JCBI
Journal of Computing & Biomedical Informatics
ISSN: 2710 - 1606
Volume 07 Issue 01
2024

Research Article
https://doi.org/10.56979/701/2024

Machine Learning for Improved Threat Detection: LightGBM vs. CatBoost

Moeed Saleem[1], Muhammad Azam[1], Zeeshan Mubeen[2*], and Gohar Mumtaz[1]

[1]Faculty of Computer Science and Information Technology, Superior University, Lahore, 54000, Pakistan.
[2]Riphah International University, Lahore, 54000, Pakistan.
[*]Corresponding Author: Zeeshan Mubeen. Email: zeeshan.mubeen@riphah.edu.pk

Received: February 19, 2024 **Accepted:** May 21, 2024 **Published:** June 01, 2024

Abstract: Since attacks on information resources are common and come from both domestic and foreign sources, it is critical to ensure their security, particularly that of the network infrastructure that provides internet access. The identification of anomalies in these networks is mostly dependent on anomaly detection systems, or IDSs. However, the algorithms that IDSs use and their ability to learn are largely responsible for their efficacy. Given the complexity of malicious activities, it's critical to use techniques that offer maximum effectiveness and superior performance. The aim of this work is to assess how well boosting algorithms—more especially, LightGBM and CatBoost—identify fraudulent network traffic. The CICID2017 dataset was used in the study to apply LightGBM and CatBoost using Google Colab. Performance criteria including recall, accuracy, precision, and F1-score were employed to evaluate the classifiers. The analysis showed that CatBoost performed better than LightGBM models, with an astounding f1-score of 99.89%. On the other hand, with little data, the LightGBM model demonstrated reduced efficacy in detecting attack types. This study emphasizes how important it is to use efficient methods, like CatBoost, to boost anomaly detection systems' efficiency and strengthen information resource security against hostile activity on network infrastructures.

Keywords: Comparative Analysis; Cyber Security; Network Traffic Scenarios; Imbalanced Class Distributions; CIC-IDS 2017.

1. Introduction

One of the sectors with the quickest growth is cybersecurity, driven by the necessity to develop new tools capable of detecting, preventing, and responding to various types of attacks. Analyzing time-related risks associated with network traffic is a fundamental aspect of developing these tools. However, the complexity of network traffic poses a significant challenge in providing accurate and effective solutions. Additionally, as information regarding existing attacks, vulnerabilities, and security measures improves, attacks become increasingly sophisticated. Thus, to safeguard crucial network infrastructures, methods like signature detection systems and anomaly detection systems must be used in conjunction with deep learning or machine learning-based algorithms.

In the past few years, there has been a massive increase in the quantity of machine learning and deep learning applications. For example, Kanimozhi and Jacob [2] achieved a 99.97% accuracy in classifying bot attacks using an artificial intelligence-based IDS. Saranya et al. [3] in their study, provide a thorough performance analysis of machine learning methods used with IDSs. Preliminary investigations in the literature have used well-known datasets like Kyoto2006+, CAIDA-2007, NSL-KDD, DARPA, KDD-Cup'99, and TU-DDoS to develop real-time IDSs. However, the CICIDS2017 dataset has recently gained considerable attention from academics due to its comprehensive coverage of contemporary network threats. In this study, we selected the CICIDS2017 dataset to apply boosting techniques, one of the primary goals of our research is to create a novel IDS.

Research utilizing the CICIDS2017 dataset for network attack detection and classification has employed various machine learning algorithms. Sharafaldin et al. [5] found that the ID3 algorithm yielded

an F1 score of 0.98, which was the best performance. Similarly, Özekes and Karakoç [6] presented ways using random forests and decision trees with accuracy scores. Tama et al. [7] utilized Random Forest, Gradient Boosting Machine, and XGBoost algorithms, achieving 99.98% accuracy with a stacked ensemble approach. Abdulrahman and Ibrahem compared the performance of classifiers, with Random Forest and C5.0 outperforming others. Hosseini and Seilani sought to increase system precision while cutting down on training time. utilizing the NSL-KDD and CICIDS2017 datasets, they achieved over 99% accuracy with a runtime of 27.36 seconds utilizing the K-Nearest Neighbors, Random Forest, Decision Tree, and Logistic Regression methods.

