

# Comprehensive Analysis of MLP and Ensemble Learning Approaches for Intrusion Detection using CICIDS2017 Dataset

Md Salim Raza, Humaira Arif, Sai Mounika Errapotu, Virgilio Gonzalez

*Department of Electrical and Computer Engineering, The University of Texas at El Paso, El Paso, USA*

Email: mraza@miners.utep.edu, harif@miners.utep.edu, serrapotu@utep.edu, vgonzalez3@utep.edu

**Abstract**—This paper presents a comprehensive evaluation of the Multi-Layer Perceptron (MLP) and ensemble learning classifiers, including Random Forest, GBM (Gradient Boosting Machine) and XGBoost, for intrusion detection using the CICIDS2017 dataset. The study analyzes the workflow towards improving accuracy in anomaly based intrusion detection, and assesses the performance of these models across various metrics such as accuracy, precision, recall, and F1 score, with a detailed consideration of receiver operating characteristic (ROC) and learning curves. The training process includes data balancing using Synthetic Minority Over-sampling Technique (SMOTE), feature selection, and hyperparameter tuning through grid search. The results reveal that Random Forest outperforms other classifiers, achieving the highest accuracy, area under the curve (AUC), demonstrating strong learning curve convergence even with smaller data samples. Interestingly, when retrained on reduced data samples, Random Forest maintained its superior accuracy without signs of overfitting, outperforming the other classifiers. These findings highlight Random Forest's effectiveness in distinguishing between attack and benign traffic, making it the preferred choice for intrusion detection in this study.

**Index Terms**—Intrusion Detection System (IDS), Machine Learning, CICIDS2017, Multi-Layer Perceptron (MLP), Random Forest, XGBoost, SMOTE, Feature Selection, Hyperparameter Tuning, ROC Curve, Learning Curve, Performance Metrics.

## I. INTRODUCTION

### A. Background

The Internet, knowledgeable users, voice and video over IP, distributed data storage systems, encryption and authentication methods, remote and wireless access, and web services are just a few of the components that constitute the global ecosystem that has developed into cyberspace to enable the exchange of electronic resources [1]. Since 2017, developed countries have experienced an 81% surge in Internet usage with expansion in cyberspace, significantly impacting the data transfers and information exchange. However, this growth has also led to an increase in cyber threats such as phishing, malware, and network intrusions. Cybersecurity, an area dedicated to safeguarding cyberspace, has evolved from traditional methods to advanced automated systems over the years. These automated cybersecurity methods play a critical role in identifying and combating increasingly sophisticated polymorphic cyber-attacks, thus keeping up with the progressively complex nature of cybercrimes.

Spam emails, viruses, and network intrusions are among the most common cyber hazards [2]. Ransomware and viruses fall under the category of malware, that target through spam

emails, disrupt operations and corrupt electronic data [3]. Intrusion Detection Systems (IDS) are necessary to protect against network intrusions that take advantage of system vulnerabilities. Intrusion detection system (IDS) is an a vital security tool in the security chain, which critically monitors system or network activity for any malicious activity or policy violations. These systems provide a broad range of security coverage and can be installed on a single computer or over large networks.

The two most common detection techniques used by IDS technologies are signature-based and anomaly-based detection. IDS technologies can also be differentiated depending on these methods. Signature-based detection is a technique that identifies new and undiscovered threats less successfully than anomaly-based detection. It works by comparing observed network traffic or system activity with a database of known attack patterns or signatures. However, anomaly-based detection is especially helpful for detecting new or emerging threats without an existing signature since it establishes a norm for acceptable activity and recognizes departures from this norm as potential threats [4].

Machine learning plays a crucial role in anomaly-based detection by enabling the IDS to continuously learn and update its understanding of what constitutes normal behavior. The integration of machine learning algorithms into anomaly-based intrusion detection systems (IDS) can enhance their precision, diminish the probability of false positives, and offer a more intelligent and adaptable means of real-time threat detection [5], [6].

In this paper, we will evaluate an MLP-based IDS, a type of Artificial Neural Network (ANN), along with other machine learning models, focusing on IDS implementations using ensemble learning techniques. The CICIDS2017 dataset is chosen for its comprehensive, up-to-date representation of real-world network traffic, making it ideal for evaluating IDS. Unlike older datasets like NSL-KDD and KDD99, CICIDS2017 captures modern attack scenarios, including DoS, DDoS, brute force, and infiltration, proving to be great choice for realistic evaluation of IDS models [7], [8].

### B. Existing Research

Machine learning algorithms are highly effective for developing Intrusion Detection Systems (IDS) by detecting attack patterns in extensive network traffic datasets [9]. While various models such as Decision Trees, ensemble learning, Support

Vector Machines (SVM), and Neural Networks have been explored in intrusion detection, there is still a need for comprehensive evaluations of these models, especially on datasets like CICIDS2017, which encompass a wide range of real-world attack types.

