AITA
ARTIFICIAL INTELLIGENCE
THEORY AND APPLICATIONS

■ RESEARCH ARTICLE

Intrusion Detection on CSE-CIC-IDS2018 Dataset Using Machine Learning Methods

Halil İbrahim Coşar[a†], **Çağrı Arısoy**[b], **Hasan Ulutaş**[b]

[a] Department of Electric-Electronic Engineering, Yozgat Bozok University, Yozgat, Türkiye
[b] Department of Computer Engineering, Yozgat Bozok University, Yozgat, Türkiye
† halil.cosar@bozok.edu.tr, corresponding author

RECEIVED SEPTEMBER 20, 2024
ACCEPTED SEPTEMBER 30, 2024

CITATION Coşar, H. İ., Arısoy, Ç., & Ulutaş. H. (2024). Intrusion detection on CSE-CIC-IDS2018 dataset using machine learning methods. *Artificial Intelligence Theory and Applications, 4(2)*, 143-154.

Abstract

Over the past few decades, the significance of computer and information security has grown exponentially, driven by the escalating frequency and sophistication of cyber threats. Despite the rapid advancements in both intrusion techniques and security technologies, many organizations continue to rely on outdated cybersecurity strategies, leaving them vulnerable to increasingly complex cyberattacks. Conventional defenses, such as basic firewalls and signature-based detection systems, are often insufficient against modern attackers who use advanced methods, including zero-day exploits and polymorphic malware, to evade detection. Government web servers, which house vast amounts of sensitive citizen data, are especially attractive targets for malicious actors. In response to these evolving threats, the deployment of an Intrusion Detection System (IDS) has become a critical component in securing network infrastructures, providing an essential layer of defense against unauthorized access and data breaches. This study explores the efficacy of six distinct machine learning-based classification methods; Random Forest, Gradient Boosting, XGBoost, CatBoost, Logistic Regression, and LightGBM each selected for its particular strengths in handling complex, high-dimensional data. These algorithms were applied to a comprehensive dataset to detect malicious activities, with a focus on achieving high accuracy and robustness in classification performance. Remarkably, all six models demonstrated substantial effectiveness, achieving accuracy rates as high as 0.98 and AUC values reaching 1.00, underscoring their potential in enhancing IDS capabilities. The results highlight the importance of leveraging advanced machine learning techniques in bolstering cybersecurity defenses, particularly in critical domains like government data protection, where precision and reliability are paramount.

Keywords: network intrusion, classification, cyber security, machine learning

1. Introduction

Computer and information security has become an increasingly significant issue over the past decades. While intrusion techniques and security protections have advanced rapidly, many organizations continue to rely on outdated cybersecurity measures. These traditional defences are often inadequate against modern cyberattacks, which use sophisticated methods to bypass them. Government web servers, which store sensitive

Artificial Intelligence Theory and Applications, ISSN: 2757-9778. ISBN: 978-605-69730-2-4 © 2024 İzmir Bakırçay University

information about citizens, are particularly attractive targets for hackers [1]. Today, an Intrusion Detection System (IDS) is an essential defence mechanism critical for safeguarding important networks against intrusions [2]. IDSs can be categorized into two types: anomaly-based and signature-based. Anomaly-based IDSs operate by creating a model of normal system behaviour and identifying any deviations from this baseline. In contrast, signature-based IDSs rely on a database of known attack signatures to recognize malicious activities [3]. In the commercial sector, signature-based IDSs are commonly employed. However, anomaly-based IDSs have the advantage of being able to detect previously unknown attacks. Despite this, anomaly-based IDSs typically suffer from low detection rates and high false positive rates. To improve the detection of new attacks, adaptive and efficient Machine Learning (ML) and Deep Learning (DL) algorithms are frequently utilized [4].

2. Related Work

Two recent public datasets, CICIDS2017 [5] and CSE-CIC-IDS2018 [6], are now available and include normal traffic as well as contemporary attack scenarios such as Heartbleed, Brute-force, Botnet, and Denial of Service (DoS). Although these datasets are accessible to the public, there has been limited use of them for evaluating, testing, and fine-tuning real-time IDS deployments.

Atefinia and Ahmadi [1] propose a multi-architectural modular deep neural network model aimed at enhancing anomaly-based intrusion detection systems by reducing the false-positive rate. This model includes a feed-forward module, a stack of restricted Boltzmann machine modules, and two recurrent modules, with their output weights combined in an aggregator module to make the final decision. Experiments using the CSE-CIC-IDS2018 dataset show significant improvements in detecting specific network attacks, achieving up to 100% accuracy for certain network-level attacks compared to existing methods. The models developed in this study can be effectively used in IDS to generate alerts or prevent new attacks. This deep neural network model offers a promising solution to the limitations of traditional signature-based intrusion detection systems by utilizing machine learning techniques to detect network attacks without relying solely on predefined signatures. In Basnet et al. [7] deep learning algorithms have demonstrated significant potential in network intrusion detection, as evidenced. Researchers assessed the effectiveness of several state-of-the-art deep learning frameworks, including Keras, TensorFlow, Theano, fast.ai, and PyTorch, in identifying and classifying network intrusion traffic. Using the CSE-CIC-IDS2018 dataset to evaluate these frameworks, fast.ai, a PyTorch wrapper, achieved the highest accuracy, approximately 99%, with low false positive and false negative rates in detecting and classifying various types of network intrusions. This high level of accuracy underscores the potential of deep learning frameworks in effectively identifying and categorizing network attacks. The results strongly support the effectiveness and utility of deep learning frameworks in network intrusion detection, emphasizing the importance of leveraging these techniques to enhance cybersecurity measures and effectively combat evolving cyber threats. Another paper evaluated two traditional training algorithms for Hidden Markov Models (HMM), Baum Welch (BW) and Viterbi Training (VT), using three standard initialization techniques: uniform, random, and count-based. The performance of the HMM was analysed based on detecting all states (AS), the current state (CS), and the next state (NS) given an observation sequence. The count-based initialization technique outperformed the uniform and random techniques in detecting AS and CS, achieving about 97.5% and 97.0% accuracy for AS prediction using BW and VT, respectively. For CS detection, the performance was similar to AS detection, with a slight decrease of about 0.2%. Predicting NS had an accuracy of around 65% for both uniform

