Machine Learning-Based Intrusion Detection System: A Study on the CICIDS2017 Dataset

Cao Dang Khoa (Student ID: 23560023), Nguyen Quang Minh (Student ID: 23560032)
Faculty of Computer Science, University of Information Technology, HCMC
Lecturer: Dr. Nguyen Ngoc Tu, Faculty of Computer networks and Communications, UIT, HCMC

***Abstract*—This report explores the use of machine learning for building an Intrusion Detection System (IDS) based on the CICIDS2017 dataset. It evaluates three models: Random Forest, Random Forest retrained on selected features, K-Nearest Neighbors (KNN), and XGBoost. The focus is on identifying an effective approach for detecting cyber threats with high accuracy and reliability while minimizing false positives.**

***Index Terms*—Intrusion Detection System, Machine Learning, CICIDS2017, Random Forest, XGBoost, KNN, Cybersecurity**

I. Introduction

As cyberattacks grow more frequent and sophisticated, protecting computer networks has become critical. Intrusion Detection Systems (IDS) play an essential role in detecting and preventing attacks by automatically monitoring and analyzing network traffic. Their speed and ability to detect threats in real-time make them a powerful tool for cybersecurity.

An effective IDS must handle a wide variety of threats, ranging from simple scans to complex, multi-faceted attacks. The CICIDS2017 dataset reflects this diversity, including 15 types of traffic such as *BENIGN*, *DDoS*, *DoS Hulk*, *Web Attack – SQL Injection*, and more. It also highlights the challenge of detecting **cross-attacks**, where attackers combine different strategies.

By using machine learning, IDS can improve detection accuracy and reduce false alarms. This report evaluates models like Random Forest, K-Nearest Neighbors (KNN), and XGBoost to find the best approach for building a fast, effective, and adaptable IDS.

A. Objective

The primary goal of this study is to explore how machine learning models can be applied to develop an IDS using the CICIDS2017 dataset. By analyzing the performance of models such as Random Forest, K-Nearest Neighbors (KNN), and XGBoost, the study aims to identify strengths and limitations of these models in detecting various types of cyberattacks. This exploration will provide insights for further research and development of effective IDS solutions.

B. Scope

The dataset used includes detailed network traffic labeled as normal or malicious. Models such as Random Forest, Random Forest with selected features, K-Nearest Neighbors (KNN), and XGBoost are trained and evaluated based on key metrics such as accuracy, precision, recall, and F1-score.

II. Methodology

The methodology involves processing the dataset, training the machine learning models, and evaluating their performance.

A. Dataset and Preprocessing

The dataset used in this study is the CICIDS2017 dataset, which contains a wide range of features describing network traffic. These features include protocol type, service, byte counts, and various other metrics that are critical for identifying intrusions. The dataset includes 15 different attack types, which are labeled as either benign or malicious. A detailed list of attack types and their corresponding class indices is provided in Table I.

TABLE I: Attack Types and Their Corresponding Class Indices

Class Index	Attack Type
0	BENIGN
1	Bot
2	DDoS
3	DoS GoldenEye
4	DoS Hulk
5	DoS Slowhttptest
6	DoS slowloris
7	FTP-Patator
8	Heartbleed
9	Infiltration
10	PortScan
11	SSH-Patator
12	Web Attack - Brute Force
13	Web Attack - Sql Injection
14	Web Attack - XSS

Preprocessing steps include encoding the target variable ‘Label’ into numeric values, handling missing values by imputing them with the mean value, and normalizing numerical features using MinMax scaling. The dataset is split into training and testing sets, with 80% for training and 20% for testing. Additionally, feature selection is applied to identify the most relevant attributes for classification in the retrained Random Forest model.

B. Machine Learning Models

Four models are implemented:

1) **Random Forest:** An ensemble method combining multiple decision trees to enhance accuracy and reduce overfitting.
2) **Random Forest (Selected Features):** A retrained version of Random Forest focusing on the most significant features to improve computational efficiency.
3) **K-Nearest Neighbors (KNN):** A straightforward algorithm that classifies instances based on the labels of their nearest neighbors.
4) **XGBoost:** A gradient boosting algorithm known for its efficiency and high performance in structured data.

C. Model Evaluation and Performance Metrics

The performance of the machine learning models is evaluated using several standard metrics to assess their ability to classify network traffic correctly. The following metrics are used:

1) **Accuracy:** The proportion of correctly classified instances out of all instances in the dataset.
2) **Precision:** The proportion of true positive predictions out of all instances predicted as positive.
3) **Recall:** The proportion of true positive predictions out of all actual positive instances.
4) **F1-score:** The harmonic mean of precision and recall, providing a balanced measure of the model's performance.
5) **Training time:** The time taken to train the model on the training dataset.
6) **Testing time:** The time taken to evaluate the model on the test set.
7) **Performance for each Attack Type:** The model's ability to detect each specific attack type in the dataset, measured by precision and recall for each individual attack.