2. Ensemble Learning and Boosting Algorithms:

Multiple training subsets are created by using Ensemble Learning (EL) techniques, which modify the distribution of training datasets. Several base classifiers are then developed to predict unknown data by voting on their outcomes. EL combines different classifiers using techniques such as voting, stacking, bagging, and boosting. Voting aggregates predictions from multiple regression models to produce a forecast, In contrast, stacking entails training many machine learning models as base learners, which are then combined to create a final prediction via a meta-classifier. Bagging, in contrast, involves combining multiple learners by modifying the training set for each one, thereby improving the overall accuracy of the model. Boosting algorithms, a sequential technique, aim to enhance learning by focusing on previous errors and learning from them. This approach attains high performance by iteratively presenting specific portions of the training data to the learning algorithm. Unlike bagging, where multiple learners adapt to diverse samples, boosting creates random samples from the training dataset first. A weak classifier is created for each sample and tested separately across the entire training dataset. If a sample is misclassified, its weight is increased, and a new sample is produced. The iterations persist until the system achieves high accuracy. Boosting generally exhibits a lower error rate compared to bagging. Moreover, boosting significantly improves classification rates in datasets where decision trees perform well in classification.

In this study, LightGBM, and CatBoost techniques were employed. The boosting technique's effectiveness in identifying fraudulent network traffic was examined using the CICIDS2017 dataset. The dataset underwent recursive feature elimination (RFE) following normalization, which removed 25 features and left 53. To evaluate the performance of the classifier, metrics like recall, accuracy, precision, and F1-score were used. The research revealed that although the CatBoost approach proved to be the most efficient, the LightGBM model showed less prowess in pinpointing uncommon attack types within the dataset, demonstrating the worst overall performance in contrast to other models.

This is how the study's subsequent sections are organized: Section 2 provides an explanation of Ensemble Learning and boosting methods, expounding on the boosting strategies applied in the study. The steps involved in preparing the dataset, examined in Section 3, providing context by utilizing the CICIDS2017 dataset as an example. Section 4 discusses network intrusion classification using boosting techniques, including model assessment criteria, experimental results, findings, and model comparisons. Finally, Section 5 concludes the research.

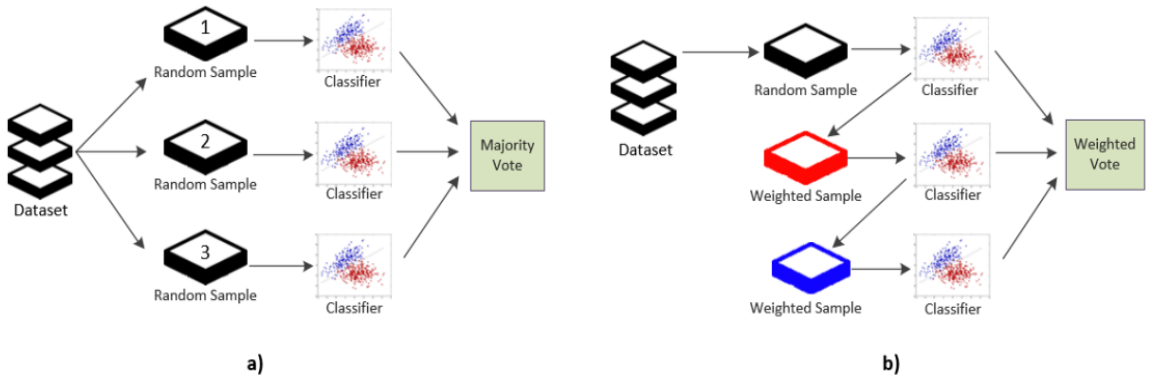

Figure 1. a)Bagging b)Boosting

2.1. Light GBM

Compared to other boosting techniques, LightGBM is a machine-learning technology that produces faster and more accurate outcomes. Gradient boosting is used to solve regression, classification, and ranking problems. Two cutting-edge methods are offered by LightGBM to increase speed and efficiency: gradient-based one-side sampling (GOSS) and exclusive feature bundling (EFB). GOSS prioritizes data points with larger gradients, which speeds up the process of determining the optimal split point. Through the combination of mutually incompatible qualities, EFB minimizes the number of features while increasing algorithm performance. These improvements have made LightGBM 20 times quicker than traditional gradient-boosting decision trees, and as a result, it is becoming more and more popular in machine learning for tasks like regression and classification. Figure 2 displays a visualization of the light GBM model.