and random initialization techniques with BW and VT. The study found no significant improvement with increasing the window sample size, and the training techniques can be practically implemented by connecting the output of an IDS or a database storing alerts to an HMM [4]. In the other study explored the inter-dataset generalization of supervised machine learning methods for intrusion detection, aiming to differentiate between benign and various types of malicious network traffic. Classification benchmarks were established using two labelled datasets, CIC-IDS2017 and CSE-CIC-IDS2018, which include attack classes such as DoS, DDoS, infiltration, and botnet. Twelve supervised learning algorithms from different families were compared. The research revealed that high generalization within a dataset does not necessarily translate to high generalization across different datasets, especially for attack types like DoS/SSL and botnet. The trained models failed to maintain high classification performance when tested on new but related samples without additional training. These findings challenge the assumption that strong intra-dataset performance guarantees strong inter-dataset performance. Further investigation is needed to understand the limitations and develop solutions to enhance inter-dataset generalization in supervised ML-based intrusion detection systems [8]. Another paper presented a comparative analysis of deep learning methods for intrusion detection, specifically examining deep discriminative models and generative unsupervised models. Seven different deep learning techniques were evaluated: recurrent neural networks (RNNs), deep neural networks (DNNs), restricted Boltzmann machines (RBMs), deep belief networks (DBNs), convolutional neural networks (CNNs), deep Boltzmann machines (DBMs), and deep autoencoders. The evaluation was conducted using two novel datasets, CSE-CIC-IDS2018 and Bot-IoT, and was based on three primary performance metrics: false alarm rate, accuracy, and detection rate. The goal of the study was to assess the effectiveness of these deep learning models in various intrusion detection scenarios, offering insights into their performance for both binary and multiclass classification tasks. The findings are crucial for advancing cybersecurity measures by employing sophisticated deep learning techniques, thereby enhancing the accuracy and efficacy of intrusion detection systems in identifying cyber threats. Deep autoencoders exhibited the highest accuracy on both the CSE-CIC-IDS2018 and Bot-IoT datasets, with accuracy rates of 97.372 and 98.394, respectively. These results were achieved using a configuration of 100 hidden nodes and a learning rate of 0.5 [9]. Fitni and Ramli employed ensemble learning, which combined logistic regression, decision trees, and gradient boosting, to increase the performance of intrusion detection systems This method harnessed the strengths of each classifier to enhance detection accuracy, minimize false alarms, and improve the identification of unknown attacks. Feature selection techniques were used to pinpoint the most critical data features for intrusion detection. Using Spearman’s rank correlation coefficient, 23 out of 80 features were selected, enhancing the model's efficiency by concentrating on the most informative features. The proposed model achieved high performance on the CSE-CIC-IDS2018 dataset, attaining an accuracy of 98.8%, precision of 98.8%, recall of 97.1%, and an F1 score of 97.9%. These results underscore the effectiveness of ensemble learning and feature selection in improving anomaly-based intrusion detection systems, significantly enhancing detection capabilities, reducing false alarms, and bolstering overall network security within organizational information systems [10]. Kanimozhi and Jacob proposed a system which applies AI to the CSE-CIC-IDS2018 dataset and achieves outstanding performance metrics: 99.97% accuracy, an average area under the ROC curve of 0.999, and a low false positive rate of 0.001. These results highlight the system's high accuracy and precision in detecting botnet attacks. Its effectiveness in identifying botnet attacks underscores its potential to improve security in financial sectors and banking services, where such threats are particularly serious. This demonstrates the practical importance and applicability of AI-based intrusion detection systems in protecting critical systems and data. Additionally,

the system's scalability allows for deployment across multiple machines, making it suitable for various applications such as network traffic analysis, cyber-physical system traffic data analysis, and real-time network traffic monitoring. This versatility enhances its relevance and utility in diverse cybersecurity contexts [11]. In another study, six machine learning models were implemented using the CSE-CIC-IDS2018 dataset. Data sampling techniques, such as the Synthetic Minority Oversampling Technique (SMOTE), were applied to increase the representation of minority classes and enhance detection rates for less common intrusions. The experimental results indicated that the implemented models achieved a high level of accuracy compared to recent studies. Using a sampled dataset led to an increase in the average accuracy of the models by between 4.01% and 30.59% [12].

