INTRUSION DETECTION IN A STAND-ALONE 5G NETWORK USING MACHINE LEARNING EVALUATION

Hafiz Bilal Ahmad[1] Haichang Gao[2], and Fawwad Hassan Jaskani[3]

[1, 2] Department of Computer Science and Technology, Xidian University, Xi'an, China
[3] Department of Computer Engineering, Islamia University, Bahawalpur, Pakistan

ABSTRACT

In order to meet the specific requirements of various industries and the stringent demands of 5G, the control and management of 5G networks will heavily depend on the integration of Software Defined Networking, Network Function Virtualization, and Machine Learning. Machine learning can play a crucial role in addressing challenges such as slice type prediction, route optimization, and resource management. To effectively evaluate the use of machine learning in 5G networks, a suitable testing environment is necessary. This study proposes a lightweight testbed that leverages container virtualization technologies to support the development of machine learning net-work functions within 5G networks. The Deep Slice 5G dataset from Kaggle was utilized to predict the type of communication between users based on packet loss and delay budget ratio, with the goal of making 5G systems more efficient. To accomplish this, we applied several Boosted Machine Learning models such as XGBoost, Gradient Boost, AdaBoost, LightGradientBoosting, CatBoost, and HistGradientBoosting. After evaluation, the Catboost model demonstrated the highest accuracy of 99% in identifying the correct slice of 5G based on the selected features of the dataset.

KEYWORDS

5G Network, Machine Learning, Intrusion Detection System, Slice Type Prediction.

1. INTRODUCTION

The Internet of Things (IoT) and mission-critical communications applications are only two examples of the kinds of use cases that will place new demands on the capabilities of 5G networks [1]. Efficient, intelligent, and agile network administration is essential in the face of the increased problems posed by these needs and use cases. In addition, 5G will foster an environment that encourages the development of cutting-edge software in a wide variety of sectors, including the industrial, medical, media, financial, public safety, transportation, agricultural, dietary, and municipal sectors [2].Latency, throughput, availability, dependability, coverage, mobility, and so on are only some of the metrics that must be met [3]. 5G will offer a versatile network that can meet these wide-ranging needs. More software, virtualization, and automation in the network is required for greater portability [4]. Software Defined Networking (SDN) and Network Function Virtualization (NFV) are often viewed as the realization of the softwarization notion and the virtualization paradigm, respectively, from a networking perspective [2].

David C. Wyld et al. (Eds): ICCSEA, NLAI, SCAI, CSIA, IBCOM, SEMIT, NECO, SPPR, MLDS -2023
pp. 223 - 236, 2023. CS & IT - CSCP 2023 DOI: 10.5121/csit.2023.132417

Network slicing is a crucial part of allowing network flexibility because it allows us to build specialized logical networks on top of a shared physical infrastructure to more effectively meet the unique requirements of each individual business. The network functions and supporting infrastructure that make up a network slice. Network slices can be implemented with the help of SDN and NFV because of their programmability, flexibility, and modularity [5].

Software Defined Networks (SDN) and Network Function Virtualization (NFV) give the network more adaptability and configurability by enabling network services to run in software rather than being hardwired into the system [6]. With this flexibility, network operations can be moved, upgraded, and installed at any node. However, manual provisioning, maintenance, and control of network slices is impractical due to the dynamic behaviour of network operations [7]. Due to the ever-changing nature of the environment, network analytics and constant monitoring are now necessities for gaining insight into how networks function. In a similar vein, the provision of automation capabilities to the network is crucial for network operation and management. By eliminating the potential for human mistake and accelerating the time it takes to bring a service to market, automation in the network helps to keep operational costs down [8]. In addition, when Machine Learning (ML) is applied to network analytics, the network gains the ability to learn and make decisions for itself. In order to govern and maintain networks autonomously and provide services, ML approaches can extract useful information from the data collected by the networks [9]–[11]. Allocating the necessary amount of network resources without overprovisioning, ML methods may predict network behavior based on historical and real-time data and adapt to the changing network conditions [12], [13]. ML can also be used to optimize for energy efficiency. To save money on energy expenditures, it may be able to turn off unused components or move services to areas with lower demand. ML has the potential to be successfully employed in automatically orchestrating and managing networks, which would pave the way for self-organizing networks. To rephrase, ML serves as a critical enabler of automation and helps solve the issue of delivering network intelligence. In this light, SDN, NFV, and ML all play important roles in facilitating the deployment of 5G networks [14].