An intrusion detection system was presented by Tama et al. [10], which used ensemble techniques—like Random Forest, Gradient Boosting Machine, and XGBoost—instead of conventional classifiers to identify web attacks. Their method yielded a top accuracy of 99.98% on the CICIDS-2017 dataset, demonstrating exceptional performance.

Nzuvu et al. [11] trained a variety of algorithms, such as ensemble models and artificial neural networks, using the CICIDS2017 dataset in order to address the problem of real-time intrusion detection failures. With C4.5 serving as the basis classifier, AdaBoost proved to be the most effective combination, with an accuracy of 99.06% after additional refining.

Gosai et al. [12] looked into the application of ensemble methods to machine learning model classification of intrusions. They assessed performance measures in their trials using the KDD-Cup'99 and CICIDS-2017 datasets, and on the latter they achieved an accuracy of 99.87% on CICIDS-2017 dataset.

Using the CICIDS-2017 dataset, Li et al. [13] concentrated on improving intrusion detection systems via anomaly detection. They employed an ensemble strategy using bagging and XGBoost in addition to SMOTE for class balancing, and the outcome was remarkable macro metrics: precision at 93.2%, F1-Score at 95.5%, recall at 98%, and a ROC Curve at 99.4%.

Using machine learning and ensemble approaches, Thaker et al. [14] suggested an intelligent framework for intrusion detection in autonomous cars. Using the CICIDS-2017 dataset and attack traffic recorded using Wireshark from a Controller Area Network (CAN), their analysis discovered that the XGBoost classifier performed the best, with an accuracy of 98.57%.

An intrusion detection system was created by Mhawi et al. [15] by combining an ensemble approach with the hybrid feature selection technique known as CFS-FPA. Using AdaBoosting and bagging approaches, they improved four conventional classifiers. These improvements addressed issues including high dimensionality and the occurrence of false positives and false negatives. Their algorithm demonstrated an astounding 99.7% accuracy when tested on the CICIDS-2017 dataset.

### C. Limitations of Existing Research

- Insufficient AUC Analysis:** Many studies fail to thoroughly evaluate the AUC, a critical metric for assessing model performance in imbalanced datasets.
- Lack of Decision Threshold Tuning:** Research commonly uses default decision thresholds without optimizing them, potentially compromising the balance between precision and recall.
- Limited Focus on Learning Curves:** The progression and effectiveness of model learning over time or with varying data sizes are not adequately analyzed.

- Overlook of MLP (ANN) Models:** Existing research often neglects the potential of Multi-Layer Perceptron models, missing out on valuable insights from ANN-based approaches.

### D. Our Contributions

In this paper, we address the gaps identified in existing research by providing a comprehensive evaluation of Multi-Layer Perceptron (MLP) and ensemble learning classifiers, including Random Forest, GBM, and XGBoost, using the CICIDS2017 dataset. Our contributions include the following:

- Holistic Model Evaluation:** We assess model performance across various metrics, such as accuracy, precision, recall, F1 score, and AUC, with a detailed analysis of ROC and learning curves. This comprehensive evaluation ensures a more balanced understanding of each model's strengths and weaknesses.
- Data Imbalance Handling:** Unlike some existing studies, we incorporate data balancing through SMOTE to address class imbalance, thereby improving the detection of rare attacks and reducing the likelihood of model bias.
- Scalability and Efficiency:** We examine the scalability of the evaluated models, particularly their performance with smaller data samples. Our findings indicate that Random Forest maintains high accuracy, even with reduced data selected from the ROC curve, without suffering from overfitting.
- Extensive MLP Investigation:** We will extensively investigate the MLP classifier, employing grid search and hyperparameter tuning to evaluate its overall performance on the CICIDS2017 dataset, ensuring optimal configuration and comparison with other models.

These contributions collectively advance the state of intrusion detection research, offering practical insights for deploying effective IDS models in various real-world scenarios.

### E. Organization of the Paper

The remainder of this paper is organized as follows: Section II describes the proposed methodology, including data preprocessing steps, model training, and validation procedure and finally presents the performance analysis of the selected best models with optimized decision thresholds, while Section III discusses the key findings and conclusions drawn from the study. Finally, Section IV acknowledges contributions and references relevant literature.

## II. PROPOSED METHODOLOGY

The proposed method, as depicted in Figure 1, outlines a sequential workflow that starts with raw data and progresses towards an optimized decision threshold. The effectiveness of this process is evaluated using performance metrics including precision, recall, accuracy, and F1 score. Each step involved in the structure of the proposed IDS is detailed below.