3. Materials and Methods

3.1. Dataset

The CSE-CIC-IDS2018 dataset contains network traffic data from various services and protocols, predominantly HTTPS and HTTP, along with others like SMTP, POP3, IMAP, SSH, and FTP. It includes numerous attack scenarios. The final dataset encompasses seven distinct attack scenarios: brute-force attacks, Heartbleed exploitation, botnet activity, DoS (Denial of Service), DDoS (Distributed Denial of Service), web-based attacks, and internal network infiltration. The attacking infrastructure is composed of 50 machines, while the targeted organization includes five departments, comprising 420 computers and 30 servers. The network traffic from this dataset was processed using the CICFlowMeter-V3 tool, extracting 80 features for training, such as the number of packets per second, specific TCP flag packet counts, and the standard deviation of packet sizes in a session [6].

3.2. Preprocessing

In the data set one file includes 84 features and this file was not processed because files with an equal number of features were processed in this study. Then, the intrusions within the CIC-IDS2018 training dataset were categorized into two traffic types: benign and attack. To streamline the experiments and ensure clarity, any data points containing Infinity or NaN values were excluded from the dataset, which also helped improve the quality of the input data for the models. In cases where text data was present, it was converted to float to ensure uniformity in the dataset and to facilitate the mathematical operations needed for machine learning models. Timestamps, which did not contribute significantly to the feature space, were removed from the dataset to avoid any potential bias in time-based patterns. Following this, the dataset underwent a normalization process using the StandardScaler technique. This approach scales the data such that it has a mean of 0 and a standard deviation of 1, which is often critical for models that are sensitive to the scale of features. Normalization helps ensure that features with varying ranges do not disproportionately influence the model's learning process, resulting in a more balanced and accurate performance. The pre-processed dataset was then split into training and validation sets in an 80-20 ratio, with 80% of the data allocated for training and 20% reserved for validation. The training set was employed to fit the machine learning models, while the validation set was used to evaluate the final model performance, ensuring that the models could generalize well to unseen data. To address the issue of class imbalance, an under-sampling technique was applied to the training set. This process involved reducing the number of samples in the majority class, which in this case was the benign traffic data, to match or closely match the minority class, representing the attack traffic. By randomly removing excess samples from the majority

class, a more balanced dataset was created, which helped the models avoid overfitting to the dominant class and improved their ability to detect intrusions in the minority class. This step was crucial for enhancing model accuracy, particularly in imbalanced data scenarios where the majority class can overwhelm the learning algorithm.

3.3. Evaluation metrics

Various metrics are commonly used to assess and compare the performance of machine learning classifiers. The proposed model was evaluated using the following performance metrics.
Accuracy: Measures the percentage of correctly classified samples out of the total number of samples. The formula is:

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

Recall (Sensitivity): The ratio of correctly classified samples of a specific category (X) to the total samples of that category, indicating the system's effectiveness in detecting anomalies.

$$Recall = \frac{TP}{TP + FN} \quad (2)$$

Precision: Represents the ratio of correctly predicted positive observations to the total predicted positives.

$$Precision = \frac{TP}{TP + FP} \quad (3)$$

F1 Score: The harmonic mean of precision and recall, accounting for both false positives and false negatives.

$$F1\ Score = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (4)$$

where TP, FP, TN, and FN represent true positives, false positives, true negatives, and false negatives, respectively.

3.4. Classification

In this study, six different classification methods: random forest, gradient boosting, XGBoost, CatBoost, logistic regression, and LightGBM were implemented. Each of these methods was chosen for its unique strengths. Random forest reduces overfitting and handles noisy data well by constructing multiple decision trees. Gradient boosting incrementally improves performance by correcting errors from weak learners. XGBoost, an optimized version of GBM, offers faster performance and handles large datasets effectively. CatBoost excels with categorical data and requires less preprocessing. Logistic regression provides a simple yet powerful approach for linear relationships and is easily interpretable. LightGBM is optimized for large datasets and delivers high-speed performance with low memory usage. By using these diverse methods, it is aimed to explore various model structures and approaches to achieve optimal classification performance based on the dataset's characteristics. In the classification, the system being used is equipped with 64 GB of memory and is powered by two Intel(R) Xeon(R)

Silver 4114 CPUs, each running at 2.20 GHz. The server model is an HP Z6 G4, and it features an NVIDIA GeForce RTX 3090 Ti graphics card. The operating system is Windows 10 Pro for Workstations, and Python 3 is the language being used within the Jupyter Notebook framework.

4. Results and Discussion

While the results of used classification algorithm are analysed, three important visuals are used which are confusion matrix, ROC (Receiver Operating Characteristic) curve and learning curve. The ROC curve, learning curve, and confusion matrix are essential tools for evaluating classification models. The ROC curve plots the true positive rate (sensitivity) against the false positive rate, helping to assess a model's performance across different thresholds and its ability to distinguish between classes. The area under the ROC curve (AUC) is a key metric, where a higher value indicates better performance. The learning curve shows how a model's accuracy or error rate changes with varying amounts of training data, offering insights into whether the model is underfitting or overfitting and how it improves as it learns from more data. Finally, the confusion matrix provides a detailed breakdown of the model's predictions, showing true positives, true negatives, false positives, and false negatives, enabling a more granular understanding of classification accuracy and potential misclassifications. Together, these tools give a comprehensive view of a model's effectiveness, training behaviour, and areas for improvement.