As a result, organizations like the 3rd Generation Partnership Project (3GPP), European Telecommunications Standards Institute (ETSI), and International Telecommunication Union Telecommunication Standardization Sector are all working to ensure that AI and ML are properly represented in 5G and B5G mobile networks (ITU-T). For instance, the ITU-Focus T's Group on Machine Learning for Future Networks, Including 5G (FG-ML5G) designed an ML architecture to be incorporated into future mobile networks, standardizing the terminology used to describe ML-related mechanisms while maintaining compatibility with existing network infrastructure [15]. As has been widely explored in the context of other communications paradigms, such as mesh networks [2]–[4], testbeds provide a viable alternative to simulators for evaluating and integrating the newly created AI algorithms. Since testbeds offer more realistic simulation settings, and solutions built on testbeds have a quicker path from the research stage to products, they are crucial for creating and evaluating network technologies [16]. Due to the complex nature of 5G and B5G networks, simulations often fail to capture key details. This makes testbeds increasingly crucial.

This article details the machine learning-based testbed architecture necessary to rapidly establish realistic 5G scenarios with varying degrees of network slicing based on predetermined budgets for packet loss and latency. The main objectives of this study are:

We have collected the Deep Slice 5G dataset from Kaggle in order to make 5G systems lighter by predicting the sort of communication that will occur between users based on the ratio of packet loss to delay budget. This will allow us to reduce the amount of time that data will be lost.

Boosted models of machine learning, such as XGBoost, Gradient Boosting, AdaBoost, LightGradientBoosting, CatBoost, and HistGradientBoosting, have been utilized on our end for the purpose of making predictions regarding the slices.

2. METHODOLOGY

In this study we have collected the Deep Slice 5G dataset from Kaggle in order to make 5G systems more lightweight. This was done in order to forecast the sort of communication that takes place between users based on the amount of packet loss and the delay budget ratio. We have employed boosted models of machine learning, such as XGBoost, Gradient Boost, AdaBoost, LightGradientBoosting, CatBoost, and HistGradientBoosting, for the prediction of slices.

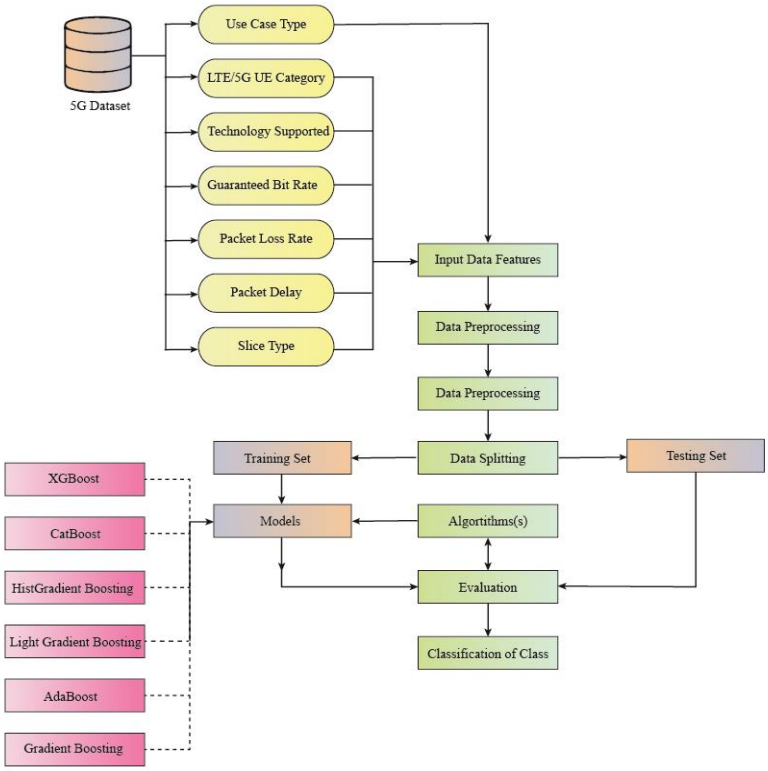

Figure 1. Proposed Model Structure

2.1. Dataset Description

The dataset contains number of features with 6 inputs and 1 output variable. Each feature contains multiple scenarios. List of input and output features has been given in Table 1:

Table 1. Units for Magnetic Properties

Feature	Description	Division	Type
Use Case Type	Real-World 5G Applications: With high capacity and ultra-low latency, 5G will drive AI and IoT applications	Gaming, Healthcare, Industry, IoT Devices, Public Safety, Smart	Input

	across industries and use cases.	City, Home Smart Transportation Smartphone	
LTE/5G UE Category	Existing equipment can be confidently transferred from old 2G/GSM and 3G/UMTS systems to 4G/LTE networks using the LTE Cat M or LTE Cat NB-IoT standards. 5G coverage is currently confined to high-user locations.	1: LTE 2: 5G	Input
Technology Supported	5G uses OFDM to modulate a digital signal across many channels to reduce interference. 5G NR air interface leverages OFDM concepts.	1: LTE 2: 5G 3: IoT	Input
GBR	GBR services with minimum GBR requirements and non-GBR services are supplied in enhanced mobile Broadband usage scenarios.	0: Non GBR 1: GBR	Input
Packet Loss Rate	The packet loss ratio is the percentage of sent packets lost.	0.001 to 0.00001	Input
Packet Delay Budget	The Packet Delay Budget (PDB) limits packet delay between the UE and the N6 termination point at the UPF.	50ms to 1000ms	Input

2.2. Classification Models

We have collected the Deep Slice 5G dataset from Kaggle in order to make 5G systems lighter by predicting the sort of communication that will occur between users based on the ratio of packet loss to delay budget. This will allow us to reduce the amount of time that data will be lost. Boosted models of machine learning, such as XG Boost, Gradient Boost, AdaBoost, Light Gradient Boosting, Cat Boost, and Hist Gradient Boosting, have been utilised on our end for the purpose of making predictions regarding the slices. As a consequence of this, the Cat boosting Model has demonstrated the maximum accuracy, which is 99.8%, in determining the appropriate slice of 5G on the basis of specified aspects of the dataset.

2.2.1. XG Boost

Extreme Gradient Boosting (XG Boost) is a machine learning toolkit that provides a scalable implementation of a gradient-boosted decision tree (GBDT). It is the most popular machine learning library, and it offers parallel tree boosting, which can be applied to issues of regression, classification, and ranking. Let's approximate a function f(x) with the easiest linear approximation we can calculate as:

$$f(x) = f(a) + f'(a)(x - a) \dots (1)$$

X can be considered as change in a prediction function when we have multiple outputs:

$$f(\Delta x) = f_t x_i \dots (2)$$

In this scenario, the loss function l is denoted as $f(x)$, the anticipated value from the previous iteration $(t-1)$ and x is the new learner to be introduced in iteration t. To optimize in Euclidean space, we can use the above at each iteration t to define the objective (loss) function as a simple function of the newly added learner. To recap, in step (t), the prediction from step $(t-1)$ is a,

and in step (t), the new learner we need to add in order to greedily reduce the goal is $(x-a)$. In this case, if we adopt the Taylor approximation of the second order, we obtain:

$$f(\Delta x) = \sum f(a) + f^{\wedge\prime}(a)(x-a) + \llbracket 1/2\ g \rrbracket^{\wedge\prime}(a)(x-a) + f\_t\ x\_i \ldots (3)$$

Where g is the second order gradient used for multi classification.

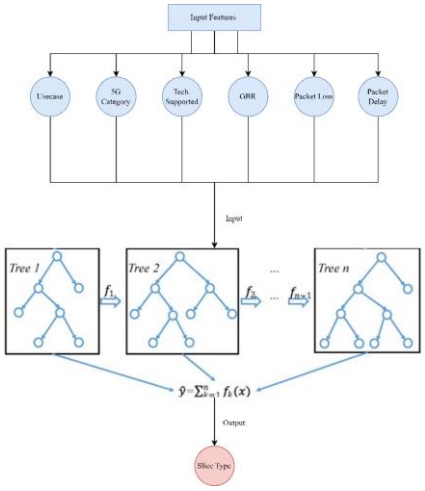

Figure 2. XGBoost Architecture

2.2.2. Cat Boost

Cat Boost can automatically manage features that can be categorized. Despite its popularity, one-hot encoding can no longer be used when there are too many features in the data set. To address this issue, features are classified into groups based on the statistics used to evaluate success (estimate target value for each category). The desired estimate for the ith categorical variable in the kth element of D can be written in mathematical notation as,