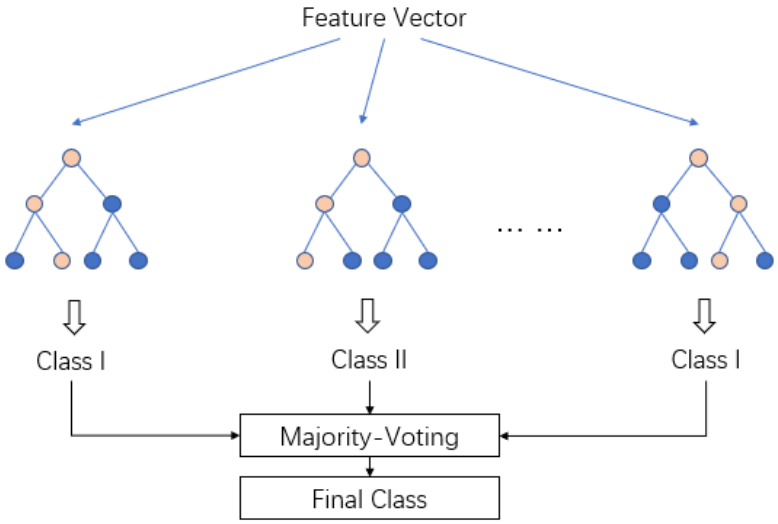

Figure 2. Basic Structure of LightGBM model

2.2. Catboost:

One component of CatBoost, an open-source machine learning tool used for classification, regression, and ranking, is gradient boosting. Dorogush et al. created CatBoost to address the challenge of managing categorical features during data preparation. It makes use of encoding techniques to convert categorical qualities into numerical ones. Datasets with both numerical and category information can be handled by CatBoost with ease. CatBoost employs a technique that lessens overfitting to handle categorical variables. This strategy involves randomly permuting or shuffling features. For each example, CatBoost calculates the average label value for examples in the shuffled list that have the same category value as the provided example and are positioned before it. Every case is substituted with the computed average label value during model training. CatBoost employs a collective strategy, wherein permutation is performed again on the dataset when generating a new decision tree, and the procedure for figuring out a category example's numerical value is repeated. Additionally, CatBoost employs a useful technique for transforming categorical characteristics into numerical ones by calculating the number of occurrences in the dataset and replacing the count values with the training examples. CatBoost also presents feature combination, which is the process of combining features from the dataset to create more potent features. To manage computation overhead, CatBoost combines features in a greedy manner, Steer clear of pairings that lead to the tree's initial split. CatBoost mixes every possible combination of category features in the current tree with every category feature in the dataset for splits that come after. Another improvement in CatBoost is its approach to fighting gradient bias. Unlike XGBoost and LightGBM, CatBoost builds the tree structure using a modified version of gradient-based decision trees and sets the leaf values of the constructed tree using conventional gradient-boosting decision trees. According to Dorogush et al., Due to these characteristics, CatBoost performs more accurately and quickly computationally than other cutting-edge libraries like XGBoost and LightGBM.

3. Dataset and Preprocessing

3.1. Dataset

The Canadian Cyber Security Institute assisted in creating the CICIDS2017 dataset, diverges from preceding datasets utilized in literature across various significant facets. In contrast to previous datasets, the victim network in CICIDS2017 encompasses all fundamental infrastructure elements, including routers, firewalls, switches, as well as several Macintosh, Linux, and Windows operating system versions.

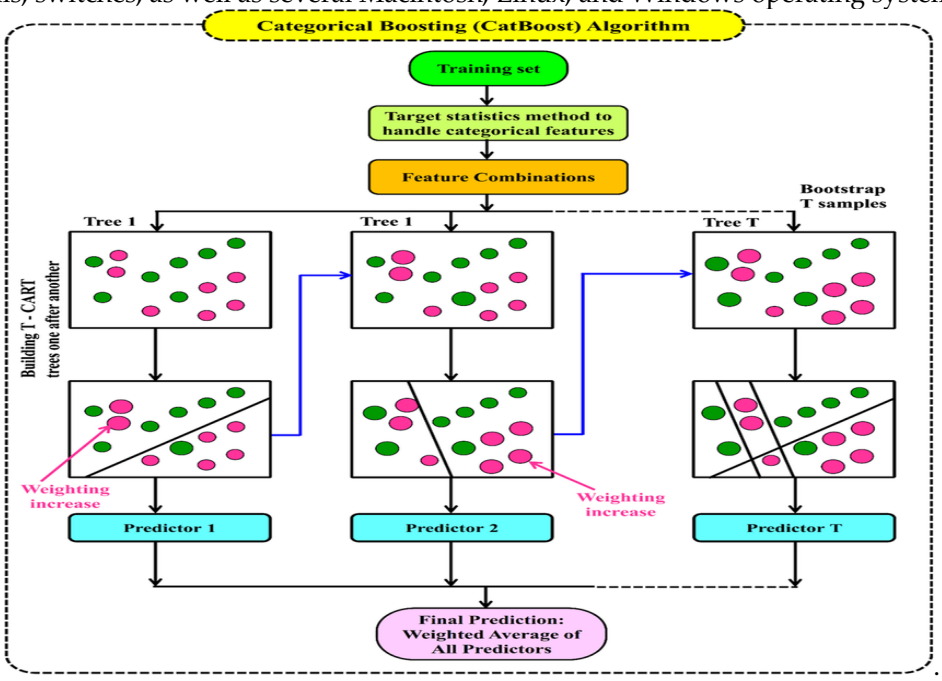

Figure 3. Basic Structure of CatBoost model.