RF	*precision*	*recall*	*f1-score*	*Support*
0	0.97	0.95	0.96	434183
1	0.98	0.99	0.99	1222666
accuracy			0.98	1656849
macro avg	0.97	0.97	0.97	1656849
weighted avg	0.98	0.98	0.98	1656849

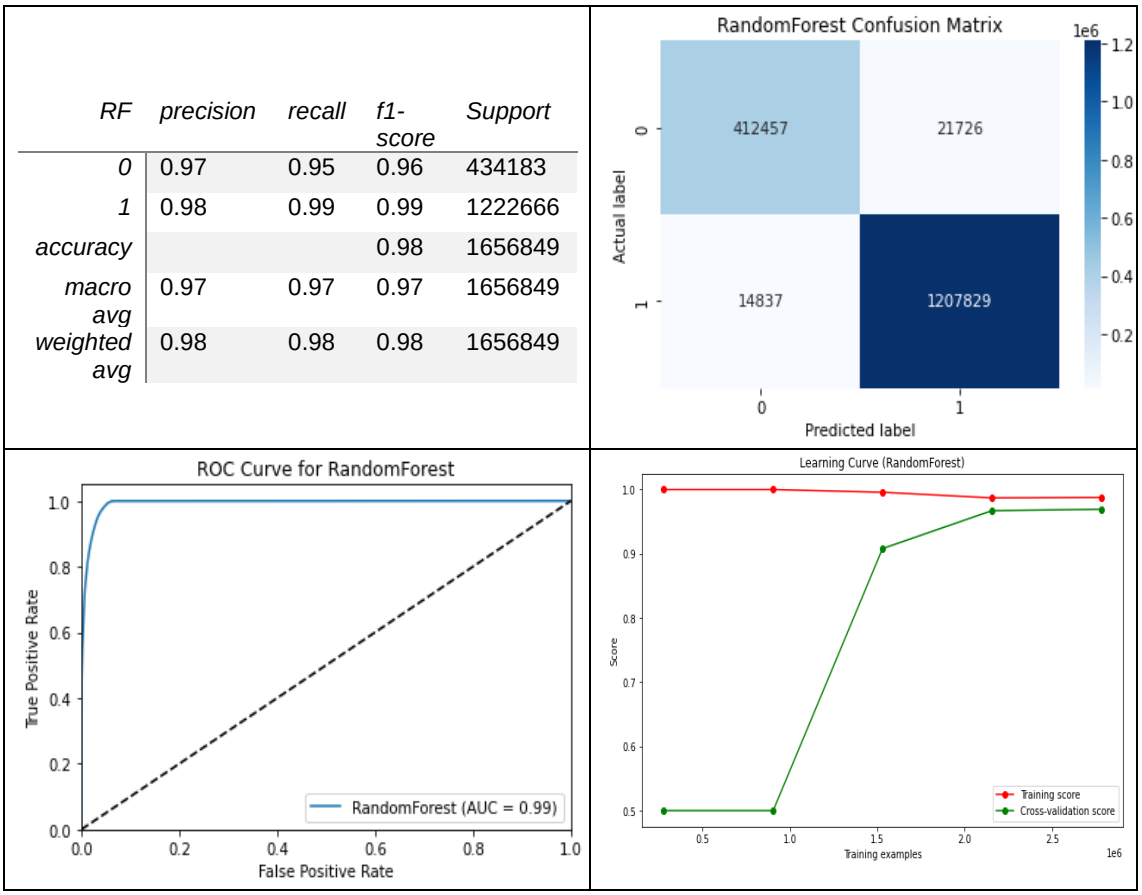
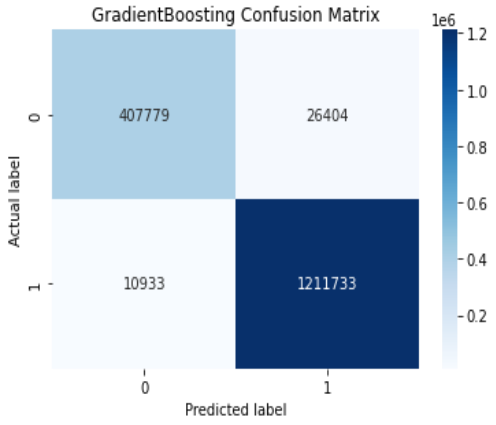


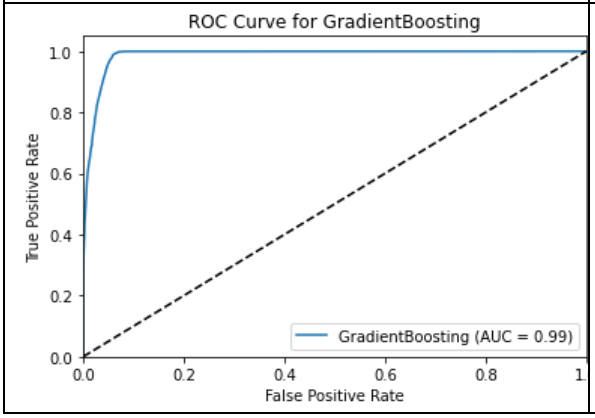
Figure 1. Random Forest classification report

The performance of the machine learning algorithms used in this study is illustrated in Figures 1-6. Based on these results, it is evident that all six techniques demonstrate exceptional performance on the given dataset, highlighting their suitability for network intrusion detection tasks. XGBoost, LightGBM, and CatBoost emerge as the top-performing models, achieving an impressive accuracy rate of 0.98 and an AUC score of 1.00, signifying near-perfect classification capabilities. These results suggest that these gradient-boosting-based methods are highly effective at distinguishing between normal and malicious network traffic, likely due to their advanced handling of complex interactions and non-linear relationships within the data.