In order to reduce prediction time, CatBoost uses oblivious decision trees, which are binary trees in which the same features are utilized to create left and right splits for each level of the tree. It efficiently deals with categorical features by using sorted target statistics.
In the first step we will initialize the model,

$$F_o(x) = argmin_\gamma \sum_{i=1}^{n} L(y,\gamma) \ldots (4)$$

For m = 1 to M, we will compute the residuals.

$$\gamma_{im} = -\left[\frac{\partial L[y,F(x_i)]}{\partial F x_i}\right]_{F(x)=F_{M-1}(x)} \ldots (5)$$

Then we will fit the base leaner to compute it with pseudo residuals:

$$\gamma_{im} = argmin_{\gamma} \sum_{xi}^{n} L(y, F_{M-1}(x)) \dots (6)$$

Updated Model will be:

$$y = F_m(x) = F_{M-1}(x) + \alpha \sum_{i=1}^{n} \gamma_{im} \dots (7)$$

The architecture of CBC Model has been shown in Figure 3:

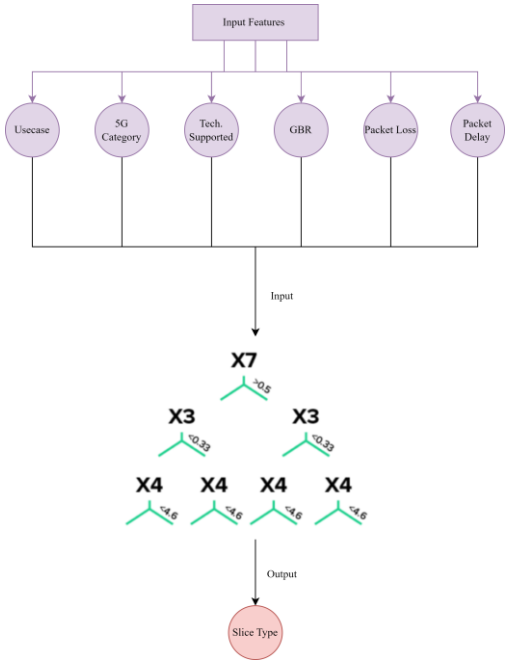

Figure 3. CatBoost Architecture

2.2.3. Gradient Boosting

Specifically, Gradient Boosting is an iterative functional gradient algorithm, which minimizes a loss function by repeatedly selecting a function that tends toward the negative gradient, or a hypothesis with low statistical support. Mathematical model of GBC Classification model is as follows:

$$y = y^i = y^i + \alpha * \frac{\partial \sum (y_i - y_i^p)^2}{\partial y_p^i} \dots (8)$$

The architecture of GBC Model has been shown in Figure 4.

2.2.4. AdaBoost

AdaBoost, which stands for "Adaptive Boosting," is an algorithm that employs the Boosting technique as part of an Ensemble Method for machine learning. Adaptive boosting gets its name from the fact that it re-assigns weights to each instance, giving more weight to examples that were mistakenly classified. This model has been developed by ensembling trees model into

AdaBoost Classifier to improve accuracy. Mathematical model of ABC Classification model is as follows

$$y = significance \sum_{t=1}^{T} \alpha_t h_t(x) \dots (9)$$

The architecture of ABC Model has been shown in Figure 5.