The assault network is made up of a router, a switch, and four computers running Windows 8.1 and Kali. Figure 4 provides an overview of the varieties of cyberattacks occurring daily in the CICIDS2017 dataset [36].

Days	Labels
Monday	BENING
Tuesday	BForce, SFTP and SSH
Wednesday	DoS and Heartbleed Attacks, slowloris, Slowhttptest, Hulk and GoldenEye
Thursday	Web and Infiltration Attacks, Web BForce, XSS and SQL Inject. Infiltration Dropbox Download and Cool disk
Friday	DDoS LOIT, BotNet ARES, PortScans (sS, ST, sF, sX, sN, sP, sV, sU, SO, sA, sW, sR, SL and B)

Figure 4. Daily Label Data Set

Figure 5 illustrates the distribution of attack types documented in the CICID2017 dataset, along with their corresponding percentages relative to the entire dataset. Two, 273,097 records, or 80% of the entire data, are benign (BENIGN) packet data [37] [38].

3.2. Data Preprocessing

The feature selection, normalization, and data preprocessing methods are described in this section. That carried out on the CICIDS2017 dataset prior to employing boosting algorithms for classification. Initially, missing, erroneous, and corrupt data within the dataset were identified, resulting in the extraction of a total of 1358 data points. Subsequently. The attack class data were encoded using the Label Encoding technique, resulting in values within the range of 0 to 14. Standardizing and normalizing the dataset is essential to optimize model accuracy for machine learning algorithms. Effective distribution and organization of data significantly improve the performance of machine learning algorithms hence, the

refined and rectified data underwent Z-Score Scaling. Following normalization, feature selection was applied to the CICIDS2017 dataset [39] [40]. In this study, the Recursive Feature Elimination (RFE) method was utilized to choose features, entailing the elimination of non-contributing features to gradually delineate distinct classes from all available features.

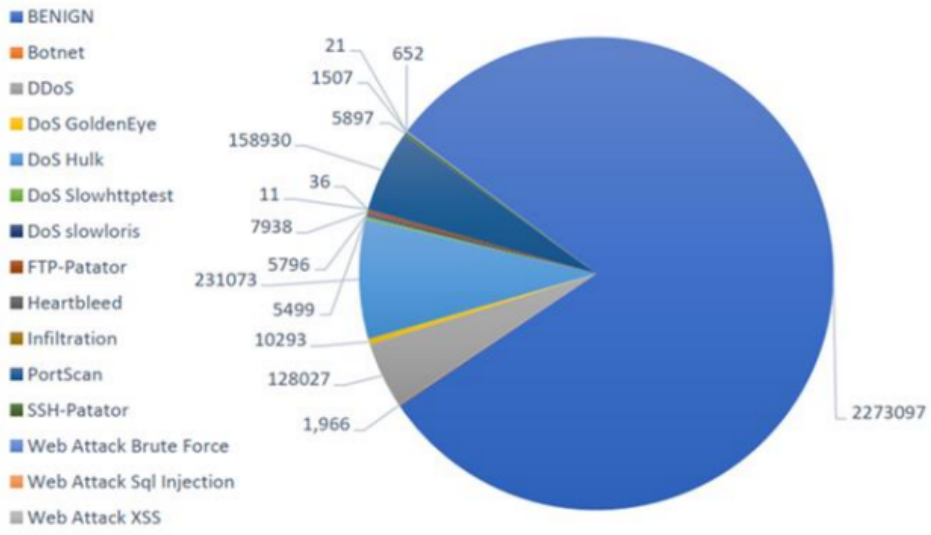

Figure 5. CICIDS2017 dataset assault class and BENIGN distribution

Feature importance is assessed by employing a classification algorithm on the features, and iteratively eliminating those with lesser discriminatory power until only those with the highest discriminatory power remain. For this study, a threshold importance value of 0.005 was selected, determined through the assessment of the employed algorithms and their respective performance metrics. Consequently, twenty-five characteristics whose significance values fell below this cutoff were eliminated from the dataset, leaving a total of 53 features.

Figure 6 depicts the significance values of attributes within the CICIDS2017 dataset. Notably, attributes Certain properties, such as the destination port and packet length, are considered crucial for detecting assaults.