Similarly, the Gradient Boosting and Random Forest algorithms also achieve strong performance, reaching an accuracy rate of 0.98 and an AUC value of 0.99. While slightly below the top-performing models, these results still demonstrate robust classification abilities, confirming their reliability in identifying potential intrusions. The success of these ensemble methods may be attributed to their ability to reduce overfitting and enhance model generalization by combining the predictions of multiple trees.

GB	*precision*	*recall*	*f1-score*	*support*
0	0.97	0.94	0.96	434183
1	0.98	0.99	0.98	1222666
accuracy			0.98	1656849
macro avg	0.98	0.97	0.97	1656849
weighted avg	0.98	0.98	0.98	1656849





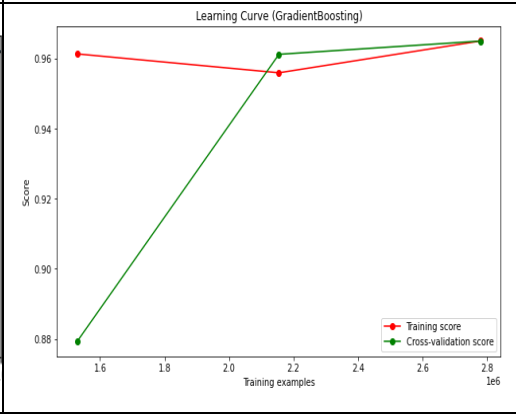
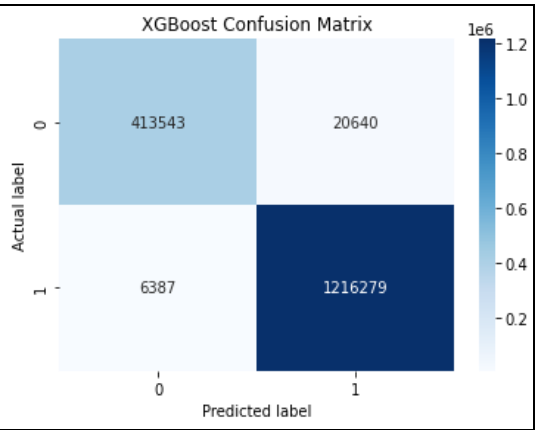
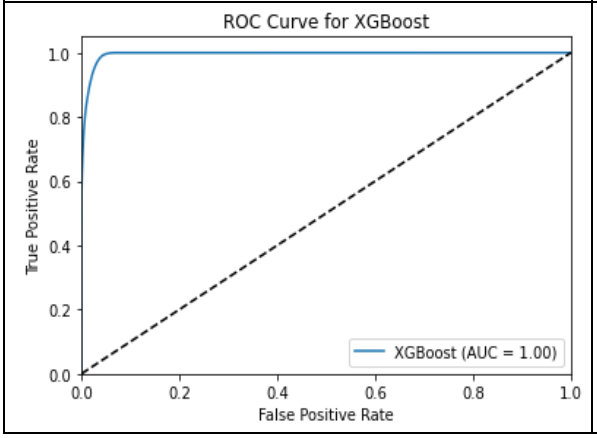


Figure 2. Gradient Boosting classification report

XGBoost	precision	recall	f1-score	support
0	0.98	0.95	0.97	434183
1	0.98	0.99	0.99	1222666
accuracy			0.98	1656849
macro avg	0.98	0.97	0.98	1656849
weighted avg	0.98	0.98	0.98	1656849





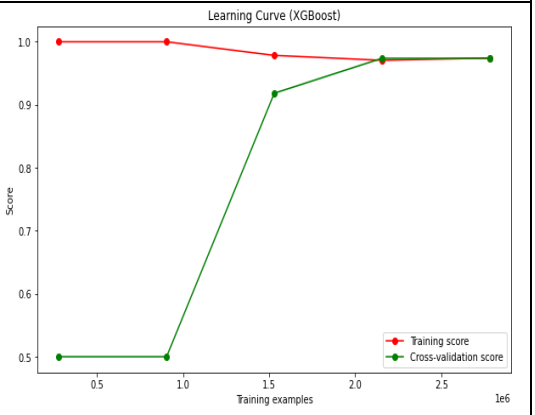
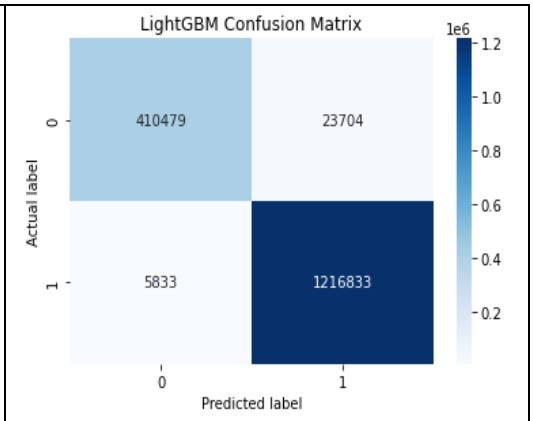
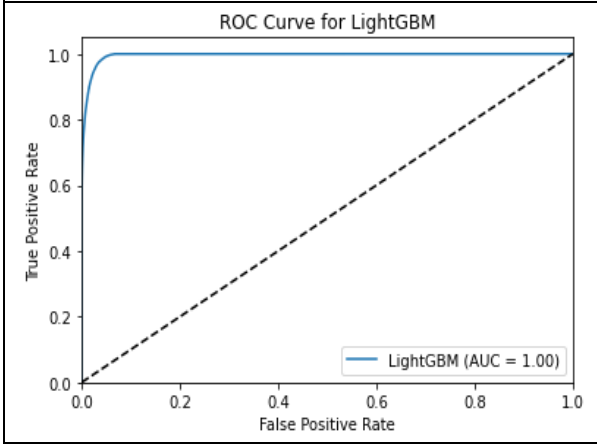