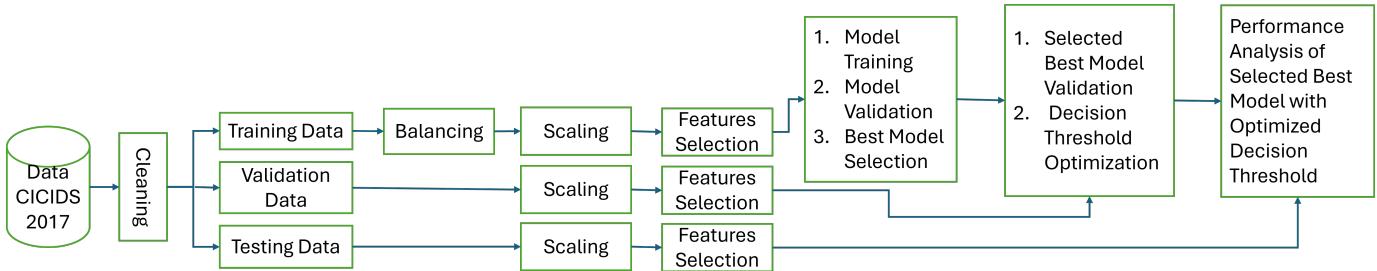


Fig. 1. Proposed IDS Structure

### A. CICIDS2017 Dataset

The CICIDS2017 dataset is used as the primary data source, providing network traffic data essential for training and evaluating an intrusion detection system. This dataset includes various types of attacks alongside benign traffic, offering a comprehensive set for model training and testing. Figure 2 illustrates the distribution of the dataset. Our primary goal is to classify the traffic as either attack or benign, consolidating all attack types into a single category called "attack," while other category called "benign" traffic is illustrated in Figure 3.

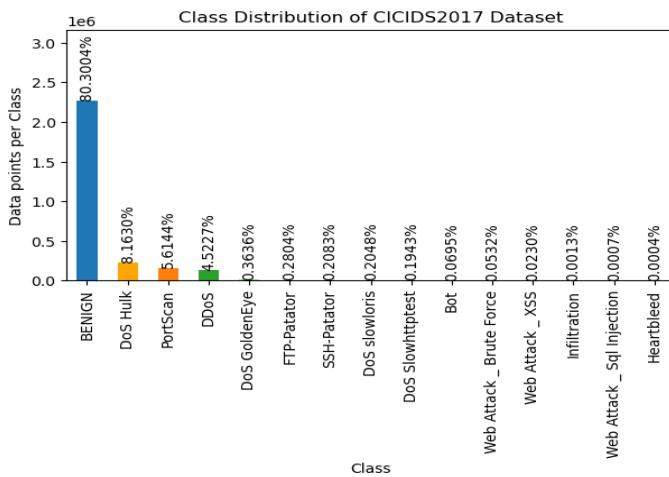


Fig. 2. CICIDS2017 Dataset with several types of classes.

### B. Cleaning

Identify and handle missing values in the dataset by detecting any missing data. Remove columns that contain NaN values entirely. Replace any infinity values with the mean of the respective column to ensure data quality and consistency.

### C. Splitting Dataset

The dataset is divided into training, validation, and testing subsets. The training data is used to train the model. The validation data is used to tune the model's hyperparameters and to optimize the decision threshold. The testing data is used to evaluate the final performance of the model.

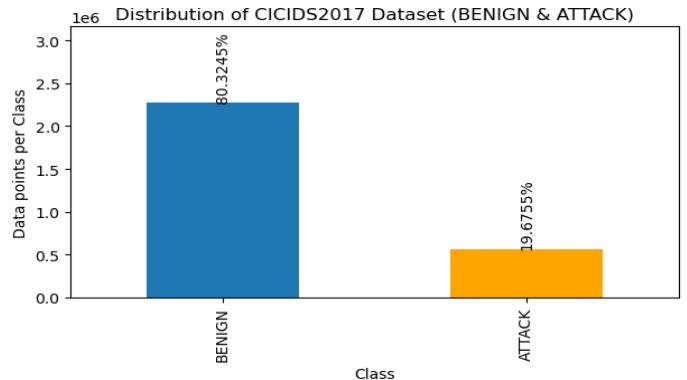


Fig. 3. CICIDS 2017 Dataset with two classes (benign and attacks).

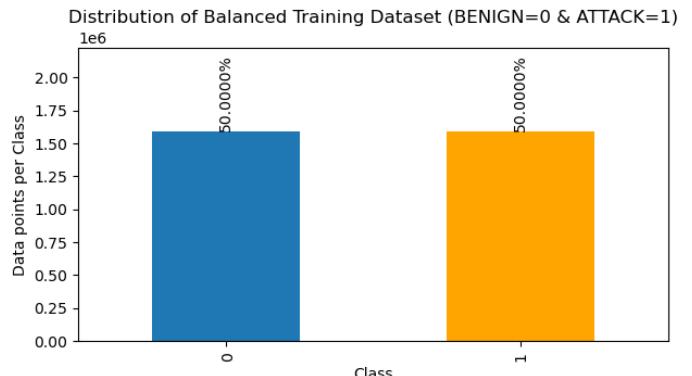


Fig. 4. Distribution of training dataset after apply SMOTE.