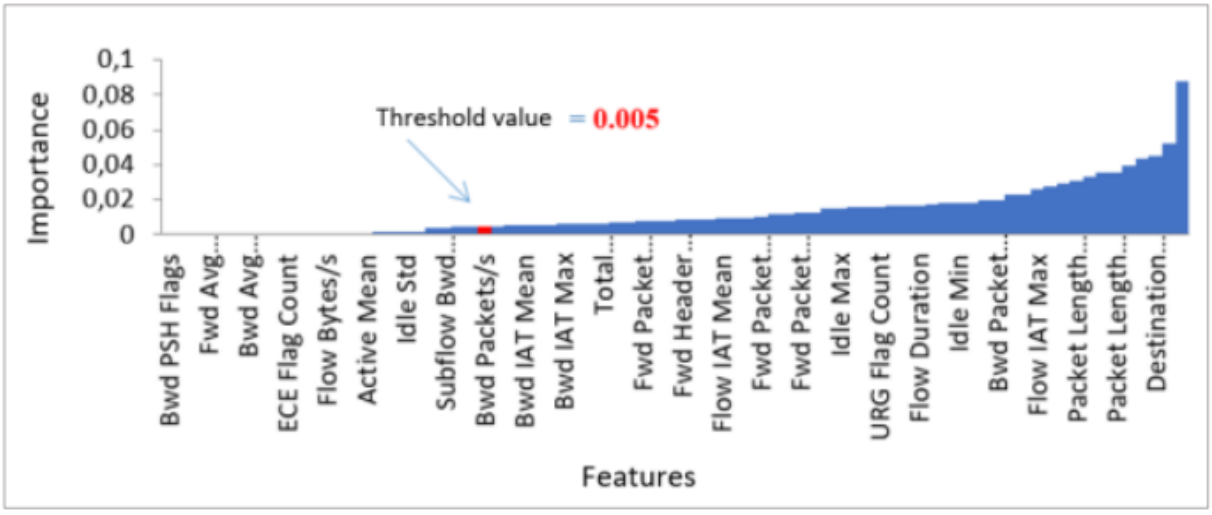

Figure 6. The CICIDS 2017 dataset's feature importance using the RFE approach.

4. Classification with Boosting Algorithm

4.1. Modeling Datasets for Boosting Algorithms and Assessment Measures

The Jupyter Notebook virtual server and the Python programming language were used in the Colab platform analysis. The CICIDS2017 dataset was divided into two halves in order to apply machine learning

techniques: 70% was put aside for training, and 30% was set aside for testing. During the construction of the training and test datasets, data were randomly selected from the dataset, ensuring homogenous acquisitions based on attack types. Proportional additions from attack categories and regular traffic data were made to both the training and test datasets. For example, 30% of the sluggish loris packages were assigned to the test set and 70% to the training set; similar allocations were made for other kinds of attacks. Additionally, the accuracy of the study was improved by using cross-validation with k-fold 10.

In the study, fundamental parameters including Both models, Catboost and LightGBM used a maximum depth of 50, a learning rate of 0.1, and n_estimators of 100. Following the training procedure, Metrics like accuracy, precision, recall, and f1-score were used to assess the boosting algorithms' performance.

Accuracy, as defined by Equation (1), represents Accuracy is represented by dividing the total number of observations by the proportion of outcomes that are genuine positives (TP) to true negatives (TN). Precision, conversely, is the ratio of observations correctly identified as positive (TP) to all observations classified as positive, as outlined in Equation (2). The proportion of observations with a genuine positive value (TP) to those that were mistakenly classified as negative (False Negative - FN) is known as recall/sensitivity, divided by the sum of observations with a true positive value and those incorrectly classified as negative. This is expressed in Equation (3). The f1-score, which is the precision and recall values' harmonic average is defined by Equation (4). The model's performance is shown by the f1-score value, which ranges from 0 to 1. A value closer to 1 denotes better performance.

Accuracy= (TP+TN)/ (TP+TN+FP+FN) (1)

Sensitivity=TP/ (TP+FN) (2)

Precision=TP/ (TP+FP) (3)

F1-Score=2TP/ (2TP+FP+FN) (4)

4.2. Experimental Results and Discussion

The overall average values of performance metrics for the models used in the CICID2017 dataset's detection and classification of network threats are shown in Table 2. The accuracy, precision and F1-score of the CatBoost model were highest, at 0.9989, 0.9942, and 0.8945, respectively.

Table 1. Evaluation metrics comparison.

Model	Accuracy	Percision	Recall	F1-Score
LightGBM	0.9773	0.4653	0.5126	0.4817
CatBoost	0.9989	0.9942	0.8745	0.8937

The confusion matrices depicted in Figure 5(a)-(b) illustrate the accuracy of class predictions and identify misclassified classes. In both models, certain data belonging to the attack categories labeled as 'Bot,' 'Infiltration,' and 'Web Attack,' for instance, are misclassified. Regarding the 'Infiltration' attack, the LightGBM The data was mislabeled as regular traffic by the model, which was unable to identify it effectively.