Figure 3. XGBoost classification report

LightGBM	precision	recall	f1-score	support
0	0.99	0.95	0.97	434183
1	0.98	1.00	0.99	1222666
accuracy			0.98	1656849
macro avg	0.98	0.97	0.98	1656849
weighted avg	0.98	0.98	0.98	1656849





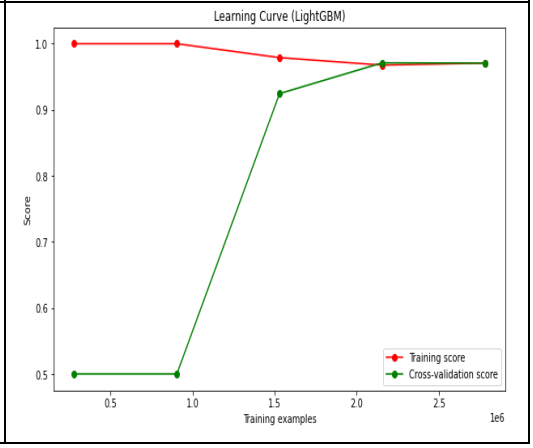
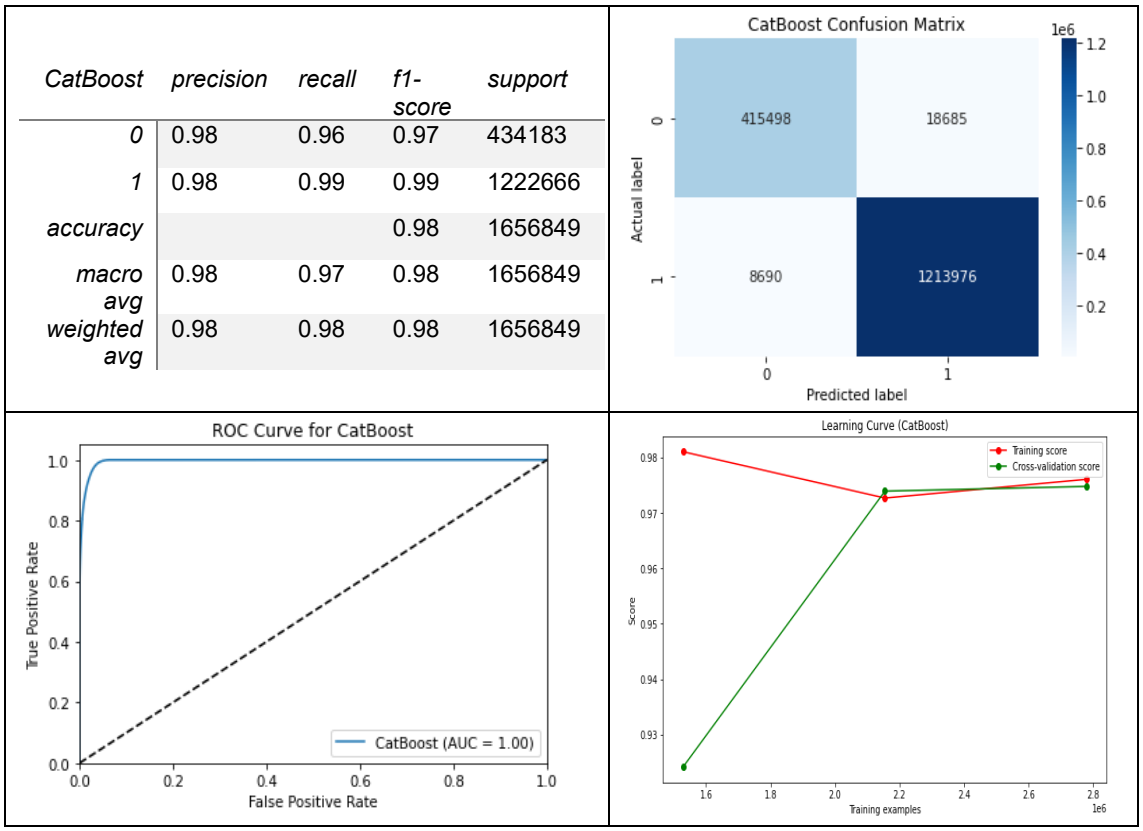


Figure 4. LightGBM classification report



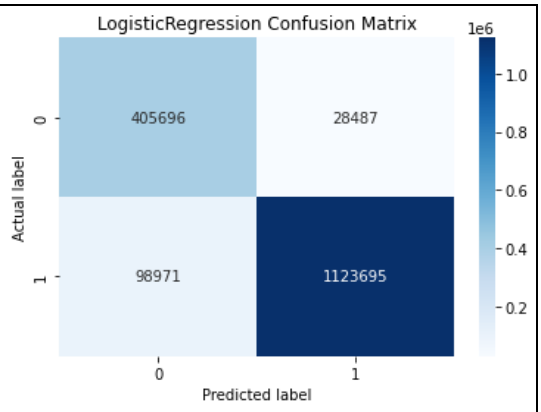
CatBoost	precision	recall	f1-score	support
0	0.98	0.96	0.97	434183
1	0.98	0.99	0.99	1222666
accuracy			0.98	1656849
macro avg	0.98	0.97	0.98	1656849
weighted avg	0.98	0.98	0.98	1656849