These metrics provide a comprehensive evaluation of each model's ability to detect both benign and malicious network traffic. The models will be compared based on their scores for these metrics, with the goal of identifying the most effective approach for intrusion detection.

III. Results

This section presents the evaluation results for each machine learning model applied to the Intrusion Detection System (IDS) task. The models are assessed on a range of metrics, including training time, testing time, accuracy, precision, recall, F1-score, and their performance in detecting specific attack types.

A. Individual Model Results

1) Random Forest (RF): Random Forest demonstrated robust performance with a perfect overall F1-score of 1.0. Its training and testing times were 112.13 seconds and 1.20 seconds, respectively. It achieved exceptional precision, recall, and F1-scores for most attack types, with slight challenges in detecting less frequent attacks like *Infiltration* and *Brute Force*.

2) Retrained Random Forest (RF): The retrained RF, optimized with hyperparameter tuning, showed a slight improvement in training (97.49 seconds) and testing time (0.96 seconds). While its overall F1-score remained perfect (1.0), it exhibited a minor decrease in precision and recall for attack types *DoS* and *Brute Force*.

3) K-Nearest Neighbors (KNN): KNN achieved an F1-score of 0.99. It was the fastest model to train (1.01 seconds), but its testing time (1010.92 seconds) was significantly longer, highlighting scalability concerns for real-time detection. While it performed well on common attack types, its ability to detect rare attacks, such as *Brute Force* and *DDoS*, was limited, resulting in undefined precision for these types.

4) XGBoost: XGBoost balanced speed and accuracy, with training and testing times of 84.97 seconds and 1.77 seconds, respectively. It achieved a perfect F1-score of 1.0 and consistently high precision and recall across most attack types. Notable challenges were observed in identifying attack types *DDoS* and *Brute Force*, similar to RF models.

B. Comparison of Models

The following images compare the performance metrics of the four models: training time, testing time, and F1 scores.

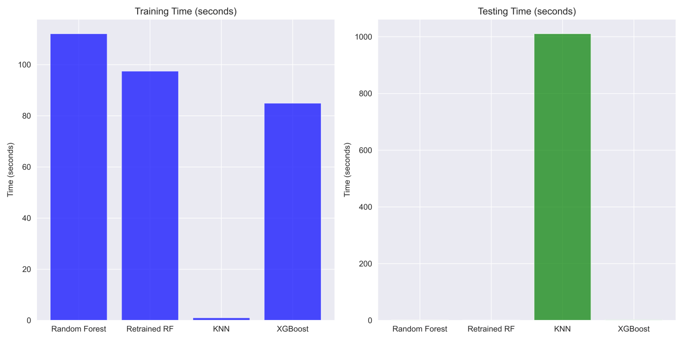

Fig. 1: Training Time and Testing Time Comparison Across Models.

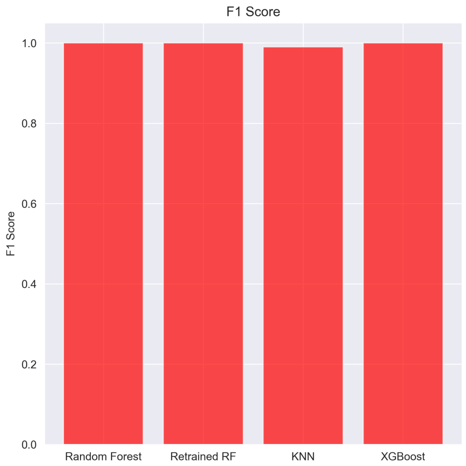

Fig. 2: F1 Score Comparison Across Models.

C. Per-Attack Metrics

The detailed F1-scores for each attack type are presented for the Random Forest model in Table II. Similar tables for other models are included in the Appendix.

TABLE II: Attack Type Performance for Models

Attack Type	**RF**	**RF-R**	**KNN**	**XGBoost**
Normal	1.00	1.00	1.00	1.00
Bot	0.83	0.84	0.75	0.83
DDoS	1.00	1.00	1.00	1.00
DoS GoldenEye	1.00	1.00	1.00	1.00
DoS Hulk	1.00	1.00	1.00	1.00
DoS Slowhttptest	1.00	1.00	1.00	1.00
DoS Slowloris	1.00	1.00	1.00	1.00
FTP-Patator	1.00	1.00	1.00	1.00
Heartbleed	1.00	1.00	1.00	1.00
Infiltration	0.83	0.84	0.75	0.83
PortScan	0.99	0.99	0.94	0.99
SSH-Patator	1.00	1.00	0.98	1.00
Web Attack - Brute Force	0.75	0.74	0.00	0.67
Web Attack - Sql Injection	0.33	0.00	0.00	0.50
Web Attack - XSS	0.49	0.40	0.74	0.57

D. Key Observations

- **Efficiency:** KNN, while fast to train, suffers from scalability issues due to its long testing time.
- **Effectiveness:** Random Forest, Retrained RF, and XGBoost achieved near-perfect scores across all major metrics.
- **Challenges:** Rare attack types such as *Infiltration*, *Brute Force*, and *Port Scan* were challenging for all models, indicating room for further refinement.