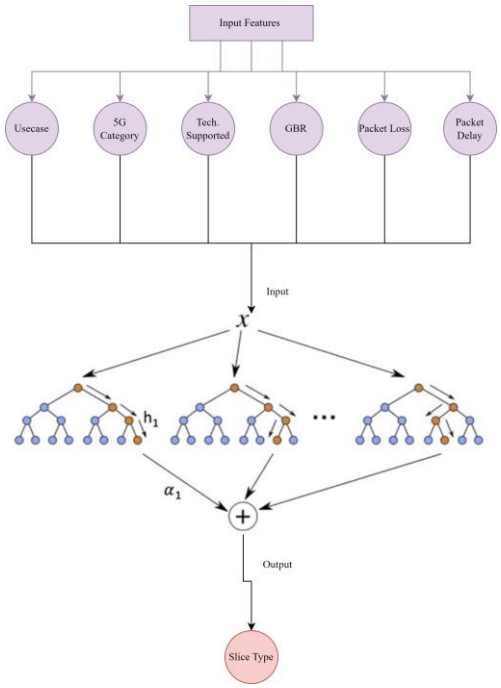

Figure 4. Gradient Boosting Architecture

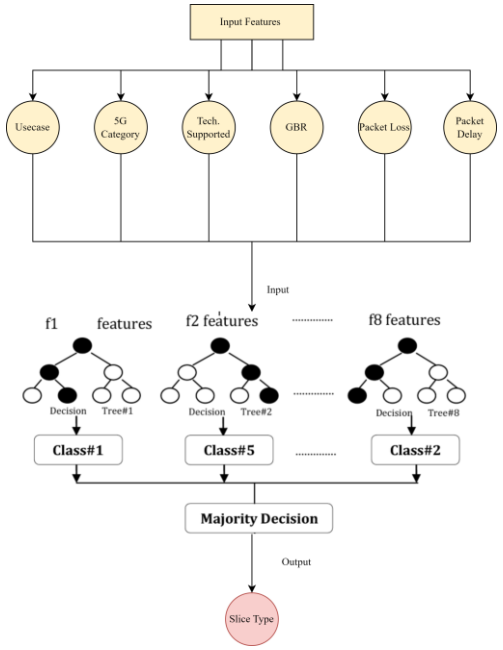

Figure 5. Adaboosting Architecture

2.2.5. Hist Gradient Boosting

A histogram can be used to visualize or tally the frequency of data (number of occurrences) over intervals (bins). The histogram approach is essentially straightforward, with each bin representing the frequency of the corresponding value. Mathematical model of HGBC Classification model is as follows

$$y = \frac{sum\ of\ residuals}{sum\ of\ each\ (1-p) for\ each\ sample\ in\ the\ leaf} \dots (10)$$

The architecture of HGBC Model has been shown in Figure 6.

2.2.6. Light Gradient Boosting

In contrast to traditional boosting algorithms, LightGBM divides the tree at each leaf level as it expands. In order to maximize its delta loss, it selects the leaf that is the most advantageous for growth. The loss of the leaf-wise approach is less than that of the level-wise algorithm because the leaf is always known. Mathematical model of LGBM Classification model is as follows

$$y = \alpha \sum_{t_i \epsilon\, Tree}^{T} \eta^i * leaf\,(t_i) \ldots (11)$$

The architecture of LGBC Model has been shown in Figure 7.

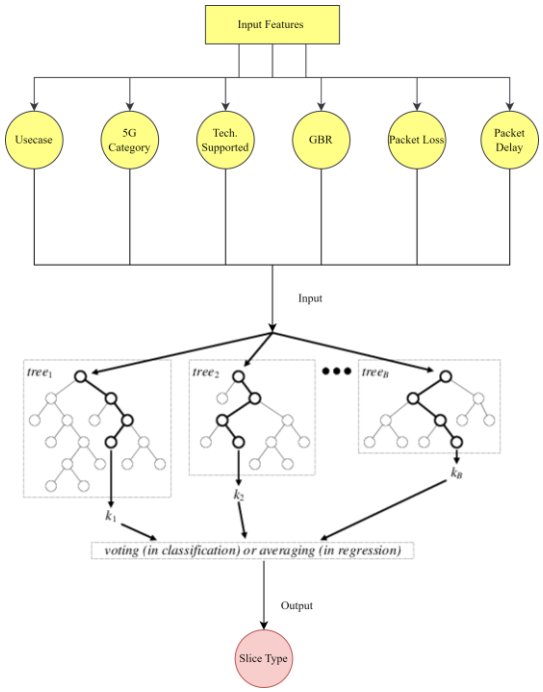

Figure 6. HGBC Architecture

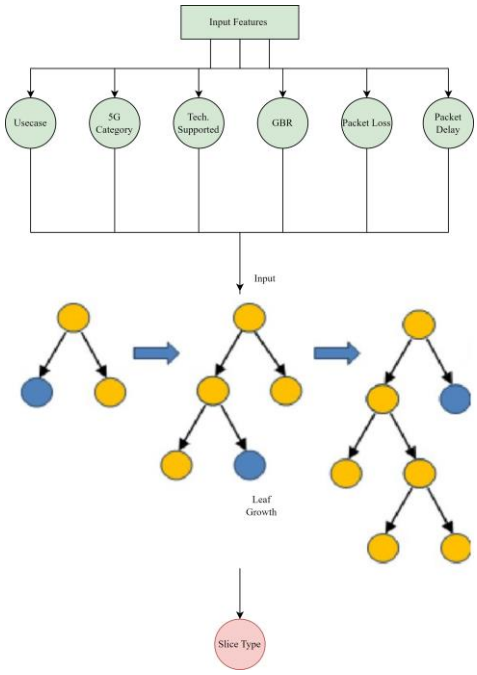

Figure 7. Light Gradient Boosting Architecture

2.3. Performance Metrics

2.3.1. TPR (True Positive Rate):

True positives divided by total positives and false negatives, or TP/TP+FN, is a method for determining a test's sensitivity. Probability of a positive test is known as the TPR (Test Prevalence Ratio). For example, a test's true negative rate (also known as specificity) is the number of times a patient who is genuinely negative will be incorrectly identified as a negative by the test's results.

