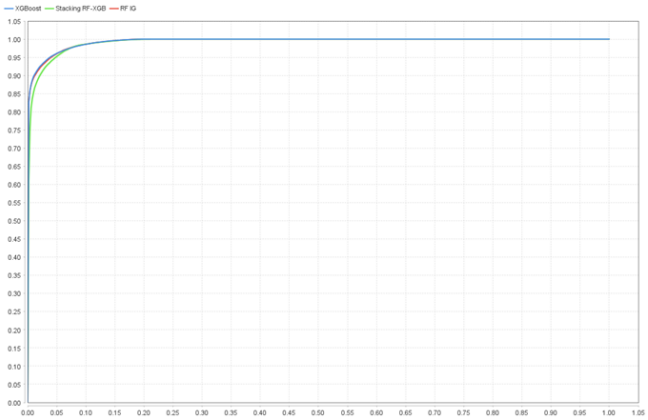



Figure 12. Comparison of ROC

Demonstrate that all three models exhibit exceptional performance in their classification tasks, with nearly identical proficiency in differentiating between positive and negative classes. After optimizing the parameters, both the Random Forest and XGBoost models achieved the same ROC AUC value of 0.994, indicating comparable performance in terms of their categorization capacity. The Random Forest model effectively mitigates overfitting by constructing each tree using a distinct subset of data and making decisions based on the aggregated average or voting outcomes of all trees. Meanwhile, XGBoost constructs models iteratively, with each subsequent model aiming to rectify errors made by the preceding model, allowing it to manage imbalanced data effectively and assign greater importance to challenging data in subsequent iterations.

However, when the Random Forest and XGBoost models are combined using the Stacking approach, the ROC-AUC value slightly decreases to 0.992. Despite this minimal difference of 0.002, the combination still exhibits excellent performance and is nearly flawless in its classification tasks, making it a robust model. This robustness is derived from Random Forest's effectiveness in mitigating overfitting and XGBoost's iterative model-building process. Since this research primarily emphasizes accuracy metrics, the model will undergo further optimization as an Intrusion Detection System (IDS) to predict anomalous activities. The accuracy results obtained from the cross-validation model training procedure will be compared with those from data testing,

providing a comprehensive evaluation of the model's predictive capabilities.

Table 2. Comparison of Accuracy results

| Optimized Model | Accuracy | |
|---|---|---|
| | Cross Validation | Testing |
| **Random Forest** | 95.99% | 87.05% |
| **XGBoost** | 95.86% | 87.19% |
| **Random Forest+XGBoost** | 96.08% | 87.13% |

The combination of Random Forest and XGBoost models in cross-validation demonstrates improved performance, achieving the maximum accuracy value compared to the separate Random Forest and XGBoost models. This suggests that the integrated model possesses superior capability in capturing patterns from the training data and maintaining consistency across various data subsets during validation. These results indicate that the combination of Random Forest and XGBoost models, known as Stacking, shows promising potential and demonstrates more stability in terms of generalization performance. However, when all three models were tested using completely separate, unseen test data, the results showed that the combination of models had lower accuracy than the individual XGBoost model but still achieved better results than the individual Random Forest model. This suggests that although the combination performs very well in a cross-validation environment, it may suffer from overfitting to the training data and struggle to maintain performance with new data. Nevertheless, this is not a definitive benchmark, as there remains the possibility that the combination could produce higher accuracy, especially considering the cross-validation results, which aim to provide a more consistent and reliable evaluation and assist in standardizing model performance.