### D. Dataset Balancing

To address the class imbalance in the data set, data balancing techniques such as undersampling and oversampling can be employed. In this case, we apply the Synthetic Minority Over-sampling Technique (SMOTE) to the training data only because the model can be trained to better recognize the minority class without compromising the validity of the performance evaluation on the test set. SMOTE generates synthetic samples for the minority class (attacks) to balance the dataset. It achieves this by selecting samples that are close in the feature space, drawing lines between these samples, and then generating new samples along these lines. This approach helps

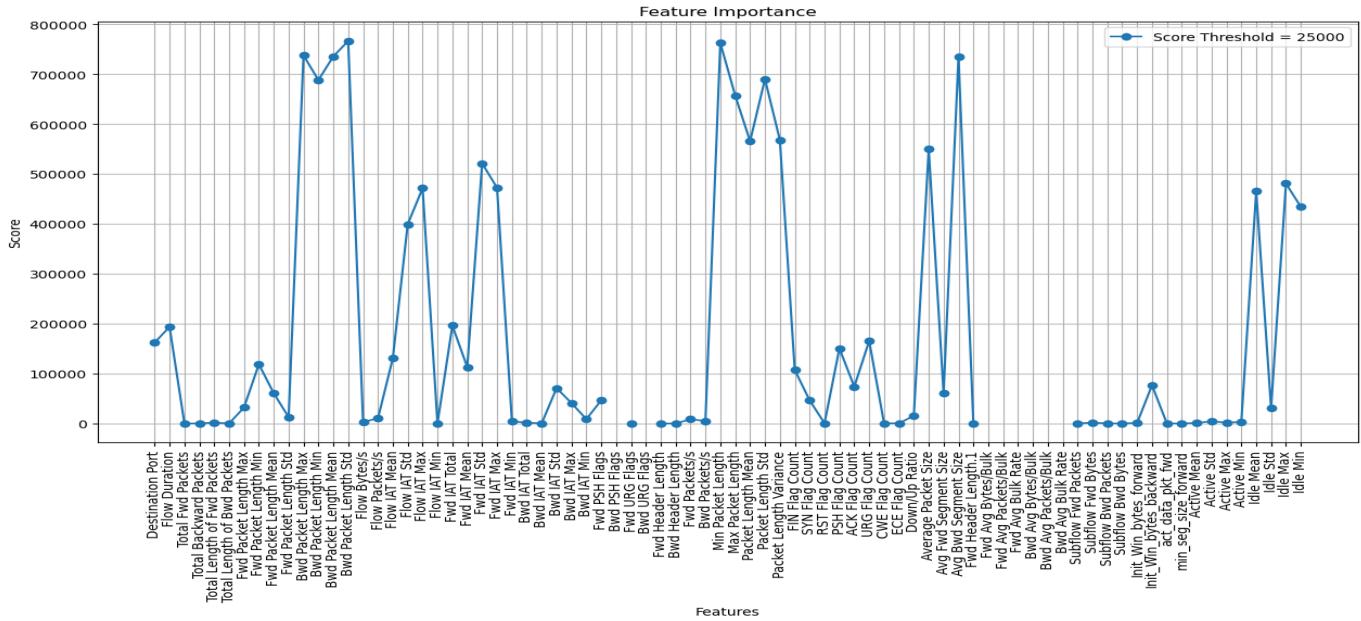


Fig. 5. Feature importance score curve of the CICIDS2017 dataset using F\_classif.

prevent overfitting and enhances the classifier's ability to generalize from the training data to unseen data. The balanced dataset is illustrated in Figure 4.

#### E. Scaling

Apply a standard scaler to the training, validation, and testing data. Scaling ensures that all features have the same scale, which can improve the performance of the machine learning model. The scaler is fitted on the training data after balancing and then applied to the validation and testing data.

#### F. Feature Selection

Feature selection plays a vital role in improving model performance and reducing overfitting by identifying the most relevant features. In our approach, we apply feature selection to the scaled training, validation, and testing data using the F\_classif method, which ranks features based on their F-statistic scores. Features with scores above 2500 are selected, resulting in the selection of 37 features out of the original 78, as shown in Figure 5. This threshold can be adjusted to meet specific requirements. By concentrating on the most informative features, this method enhances the model's predictive accuracy, leading to better generalization and increased efficiency.

#### G. Model Training, Validation, and Best Model Selection

We utilized MLP and ensemble learning classifiers, including Random Forest, GBM, and XGBoost, for model training. The training dataset was split into 5 folds, with 4 folds used for training and the remaining fold for validation, repeating this process five times. To fine-tune the hyperparameters and select the optimal model, grid search was employed. It's worth noting that grid search was not applied if a model reached over 99% validation accuracy with its default settings.

#### H. Selected Best Model Validation and Decision Threshold Optimization

After selecting the best-performing model from the training phase, the final step involves validating this model using an unseen validation dataset. This step ensures that the model generalizes well to new, untrained data. Additionally, we optimize the decision threshold for the selected model to achieve the best possible balance between precision and recall, depending on the specific requirements of the task.