$$TPR = \frac{TP}{TP + FN}$$

2.3.2. False Positive Rate

FPR, commonly known as the false positive rate, is a measure of how reliable a test is and how many false positives it has. Medical diagnostic tests, machine learning models, or any other kind of test can be used for this. The term "false positive rate" in statistics refers to the probability of rejecting the null hypothesis wrongly.

$$FPR = 1 - spcificity = \frac{FP}{TN + FP}$$

2.3.3. Accuracy

All projected data points are counted to determine Accuracy. True positives and true negatives divided by the total number of true positives, true negatives, false positives, and false negatives is a more formal definition of it. It is calculated as:

$$Accuracy = \frac{TP}{TP + FP}$$

2.3.4. Macro Average/ Weighted Average

Basically, it's just the average of the scores from each class. A macro-average recall is the sum of individual class recollections, A, B, and C combined. Weighted average is the sum of combined classes as A, B and C.

2.3.5. Evaluation of ROC and AUC

A good model has an AUC that is relatively close to 1, which indicates that it has a good measure of separability. A bad model will have an AUC that is close to 0, which indicates that it has a weak measure of separability.

3. RESULT AND DISCUSSION

3.1. XG Boost (XGB)

Extreme Gradient Boosting, often known as XG Boost, is a machine learning toolkit that offers a gradient-boosted decision tree that may be implemented in a scalable manner (GBDT). It is the most popular machine learning library, and it includes both sequence and parallel tree boosting, which may be applied to challenges of regression, classification, and ranking. It can be seen from Figure 8 that XGB has shown the highest accuracy for class 4 as 93% with a macro average of 77% and micro average of 88%.

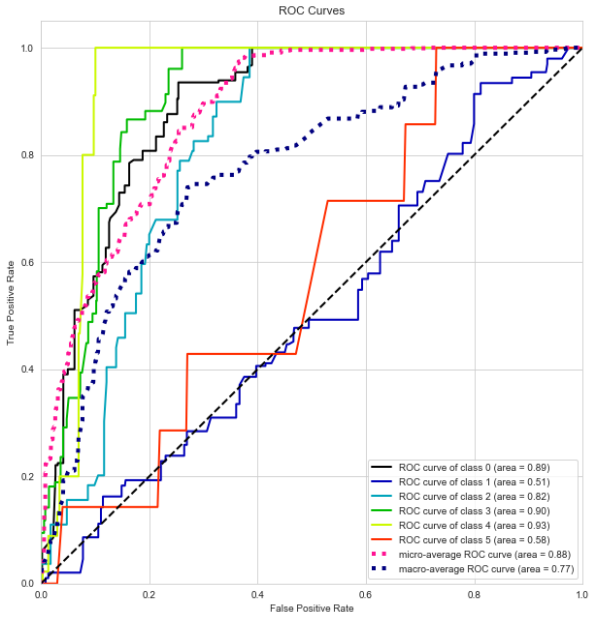

Figure 8. ROC of XGB

3.2. Cat Boost (CBC)

Cat Boost has the ability to manage in an automated fashion features that can be categorized. In spite of its widespread use, one-hot encoding cannot be applied when there are an excessive number of characteristics contained inside the data set. In order to solve this problem, features are divided into categories according to the data that are used to assess their level of success (estimate target value for each category). It can be seen from Figure 9 that CBC has shown 99% accuracy for class 0 and 1 while 100% accuracy for class 2, 3 and 4. CBC has shown the highest accuracy with micro average of 99% and macro average of 90%

.