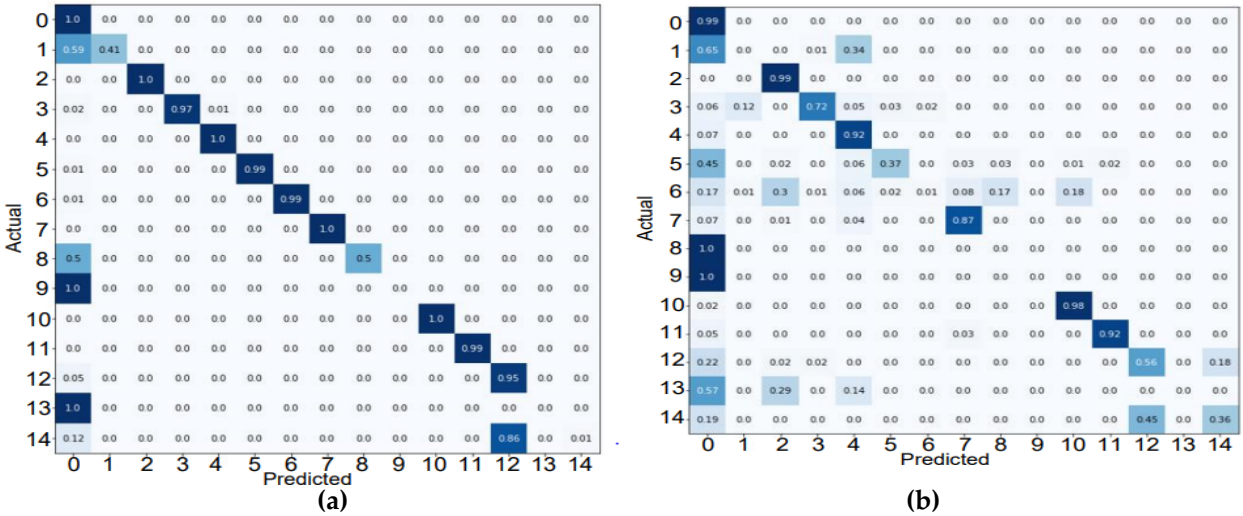

Figure 6. a) LightGBM b) CatBoost

5. Conclusion

This research employed LightGBM and CatBoost to detect malicious network traffic using the CICIDS2017 dataset. These classifiers' performance was assessed by employing measures such as f1-score, recall, accuracy, and precision. The CatBoost model showed the greatest results in terms of recall, accuracy, precision, and f1-score criterion. However, the model LightGBM showed the lowest performance rate. It faced challenges in making precise predictions, particularly for the assaults in the dataset known as The LightGBM variant, however, had the lowest performance rate. This limitation can be attributed to the These Certain network attack types are uncommon within the sample, indicating that unbalanced datasets reduce the LightGBM model's efficacy.

Moreover, all models consistently misclassified 'Bot,' 'Infiltration,' and 'Web Attack' attacks. This study suggests a technique for identifying network assaults in practical systems and putting the appropriate safeguards in place. It emphasizes the importance of identifying the type of attack to effectively counter it. As technology advances, the diversity of attacks is expected to increase, making their identification more challenging. Indeed, the swift advancement of CPU and GPU capabilities opens avenues for employing algorithms for machine learning and deep learning on powerful computers. This facilitates the identification of malicious network traffic with heightened accuracy and speed.

Data Availability: The dataset used in this study is the Canadian Institute for Cybersecurity Intrusion Detection System 2017 (CICIDS2017) dataset. The official website of the Canadian Institute for Cybersecurity provides access to this publicly available dataset.

Conflicts of Interest: We explicitly stated that there are no conflicts of interest in this research. The work reported in this study could not have been influenced in any way by any personal, financial, or other contacts. The authors have approved the final text of the manuscript and state that they have no conflicts of interest to report.