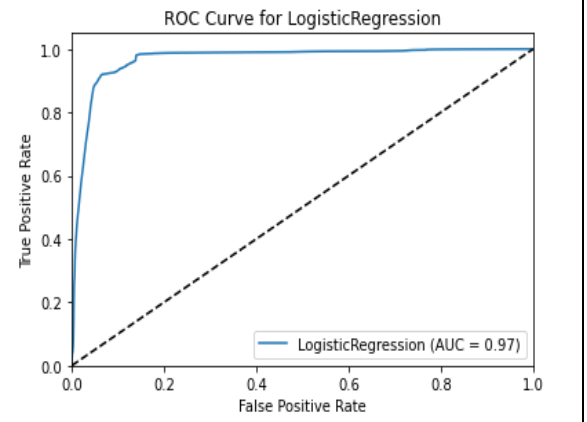
Figure 5. CatBoost classification report

In contrast, Logistic Regression, while comparatively less successful than the ensemble-based techniques, still delivers commendable results, with an accuracy of 0.92 and an AUC score of 0.97. Although it does not match the performance of the tree-based models, these results indicate that Logistic Regression remains a viable option for intrusion detection, particularly in scenarios where interpretability and simplicity are prioritized. Its lower performance could be due to its linear nature, which may limit its ability to capture more complex relationships in the data compared to non-linear models like gradient boosting or random forests.

In summary, while all the algorithms show strong performance, the results suggest that gradient-boosting-based methods, particularly XGBoost, LightGBM, and CatBoost, offer superior accuracy and AUC values, making them ideal for network intrusion detection. The relatively lower performance of Logistic Regression, although still effective, highlights the importance of algorithm selection based on the complexity and nature of the dataset. In conclusion, while all the algorithms demonstrate solid performance, gradient-boosting-based methods, specifically XGBoost, LightGBM, and CatBoost, stand out by providing the highest accuracy, making them particularly well-suited for network intrusion detection tasks. Although Logistic Regression performs adequately, its comparatively lower results emphasize the significance of choosing the right algorithm based on the dataset's complexity and characteristics.

LR	precision	recall	f1-score	support
0	0.80	0.93	0.86	434183
1	0.98	0.92	0.95	1222666
accuracy			0.92	1656849
macro avg	0.89	0.93	0.91	1656849
weighted avg	0.93	0.92	0.92	1656849





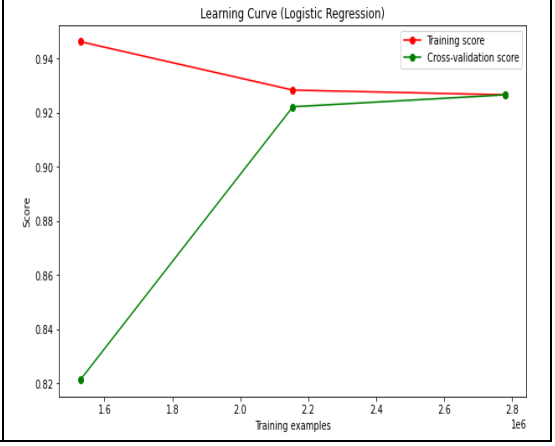


Figure 6. Logistic Regression classification report

During the classification process 5-fold cross validation was applied in order to evaluate the performance and generalization ability of machine learning models. The results are shown in figure 7. K-fold cross-validation bar chart provides a comparative visual representation of how well different classification algorithms performed on the dataset. The model with the longest bar was the most successful in classifying data consistently across all folds, while the algorithms with shorter bars were less accurate or consistent. This visual helps identify the strongest classification model, with attention to the differences in performance.

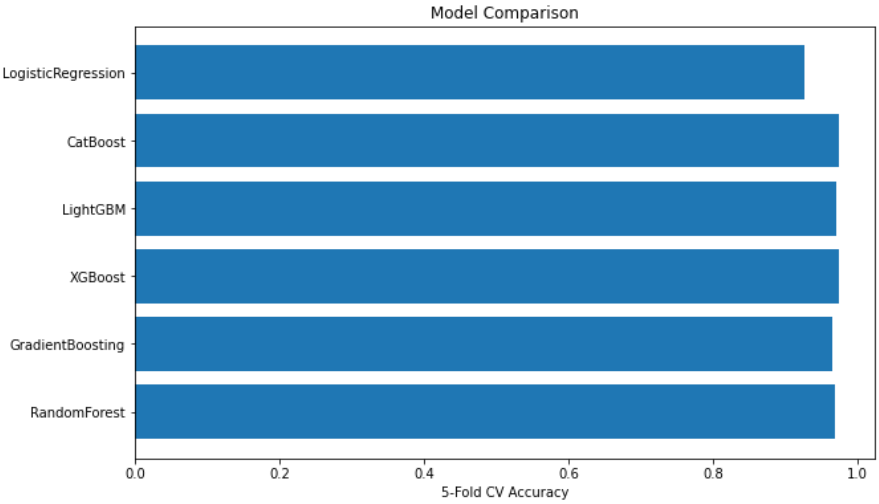


Figure 7. 5-fold cross validation result

5. Conclusion and Future Work

In this study, six different machine learning methods which are Random Forest, Gradient Boosting, XGBoost, CatBoost, Logistic Regression, and LightGBM are applied for detecting intrusions in network traffic, each demonstrating considerable potential in enhancing IDS. Our experimental results underscore the effectiveness of these algorithms, particularly when paired with appropriate preprocessing techniques. By reducing false positives for certain types of intrusions and achieving an accuracy rate of up to 98%, these methods offer promising alternatives to conventional detection systems. The performance we observed is not only competitive but also exceeds the benchmarks reported in much of the existing literature, highlighting the significance of integrating machine learning approaches for network security.