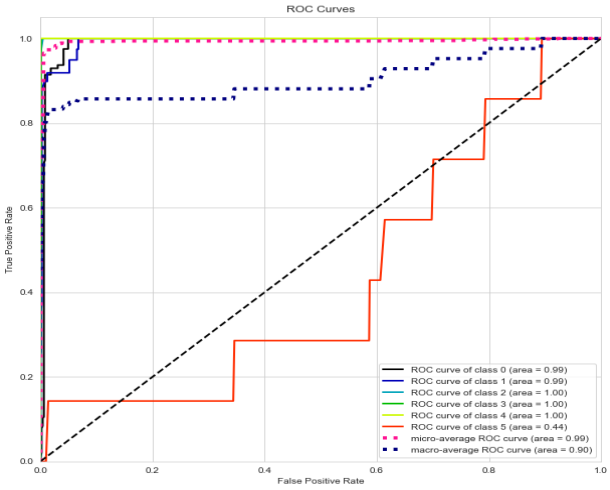

Figure 9. ROC of CBC

3.3. Gradient Boosting (GBC)

It is an iterative functional gradient algorithm that minimizes a loss function by continually selecting a function that tends toward the negative gradient or a hypothesis with poor statistical support. Gradient boosting is an example of this type of technique. GBC has shown highest 92% accuracy for class 3 and 4 with micro average 93% and macro average of 79% as shown in Figure 10.

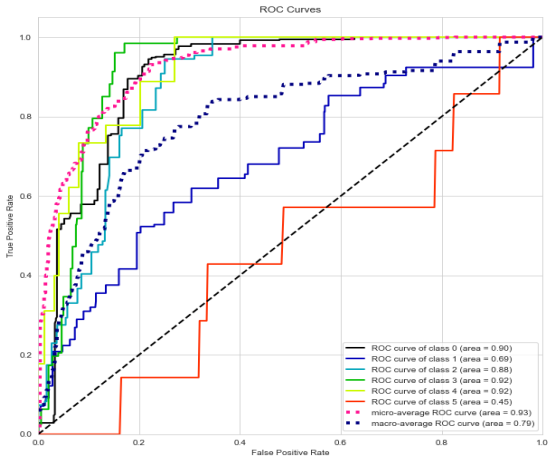

Figure 10. ROC of GBC

3.4. Ada boosting (ABC)

It uses the Boosting method as a component of an Ensemble Method for the purpose of machine learning. Adaptive boosting gets its name from the fact that it reassigns weights to each instance, providing examples that were incorrectly categorized a greater amount of weight than others in the process. It can be seen from Figure 11 that ABC model has shown highest accuracy of 91% for class 4 with micro average of 88% and macro average of 78%.

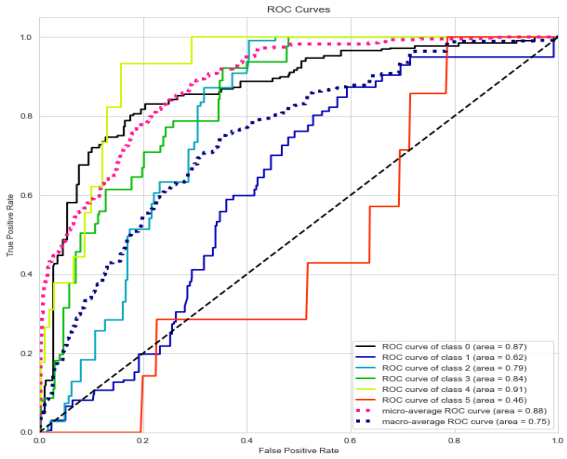

Figure 11. ROC of ABC

3.5. Light Gradient Boosting (LGBC)

LGBC is an alternative to more conventional boosting algorithms since it divides the tree at each leaf level as it grows larger. It chooses the leaf that offers the most potential for development in order to achieve the greatest possible increase in its delta loss. Because the leaf is always known, the loss that occurs when using the leaf-wise approach is far lower than when using the level-wise algorithm. It can be seen from figure 12 that LGBC has shown highest accuracy of 92% for class 2 and 3 respectively, with a micro average of 93% and macro average of 79%.