Table 3. Classifier Model Comparison Results on UNSW-NB15

| Classifier Model | Accuracy |
|---|---|
| Random Forest (Khan, 2019)[11] | 83.63% |
| ANN with FS-XGB (Kasongo, 2020)[12] | 84.39% |
| SMO-SVM (Khan 2019)[11] | 85.34% |
| GA based RF (Assiri, 2021)[13] | 86.70% |
| Stacking RF-XGB | 87.13% |

Based on the comparison table above, it is evident that the same test dataset, UNSW-NB15, was used in several studies. The combination of Random Forest and XGBoost algorithms shows an increase in accuracy compared to other classification models. For instance, in research conducted by Assiri [15], which utilized a Genetic Algorithm-based Random Forest, an accuracy of 86.70% was achieved, whereas the combination of Random Forest and XGBoost was 0.43% superior. This improvement suggests that using Random Forest and XGBoost optimization via the Stacking method can enhance classification performance in network intrusion detection systems (IDS) []14.

This improvement arises because both models possess complementary capabilities in managing complex data structures. Random Forest mitigates overfitting by using multiple decision trees and aggregating their results through majority voting, enhancing the model's stability and generalization. XGBoost, on the other hand, is a highly efficient implementation of gradient boosting that maximizes accuracy through iterative optimization and regularization techniques to prevent overfitting. By integrating Random Forest and XGBoost, models can leverage the stability and resilience of Random Forest along with the optimization capabilities and precision of XGBoost, achieving superior performance across various datasets. In the context of Network IDS, these results can be further enhanced by integrating with network monitoring tools like Wireshark and Snort for real-time traffic analysis. Additionally, the findings can be incorporated into smart routers or IoT devices or used as a live data processing pipeline with technologies like Apache Kafka or Apache Flink. Implementing an alert mechanism to notify administrators of anomalies detected by the stacking model can also be practical, using methods such as email, SMS, or integration with a security information and event management (SIEM) system.

## IV. CONCLUSION AND RECOMMENDATION

In this research, the selection of the best features and the implementation of the Random Forest and XGBoost algorithm models were successfully optimized for several parameters to enhance accuracy in predicting anomalous activity on intrusion detection systems (IDS). The combination of these two algorithms using the Stacking method was effectively executed, with Random Forest serving as the base learner and XGBoost as the meta or Stacking learner. This integration improved cross-validation accuracy to 96.08%, surpassing the performance of the models individually, with optimized Random Forest at 95.99% and optimized XGBoost at 95.86%. The Stacking method also achieved the highest F-Score of 97.15%, providing a strong foundation for overall evaluation. When tested with new data, this combination delivered an accuracy performance of 87.13%, which, although slightly lower than the individual optimized XGBoost model, still outperformed the individual optimized Random Forest model. This outcome suggests that while the combination may yield higher values with different test data, cross-validation results offer a more consistent and reliable evaluation, aiding in standardizing model performance. Overall, it can be concluded that models with optimized parameters

generally perform better than those using default settings. By integrating Random Forest and XGBoost, models can exploit the stability and robustness of Random Forest along with the optimization capabilities and accuracy of XGBoost, achieving superior performance across diverse datasets and conditions.

This research successfully optimized the Random Forest and XGBoost algorithm models to enhance accuracy in predicting anomalous activity on intrusion detection systems (IDS). Using the Stacking method, Random Forest served as the base learner and XGBoost as the meta learner, leading to an improved cross-validation accuracy of 96.08%. This surpassed the individual performances of optimized Random Forest at 95.99% and XGBoost at 95.86%, and achieved the highest F-Score of 97.15%. Although the combined model showed slightly lower accuracy on new data at 87.13%, it still outperformed the individual Random Forest model, highlighting the benefits of parameter optimization and model integration.

## REFERENCES


[1] Y. Li and Q. Liu, "A comprehensive review study of cyber-attacks and cyber security; Emerging trends and recent developments," Energy Reports, vol. 7, pp. 8176–8186, Nov. 2021, doi: 10.1016/j.egyr.2021.08.126.