IV. Discussion

The results of the evaluation of the four machine learning models—Random Forest (RF), Retrained Random Forest (RF), K-Nearest Neighbors (KNN), and XGBoost—demonstrated their strong potential for intrusion detection. Overall, all models performed excellently on frequently occurring attack types such as *DDoS*, *DoS GoldenEye*, and *Normal*, achieving high F1-scores close to 1.00. However, there were notable differences in their performance for rare attack types like *Infiltration* and *Brute Force*.

A. Model Performance

- **Random Forest (RF)** and **Retrained RF** achieved near-perfect F1-scores across most attack types, with only a slight drop in the performance for the *Infiltration* attack type. This suggests that these models excel at detecting common attacks but may require further optimization, such as improving the handling of rare attacks.
- **K-Nearest Neighbors (KNN)** showed fast training times but struggled with scalability due to its significantly higher testing time. Despite this, KNN exhibited solid performance for common attacks, although its ability to detect rare attack types like *Infiltration* and *Brute Force* was limited, resulting in lower F1-scores for these attack types.
- **XGBoost** achieved the highest consistency across the board, balancing speed and accuracy. Its performance for *DDoS* and *DoS* related attacks was impressive, with high F1-scores. However, like other models, it showed a slight drop in performance for rare attacks, although not as severe as KNN.

B. Realistic Testing Considerations

In realistic or operational settings, performance metrics such as training time, testing time, and F1-scores are crucial. For example, *KNN's* long testing time makes it impractical for real-time intrusion detection systems. On the other hand, *Random Forest* and *XGBoost* offer better scalability, though they still face challenges with rare attacks. It is important to consider trade-offs between training/testing time and detection accuracy, especially in environments where both speed and accuracy are required. Further optimization or hybrid models may be necessary to address these concerns.

Additionally, the models performed well on the CICIDS2017 dataset, which serves as a reliable source for training and evaluating IDS. However, it is essential to recognize that real-world environments may present different challenges, such as varied attack strategies, network traffic, and adversarial attempts to evade detection. More extensive testing using diverse and real-world datasets could provide further insights into the models' robustness.

V. Conclusion

This study evaluated four popular machine learning models—Random Forest (RF), Retrained Random Forest (RF), K-Nearest Neighbors (KNN), and XGBoost—for the task of Intrusion Detection System (IDS) classification. The models were assessed based on various performance metrics, including F1-scores, precision, recall, and training/testing times.

- **Random Forest** and **Retrained RF** exhibited excellent detection capabilities for common attack types, with minor issues in detecting rare attacks.
- **KNN** performed efficiently in terms of training time but showed poor scalability due to high testing times, making it less suitable for real-time detection.
- **XGBoost** balanced accuracy and speed, offering consistent results across all attack types, albeit with slight challenges in detecting rare attacks.

While the models generally performed well, further improvements are needed to enhance the detection of rare attack types and reduce testing times for models like KNN. Future work could focus on optimizing these models for real-time IDS applications and evaluating their performance on real-world attack data.

In conclusion, the evaluation suggests that *Random Forest* and *XGBoost* are suitable for IDS applications, especially when balancing detection performance and computational efficiency.

Acknowledgment

This project is part of the final project for the CSBU111 Cybersecurity course. We would like to express our gratitude to Dr. Nguyen Ngoc Tu for their invaluable guidance and support throughout the course and the project.

You can access the full code and additional documentation on the project's GitHub repository: https://github.com/dkhoa2906/CSBU111-IDS-with-ML.

We thank the creators of the CICIDS2017 dataset and all contributors to the machine learning frameworks used in this study.

References

[1] Canadian Institute for Cybersecurity (CIC), *CICIDS2017: A Labeled Dataset for Network Intrusion Detection*, University of New Brunswick (UNB), https://www.unb.ca/cic/datasets/ids-2017.html, Accessed: Jan 2025.

[2] L. Breiman, *Random Forests*, Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.

[3] T. Cover and P. Hart, *Nearest Neighbor Pattern Classification*, IEEE Transactions on Information Theory, vol. 13, no. 1, pp. 21–27, 1967.

[4] T. Chen and C. Guestrin, *XGBoost: A Scalable Tree Boosting System*, Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785–794, 2016.

[5] Cao Dang Khoa, Nguyen Quang Minh, *CICIDS2017: Machine Learning-Based Intrusion Detection System Detection*, GitHub repository, https://github.com/dkhoa2906/CSBU111-IDS-with-ML, University of Information Technology (VNU-HCM), CSBU111.P11.KHBC, Accessed: Jan 2025.