References

1. Perez, S.I., Moral-Rubio, S., Criado, R., "A new approach to combine multiplex networks and time series attributes: Building intrusion detection systems (IDS) in cybersecurity", Chaos, Solutions and Fractals, Vol. 150, Pages 1-11, 2021.
2. Kanimozhi, V. and Jacob, T.P, "Artificial Intelligence based Network Intrusion Detection with hyper-parameter optimization tuning on the realistic cyber dataset CSE-CIC-IDS2018 using cloud computing", ICT Express, Vol. 5, Issue 3, Pages 211-214, 2019.
3. Saranya, T., Sridevi, S., Deisy, C., Chung, T.D., Ahamed, K.M., "Performance Analysis of Machine Learning Algorithms in Intrusion Detection System: A Review", Third IC on Computing and Network Communications (CoCoNet'19), Trivandrum, 2020.
4. Ghurab, M., Gaphari, G., Alshami, F., Alshamy, R., Othman, S., "A Detailed Analysis of Benchmark Datasets for Network Intrusion Detection System" Asian Journal of Research in Computer Science, Vol. 7, Issue 4, Pages 14-33, 2021.
5. Sharafaldin, I., Lashkari, A., Ghorbani, A., "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization", 4th International Conference on Information Systems Security and Privacy, Portugal, 2018.
6. Özekes, S. and Karakoç, E.N., "Makine Öğrenmesi Yöntemleriyle Anormal Ağ Trafiğinin Tespit Edilmesi", Düzce Üniversitesi Bilim ve Teknoloji Dergisi, Vol. 7, Issue 1, Pages 566-576, 2019.
7. Tama, B.A., Nkenyereye, L., Islam, S.R., Kwak, K.S., "An Enhanced Anomaly Detection in Web Traffic Using a Stack of Classifier Ensemble", IEEE Access, Vol. 8, Pages 24120 – 24134, 2020
8. Abdulrahman, A.A. and Ibrahem, M.K., "Toward Constructing a Balanced Intrusion Detection Dataset Based on CICIDS2017", Samarra Journal of Pure and Applied Science, Vol. 2, Issue 3, Pages 132-142, 2020.
9. Hosseini, S. and Seilani, H., "Anomaly process detection using negative selection algorithm and classification techniques", Evolving Systems, Vol. 12, Pages 769–778, 2021.
10. Hongle, D., Yan, Z., Gang, K., Lin, Z., Chen, Y.C., "Online ensemble learning algorithm for imbalanced data stream", Applied Soft Computing, Vol. 107, Pages 1-12, 2021.
11. Schapire, R.E., "The Boosting Approach to Machine Learning an Overview", In: Denison DD, Hansen MH, Holmes CC et al editors, Nonlinear Estimation and Classification. Lecture Notes in Statistics, Vol. 171, Springer, New York, Pages 1- 23, 2003.
12. Pham, X.T. and Ho, T.H., "Using boosting algorithms to predict bank failure: An untold story", International Review of Economics & Finance, Vol. 76, Pages 40-54, 2021.
13. Shahraki, A., Abbasi, M., Haugen, Q., "Boosting algorithms for network intrusion detection: A comparative evaluation of Real AdaBoost, Gentle AdaBoost and Modest AdaBoost", Engineering Applications of Artificial Intelligence, Vol. 94, Pages 1-14, 2020.
14. Li, Y., Shi, H., Duan, Z., Liu, H., "Smart wind speed forecasting approach using various boosting algorithms, big multi-step forecasting strategy", Renewable Energy, Vol. 135, Pages 540-553, 2019.
15. Ma, B., Meng, F., Yan, G., Yan, H., Chai, B., Song, F., "Diagnostic classification of cancers using extreme gradient boosting algorithm and multiomics data", Computers in Biology and Medicine, Vol. 121, Pages 1-10, 2020.
16. Abro, A.A, Taşcı, E., Uğur, A.A., "Stackingbased Ensemble Learning Method for Outlier Detection", Balkan Journal of Electrical & Computer Engineering, Vol. 8, Issue 2, Pages 191- 185, 2020.
17. Wen, L., Hughes, M., "Coastal Wetland Mapping Using Ensemble Learning Algorithms: A Comparative Study of Bagging, Boosting and Stacking Techniques", Remote Sensing, Vol. 12, Issue 10, Pages 1-18, 2020.
18. Xia, T., Zhuo, P., Xiao, L., Du, S., Wang, D., Lifeng, X. "Multi-stage fault diagnosis framework for rolling bearing based on OHF Elman AdaBoostBagging algorithm", Neurocomputing, Vol. 433, Pages 237-251, 2021.