[2] C. Osborne and Cybersecurity Ventures, "SECURITY AWARENESS TRAINING," 2023. Accessed: Jun. 17, 2024. [Online]. Available: https://www.esentire.com/resources/library/2023-official-cybercrime-report

[3] H. Alqahtani, I. H. Sarker, A. Kalim, S. M. Minhaz Hossain, S. Ikhlaq, and S. Hossain, "Cyber intrusion detection using machine learning classification techniques," in Communications in Computer and Information Science, Springer, 2020, pp. 121–131. doi: 10.1007/978-981-15-6648-6_10.

[4] D. of E. and S. A. United Nations, "The Sustainable Development Goals Report," 2022. Accessed: Jun. 17, 2024. [Online]. Available: https://unstats.un.org/sdgs/report/2022/The-Sustainable-Development-Goals-Report-2022.pdf

[5] L. Setiyani et al., "Defending Your Mobile Fortress: An In-Depth Look at on-Device Trojan Detection in Machine Learning: Systematic Literature Review," Jurnal Penelitian Pendidikan IPA, vol. 9, no. 7, pp. 302–308, Jul. 2023, doi: 10.29303/jppipa.v9i7.4209.

[6] G. Kocher and G. Kumar, "Analysis of Machine Learning Algorithms with Feature Selection for Intrusion Detection using UNSW-NB15 Dataset," International Journal of Network Security & Its Applications, vol. 13, no. 1, pp. 21–31, Jan. 2021, doi: 10.5121/ijnsa.2021.13102.

[7] K. Shaukat, S. Luo, S. Chen, and D. Liu, "Cyber Threat Detection Using Machine Learning Techniques: A Performance Evaluation Perspective," in 1st Annual International Conference on Cyber Warfare and Security, ICCWS 2020 - Proceedings, Institute of Electrical and Electronics Engineers Inc., Oct. 2020. doi: 10.1109/ICCWS48432.2020.9292388.

[8] A. Parmar, R. Katariya, and V. Patel, "A Review on Random Forest: An Ensemble Classifier," in Lecture Notes on Data Engineering and Communications Technologies, vol. 26, Springer Science and Business Media Deutschland GmbH, 2019, pp. 758–763. doi: 10.1007/978-3-030-03146-6_86.

[9] Z. Chen, F. Jiang, Y. Cheng, X. Gu, W. Liu, and J. Peng, "XGBoost Classifier for DDoS Attack Detection and Analysis in SDN-Based Cloud," in Proceedings - 2018 IEEE International Conference on Big Data and Smart Computing, BigComp 2018, Institute of Electrical and Electronics Engineers Inc., May 2018, pp. 251–256. doi: 10.1109/BigComp.2018.00044.

[10] Anwar Husain, Ahmed Salem, Carol Jim, and George Dimitoglou, "Development of an Efficient Network Intrusion Detection Model Using Extreme Gradient Boosting (XGBoost) on the UNSW-NB15 Dataset," 2019.

[11] F. A. Khan and A. Gumaei, "A Comparative Study of Machine Learning Classifiers for Network Intrusion Detection," in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Springer Verlag, 2019, pp. 75–86. doi: 10.1007/978-3-030-24265-7_7.

[12] J. O. Mebawondu, O. D. Alowolodu, J. O. Mebawondu, and A. O. Adetunmbi, "Network intrusion detection system using supervised learning paradigm," Sci Afr, vol. 9, Sep. 2020, doi: 10.1016/j.sciaf.2020.e00497.

[13] N. K. Sinha, M. Khulal, M. Gurung, and A. Lal, "Developing A Web based System for Breast Cancer Prediction using XGboost Classifier." [Online]. Available: www.ijert.org

[14] S. M. Kasongo and Y. Sun, "Performance Analysis of Intrusion Detection Systems Using a Feature Selection Method on the UNSW-NB15 Dataset," J Big Data, vol. 7, no. 1, Dec. 2020, doi: 10.1186/s40537-020-00379-6.

[15] A. Assiri, "Anomaly classification using genetic algorithm-based random forest model for network attack detection," Computers, Materials and Continua, vol. 66, no. 1, pp. 767–778, 2021, doi: 10.32604/cmc.2020.013813.