Despite the success of these models, there remain numerous opportunities for future research. One key direction would be to further refine feature extraction techniques to more accurately capture the characteristics of network traffic, particularly for anomaly-based intrusion detection systems. The integration of advanced feature engineering, or the use of deep learning-based automatic feature extraction, could potentially uncover hidden patterns in network data, further improving detection accuracy and reducing false alarms. Moreover, different types of datasets, including real-world network traffic from varied domains, could be explored using the methodology outlined in this research. This would provide a broader understanding of how these algorithms generalize across diverse environments and attack scenarios.

Another promising area for future work is the exploration of hybrid models that combine the strengths of multiple machine learning techniques, or the development of ensemble methods tailored specifically to network intrusion detection. Additionally, the impact of real-time data processing and online learning could be investigated to assess how well these models perform in dynamic environments where network conditions change frequently. Finally, further investigation into model interpretability and the ability to explain detection decisions will be crucial for fostering trust in machine learning-driven IDS systems, especially in high-stakes domains like government, healthcare, and financial networks.

By continuing to build upon the findings of this study, future research has the potential to significantly advance the capabilities of IDS systems, leading to more robust and adaptive network security solutions capable of defending against increasingly sophisticated cyber threats.

References

[1] Atefinia, R., & Ahmadi, M. (2021). Network intrusion detection using multi-architectural modular deep neural network. Journal of Supercomputing, 77(4), 3571–3593. https://doi.org/10.1007/S11227-020-03410-Y/FIGURES/14

[2] Sharafaldin, I., Lashkari, A. H., & Ghorbani, A. A. (2018). Toward generating a new intrusion detection dataset and intrusion traffic characterization. ICISSP 2018 - Proceedings of the 4th International Conference on Information Systems Security and Privacy, 2018-January, 108–116. https://doi.org/10.5220/0006639801080116

[3] Hussein, S. M. (2016). Performance Evaluation of Intrusion Detection System Using Anomaly and Signature Based Algorithms to Reduction False Alarm Rate and Detect Unknown Attacks. 2016 International Conference on Computational Science and Computational Intelligence (CSCI), 1064–1069. https://doi.org/10.1109/CSCI.2016.0203

[4] Chadza, T., Kyriakopoulos, K. G., & Lambotharan, S. (2019). Contemporary Sequential Network Attacks Prediction using Hidden Markov Model. 2019 17th International Conference on Privacy, Security and Trust, PST 2019 - Proceedings. https://doi.org/10.1109/PST47121.2019.8949035

[5] IDS 2017 | Datasets | Research | Canadian Institute for Cybersecurity | UNB. (n.d.). Retrieved May 29, 2024, from https://www.unb.ca/cic/datasets/ids-2017.html

[6] IDS 2018 | Datasets | Research | Canadian Institute for Cybersecurity | UNB. (n.d.). Retrieved May 29, 2024, from https://www.unb.ca/cic/datasets/ids-2018.html

[7] Basnet, R., Johnson, C., Basnet, R. B., Shash, R., Walgren, L., & Doleck, T. (n.d.). Towards Detecting and Classifying Network Intrusion Traffic Using Deep Learning Frameworks. Researchgate.NetRB Basnet, R Shash, C Johnson, L Walgren, T DoleckJ. Internet Serv. Inf. Secur., 2019•researchgate.Net. https://doi.org/10.22667/JISIS.2019.11.30.001

[8] D'hooge, L., Wauters, T., Volckaert, B., & De Turck, F. (2020). Inter-dataset generalization strength of supervised machine learning methods for intrusion detection. Journal of Information Security and Applications, 54, 102564. https://doi.org/10.1016/J.JISA.2020.102564

[9] Ferrag, M. A., Maglaras, L., Moschoyiannis, S., & Janicke, H. (2020). Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study. Journal of Information Security and Applications, 50, 102419. https://doi.org/10.1016/J.JISA.2019.102419

[10] Fitni, Q. R. S., & Ramli, K. (2020). Implementation of ensemble learning and feature selection for performance improvements in anomaly-based intrusion detection systems. Proceedings - 2020 IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology, IAICT 2020, 118–124. https://doi.org/10.1109/IAICT50021.2020.9172014

[11] Kanimozhi, V., & Prem Jacob, T. (2019). Artificial intelligence based network intrusion detection with hyper-parameter optimization tuning on the realistic cyber dataset CSE-CIC-IDS2018 using cloud computing. Proceedings of the 2019 IEEE International Conference on Communication and Signal Processing, ICCSP 2019, 33–36. https://doi.org/10.1109/ICCSP.2019.8698029

[12] Karatas, G., Demir, O., & Sahingoz, O. K. (2020). Increasing the Performance of Machine Learning-Based IDSs on an Imbalanced and Up-to-Date Dataset. IEEE Access, 8, 32150–32162. https://doi.org/10.1109/ACCESS.2020.2973219