The decision threshold determines the point at which the model classifies a prediction as positive or negative, and optimizing this threshold can significantly impact the model's performance. In the subsequent section, titled "Performance Analysis of the Selected Best Model with Optimized Decision Threshold," we present detailed curves that illustrate the ROC curve with decision thresholds on each classifier's performance.

#### I. Performance Analysis of the Selected Best Model with Optimized Decision Threshold

Utilizing the testing dataset, the last stage entails assessing the effectiveness of the best model that was chosen together with an optimal decision threshold. We evaluate multiple critical parameters in order to fully evaluate each classifier's performance. Examining learning curves, evaluating the confusion matrix, computing critical performance metrics like accuracy, precision, recall, and F1 score, and assessing the ROC curve with the optimal decision threshold are all samples of this.

Figures 6 to 9 clearly show that XGBoost achieves the highest ROC area, with a value of 0.99990, slightly surpassing Random Forest and outperforming the other classifiers in this comparison. The optimal threshold for each classifier varies,

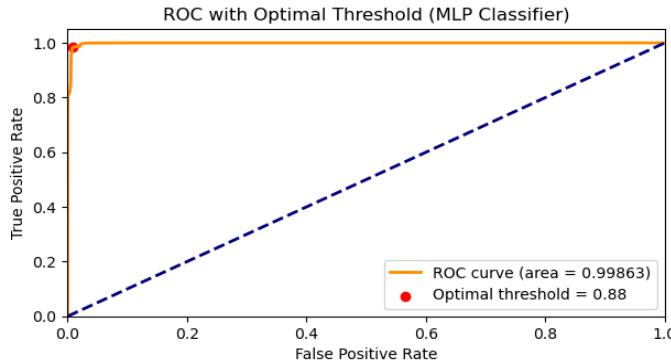


Fig. 6. Visualization of ROC curve and Optimal Threshold for MLP classifier.

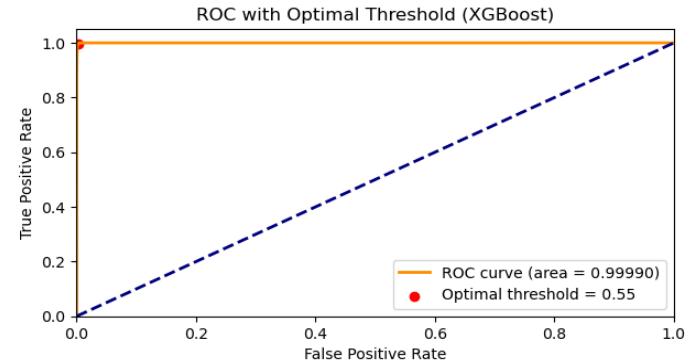


Fig. 9. Visualization of ROC curve and Optimal Threshold for XGBoost classifier.

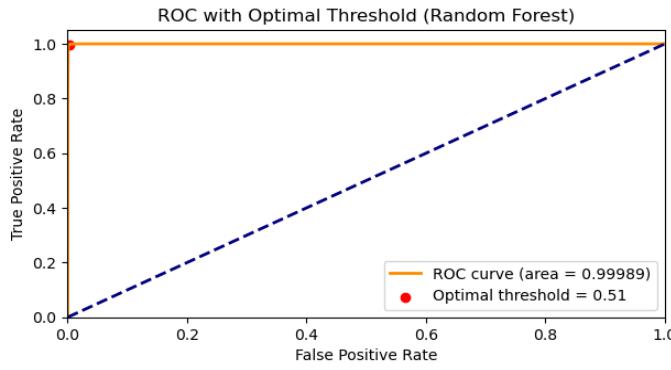


Fig. 7. Visualization of ROC curve and Optimal Threshold for Random Forest classifier.

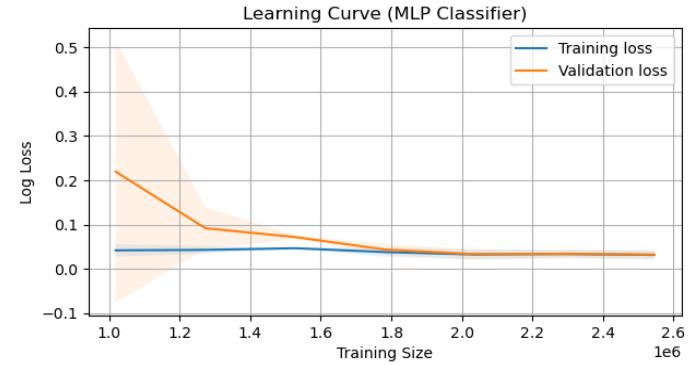


Fig. 10. Learning curve of the MLP classifier.

and these thresholds are used to evaluate the final performance metrics.

To analyze the learning curve, we refer to Figures 10 through 13. The analysis indicates that MLP exhibits better performance by converging at a smaller data size compared to the other models, while GBM shows weaker performance. Although early convergence with a smaller data size can carry the risk of overfitting, it also reduces the time required to train the model.

The confusion matrix for each classifier is depicted in

Figures 14 to 17, with the corresponding performance analysis metrics presented in Table I.