19. Andiojaya, A. and Demirhan, H., "A bagging algorithm for the imputation of missing values in time series", Expert Systems with Applications, Vol. 129, Pages 10-26, 2019.
20. Yin, S., Liu, H., Duan, Z., "Hourly PM2.5 concentrations multi-step forecasting method based on extreme learning machine, boosting algorithm and error correction model", Digital Signal Processing, Vol. 118, Pages 1-21, 2021.
21. Freund, Y. and Schapire, R.E., "A decisiontheoretic generalization of on- line learning and an application to boosting", Journal of Computer and System Sciences, Vol. 55, Issue 1, Pages 119-139, 1997.
22. Chengsheng, T., Huacheng, L., Xu, B., "AdaBoost typical Algorithm and its application research", MATEC Web of Conferences, Vol. 139, Issue 2, France, 2017.
23. Qi, C., Wang, Y., Tian, W., Wang, Q., "Multiple kernel boosting framework based on information measure for classification", Chaos, Solutions and Fractals, Vol. 89, Pages 175-186, 2016.
24. Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A.V., Gulin, A., "CatBoost: unbiased boosting with categorical features", NeurIPS - 32nd Conference on Neural Information Processing Systems, Montreal, 2018.
25. Friedman J.H., "Greedy function approximation: a gradient boosting machine", Annals of statistics, Vol. 29, Issue 5, Page s1189-1232, 2001.
26. Kearns, M. and Valiant, L., "Cryptographic limitations on learning Boolean formulae and finite automata", Journal of the ACM, Vol. 41, Issue 1, Pages 67-95, 1994.
27. Friedman, J.H. "Stochastic gradient boosting", Computational Statistics & Data Analysis, Vol. 38, Issue 4, Page 367-378, 2002.
28. Dahiya, N., Saini, B., Chalak, H.D., "Gradient boosting-based regression modelling for estimating the time period of the irregular precast concrete structural system with cross bracing", Journal of King Saud University - Engineering Sciences, Pages 1-8, 2021.
29. Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., Liu, T.Y., "LightGBM: a highly efficient gradient boosting decision tree", NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing Systems, Curran Associates Inc. California, 2017.
30. Shehadeh, A., Alshboul, O., Al Mamlook, R.E., Hamedat, O., "Machine learning models for predicting the residual value of heavy construction equipment: An evaluation of modified decision tree, LightGBM, and XGBoost regression", Automation in Construction, Vol. 129, Pages 1-16, 2021.
31. Chen, T. and Guestrin, C., "XGboost: A scalable tree boosting system", 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Pages 785–794, San Francisco, 2016.
32. Ma, J., Zhongqi, Y., Qu, Y., Xu, J., Cao, Y., "Application of the XGBoost Machine Learning Method in PM2.5 Prediction: A Case Study of Shanghai", Aerosol and Air Quality Research, Vol. 20, Issue 1, Pages 128-138, 2019.
33. Sharma, N.V. and Yadav, N.S., "An optimal intrusion detection system using recursive feature elimination and ensemble of classifiers", Microprocessors and Microsystems, Vol. 85, Pages 1-11, 2021.
34. Aksoy, B., Usta, U., Karadağ, G., Kaya, A.R., Ömür, M., "Classification of Environmental Sounds with Deep Learning", Advances in Artificial Intelligence Research, Vol. 2, Issue 1, Pages 20-28, 2022.
35. Aksoy, B. and Salman, O.K.M., "Detection of COVID-19 Disease in Chest X-Ray Images with capsul networks: application with cloud computing", Journal of Experimental & Theoretical Artificial Intelligence, Vol. 33, Issue 3, Pages 527-541, 202
36. Batool, S., Abid, M. K., Salahuddin, M. A., Aziz, Y., Naeem, A., & Aslam, N. (2024). Integrating IoT and Machine Learning to Provide Intelligent Security in Smart Homes. Journal of Computing & Biomedical Informatics, 7(01), 224-238.
37. Abbas, M., Arslan, M., Bhatty, R. A., Yousaf, F., Khan, A. A., & Rafay, A. (2024). Enhanced Skin Disease Diagnosis through Convolutional Neural Networks and Data Augmentation Techniques. Journal of Computing & Biomedical Informatics, 7(01).

38. Muhammad Kaleem , Muhammad Azhar Mushtaq , Uzair Jamil , Sadaqat Ali Ramay , Tahir Abbas Khan , Siraj Patel , Rizwan Zahidy , Sayyid Kamran Hussain. (2024). New Efficient Cryptographic Techniques For Cloud Computing Security. Migration Letters, 21(S11), 13–28. Retrieved from https://migrationletters.com
39. Hussain, S.K., Ramay, S.A., Shaheer, H., Abbas T., Mushtaq M.A., Paracha, S., & Saeed, N. (2024). Automated Classification of Ophthalmic Disorders Using Color Fundus Images, Volume: 12, No: 4, pp. 1344-1348 DOI:10.53555/ks.v12i4.3153
40. Ammar Ahmad Khan , Muhammad Arslan , Ashir Tanzil , Rizwan Abid Bhatty , Muhammad Asad Ullah Khalid , Ali Haider Khan. (2024). Classification Of Colon Cancer Using Deep Learning Techniques On Histopathological Images. Migration Letters, 21(S11), 449–463