As mentioned earlier in the section titled "Model Training, Validation, and Best Model Selection," if a model achieved over 99% validation accuracy with its default settings, grid search was not applied. Consequently, grid search was only conducted for the MLP, as the ensemble learning classifiers already achieved more than 99% accuracy using their default scikit-learn settings.

During the grid search to optimize the MLP classifier, three

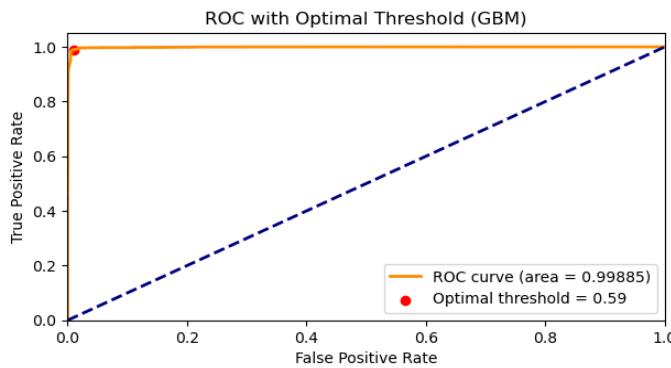


Fig. 8. Visualization of ROC curve and Optimal Threshold for GBM classifier.

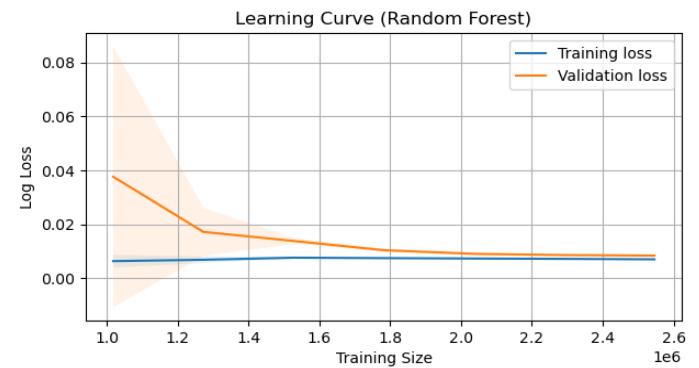


Fig. 11. Learning curve of the Random Forest classifier.

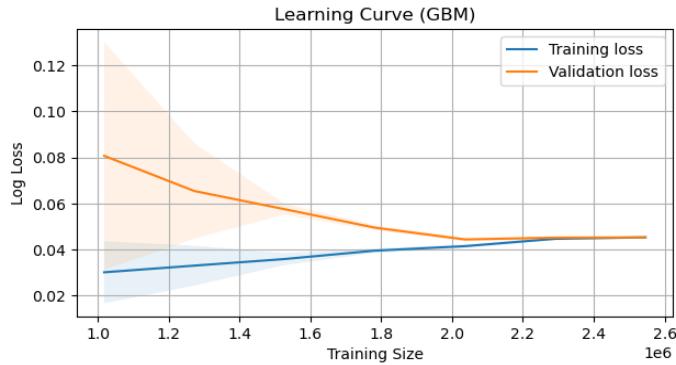


Fig. 12. Learning curve of the GBM classifier.



Fig. 13. Learning curve of the XGBoost classifier.

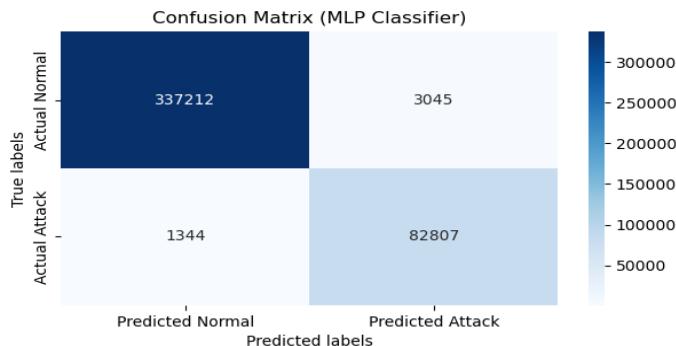


Fig. 14. Confusion matrix of the MLP classifier.

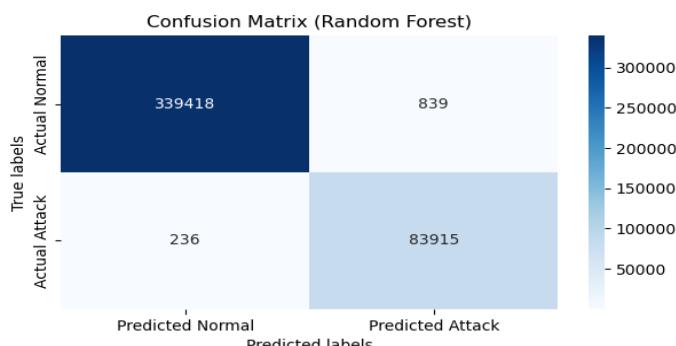


Fig. 15. Confusion matrix of the Random Forest classifier.

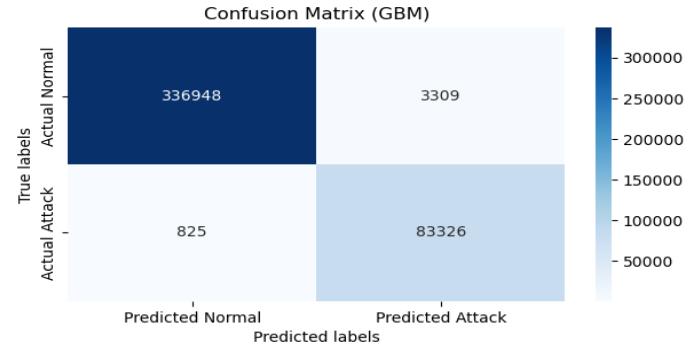


Fig. 16. Confusion matrix of the GBM classifier.

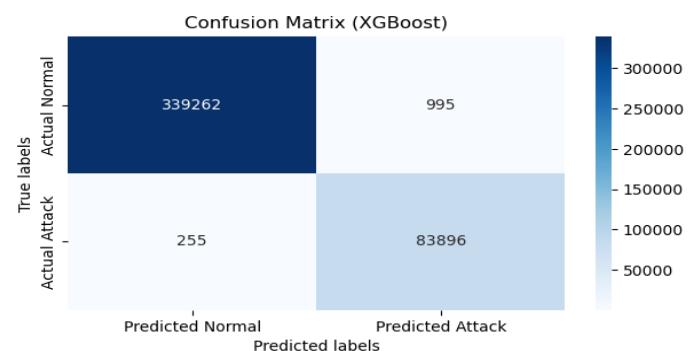


Fig. 17. Confusion matrix of the XGBoost classifier

layer architectures were tested: (50, 50, 50), (50, 100, 50), and (100,). Two optimization algorithms, `sgd` (Stochastic Gradient Descent) and `adam` (Adaptive Moment Estimation), were evaluated, along with two learning rate strategies: constant and adaptive. The grid search selected the best model as an MLP with `hidden_layer_sizes=(50, 50, 50)`, `solver='sgd'`, `learning_rate='adaptive'`, and a maximum of 5000 iterations. Despite these optimizations, the MLP achieved the lowest accuracy of 0.989658, with precision at 0.964531, recall at 0.984028, and an F1 score of 0.974182, indicating strong performance but still falling short compared to the ensemble classifiers.

Random Forest shows the highest accuracy at 0.997467, along with high precision (0.990100), recall (0.997195), and F1 score (0.993635). This indicates that Random Forest performs exceptionally well in terms of both precision and recall, leading to a balanced and robust model.

GBM (Gradient Boosting Machine) delivers solid performance, with an accuracy of 0.990259, indicating that it correctly classifies a large majority of instances. The precision is 0.961805, which shows that GBM has a strong ability to minimize false positives, though it is slightly lower compared to some other classifiers. The recall is high at 0.990196, reflecting GBM's effectiveness in identifying most true positive cases. The F1 score of 0.975794 indicates a good balance between precision and recall, though it is marginally lower than some other classifiers like Random Forest and XGBoost.

XGBoost also performs very well, with an accuracy of 0.997054, precision of 0.988279, recall of 0.996969, and an F1 score of 0.992605. Although XGBoost is slightly less accurate than Random Forest, it still provides strong results across all metrics.

TABLE I  
PERFORMANCE METRICS FOR CLASSIFIERS

Classifier	Accuracy	Precision	Recall	F1 Score
<b>MLP</b>	0.989658	0.964531	0.984028	0.974182
<b>Random Forest</b>	<b>0.997467</b>	<b>0.990100</b>	<b>0.997195</b>	<b>0.993635</b>
<b>GBM</b>	0.990259	0.961805	0.990196	0.975794
<b>XGBoost</b>	0.997054	0.988279	0.996969	0.992605

TABLE II  
PERFORMANCE METRICS FOR THE RANDOM FOREST CLASSIFIER AFTER RETRAINING WITH REDUCED DATA SIZE BASED ON THE LEARNING CURVE

Classifier	Accuracy	Precision	Recall	F1 Score
<b>Random Forest</b>	0.997594	0.990744	0.997184	0.993953

### III. CONCLUSION

In this study, we evaluated the performance of the MLP, Random Forest, GBM and XGBoost classifiers for intrusion detection using the CICIDS2017 dataset. Through extensive analysis, we observed that MLP exhibits a favorable learning curve, particularly in terms of convergence at smaller data sizes. Because of this characteristics, MLP can be trained in a shorter amount of time, which makes it a good option in situations where time and computational resources are few.

However, when considering the overall performance metrics, Random Forest emerged as the best classifier among those tested. It achieved the highest accuracy (0.997467), precision (0.990100), recall (0.997195), and F1 score (0.993635), indicating its robustness and reliability in detecting intrusions.

The impact of the ROC curves further supports these findings. While all classifiers demonstrated high ROC areas, Random Forest's ROC curve showed a nearly perfect area under the curve (AUC) of 0.99989, slightly lower than XGBoost. However, this indicates that Random Forest not only performs well in distinguishing between attack and benign traffic but also does so with exceptional accuracy and minimal false positives.

To further investigate, we analyzed the learning curve convergence when Random Forest was retrained on reduced data samples. The results presented in Table II indicated that, even with less data, Random Forest maintained a superior accuracy of 0.997594, which is nearly identical to the accuracy achieved with the full training dataset. This consistency suggests that the model avoids overfitting and continues to outperform other classifiers.

In conclusion, although MLP offers advantages in terms of training time and efficiency, Random Forest provides superior overall performance in the context of intrusion detection in this study.

### ACKNOWLEDGMENT

Research was sponsored by the DEVCOM Analysis Center and was accomplished under Cooperative Agreement Number W911NF-22-2-0001. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the DEVCOM Analysis Center or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

### REFERENCES

- [1] K. Shaukat, S. Luo, S. Chen, and D. Liu, "Cyber Threat Detection Using Machine Learning Techniques: A Performance Evaluation Perspective," in \*Proc. 2020 IEEE 17th International Conference on e-Business Engineering (ICEBE)\*, Guangzhou, China, 2020, pp. 42-49, doi: 10.1109/ICEBE51655.2020.00015.
- [2] M. Jump, "Fighting Cyberthreats with Technology Solutions," Biomedical instrumentation & technology, vol. 53, no. 1, pp. 38-43, 2019.
- [3] A. K. Jain, D. Goel, S. Agarwal, Y. Singh, and G. Bajaj, "Predicting Spam Messages Using Back Propagation Neural Network," Wireless Personal Communications, vol. 110, no. 1, pp. 403-422, 2020.
- [4] Patcha, A., & Park, J. M. (2007). "An overview of anomaly detection techniques: Existing solutions and latest technological trends." Computer Networks, 51(12), pp. 3448-3470.
- [5] Sommer, R., & Paxson, V. (2010). "Outside the closed world: On using machine learning for network intrusion detection." IEEE Symposium on Security and Privacy (SP), pp. 305-316.
- [6] Abaei, G., & Selamat, A. (2015). "A survey of anomaly detection methods in network intrusion detection systems." Computer Engineering and Applications Journal (ComEngApp), 4(1), pp. 51-58.
- [7] Sharafaldin, I., Habibi Lashkari, A., & Ghorbani, A. A. (2018). "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization." ICISSP, pp. 108-116.
- [8] Dhanabal, L., & Shanthalrajah, S. P. (2015). "A study on NSL-KDD dataset for intrusion detection system based on classification algorithms." International Journal of Advanced Research in Computer and Communication Engineering, 4(6), pp. 446-452.
- [9] T. J. Lucas, I. S. de Figueiredo, C. A. C. Tojeiro, A. M. G. de Almeida, R. Scherer, J. R. F. Brega, J. P. Papa, and K. A. P. da Costa, "A Comprehensive Survey on Ensemble Learning-Based Intrusion Detection Approaches in Computer Networks," IEEE Access, vol. 11, pp. 123456-123468, Oct. 2023, doi: 10.1109/ACCESS.2023.3328535.
- [10] B. A. Tama, L. Nkenyereye, S. M. R. Islam, and K.-S. Kwak, "An enhanced anomaly detection in web traffic using a stack of classifier ensemble," IEEE Access, vol. 8, pp. 24120-24134, 2020.
- [11] S. M. Nzuba, L. Nderu, and T. Mwalili, "Ensemble model for enhancing classification accuracy in intrusion detection systems," in Proc. Int. Conf. Electr., Comput. Energy Technol. (ICECET), Cape Town, South Africa, Dec. 2021, pp. 1-7.
- [12] K. Gosai, H. Mehta, and V. Katkar, "An intrusion detection using ensemble classifiers," in Proc. 6th Int. Conf. Comput. Methodol. Commun. (ICCMC), Mar. 2022, pp. 163-168.
- [13] F. Li, W. Ma, H. Li, and J. Li, "Improving intrusion detection system using ensemble methods and over-sampling technique," in Proc. 4th Int. Academic Exchange Conf. Sci. Technol. Innov. (IAECST), Dec. 2022, pp. 1200-1205.
- [14] J. Thaker, N. K. Jadav, S. Tanwar, P. Bhattacharya, and H. Shahinzadeh, "Ensemble learning-based intrusion detection system for autonomous vehicle," in Proc. 6th Int. Conf. Smart Cities, Internet Things Appl. (SCIoT), Sep. 2022, pp. 1-6.
- [15] D. N. Mhawi, A. Aldallal, and S. Hassan, "Advanced feature-selection based hybrid ensemble learning algorithms for network intrusion detection systems," Symmetry, vol. 14, no. 7, p. 1461, Jul. 2022.