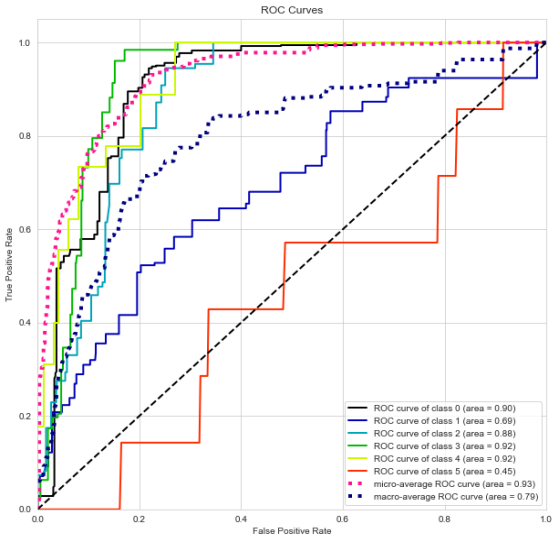

Figure 12. ROC of LGBC

3.6. HistGradient Boosting (HGBC)

The frequency of data (the number of occurrences) over intervals can be seen with the help of a histogram, which can also be used to tabulate the data (bins). The histogram method is rather easy to understand, as each box in the graph displays the frequency of the value to which it corresponds. It can be seen from Figure 13 that HGBC has shown highest accuracy of 91 for Class 4 with a micro average of 88% and macro average of 75%.

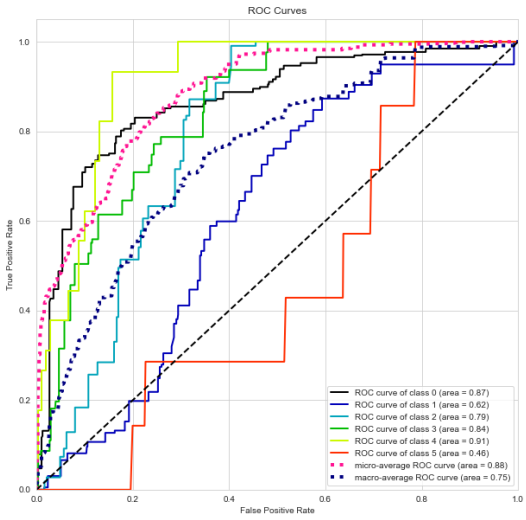

Figure13. ROC of HGBC

3.7. Comparative Analysis

CBC model has shown the highest accuracy for the classification of Deep Slices in 5G Network for Standalone Testbeds. CBC has shown 99% accuracy for class 0 and 1 while 100% accuracy for class 2, 3 and 4. CBC has shown the highest accuracy with micro average of 99% and macro average of 90%. While, LGBC has shown highest accuracy of 92% for class 2 and 3 respectively, with a micro average of 93% and macro average of 79%. On the other hand, HGBC has shown highest accuracy of 91 for Class 4 with a micro average of 88% and macro average of 75%. ABC model has shown highest accuracy of 91% for class 4 with micro average of 88% and macro average of 78%, and XGB has shown the highest accuracy for class 4 as 93% with a macro average of 77% and micro average of 88% while GBC has shown highest 92% accuracy for class 3 and 4 with micro average 93% and macro average of 79%.

4. CONCLUSIONS

Fifth-generation (5G) networks will rely largely on the implementation of Software Defined Networking, Network Function Virtualization, and Machine Learning in order to fulfil the particular needs of vertical industries and the stringent standards of 5G. Slice type forecasting, route optimization, and resource management are all areas where 5G networks could benefit from the application of machine learning. To evaluate machine learning's role in 5G networks, ideal test conditions are required. Using container lightweight virtualization technologies, this study proposes a lightweight testbed to aid in the development of machine learning network functionalities across the 5G network. To reduce the burden on 5G networks, we have collected the Deep Slice 5G dataset from Kaggle and are using it to create predictions about the nature of user-user communication based on packet loss and delay budget ratio. Machine Learning Boosted

models (XG Boost, Gradient Boost, Ada Boost, Light Gradient Boosting, Cat Boost, and Hist Gradient Boosting) were employed for slice prediction. Therefore, while selecting the proper 5G slice using a subset of the dataset's features, CBC model has shown the highest accuracy for the classification of Deep Slices in 5G Network for Standalone Testbeds. CBC has shown 99% accuracy for class 0 and 1 while 100% accuracy for class 2, 3 and 4. CBC has shown the highest accuracy with micro average of 99% and macro average of 90%. While, LGBC has shown highest accuracy of 92% for class 2 and 3 respectively, with a micro average of 93% and macro average of 79%. On the other hand, HGBC has shown highest accuracy of 91 for Class 4 with a micro average of 88% and macro average of 75%. ABC model has shown highest accuracy of 91% for class 4 with micro average of 88% and macro average of 78%, and XGB has shown the highest accuracy for class 4 as 93% with a macro average of 77% and micro average of 88% while GBC has shown highest 92% accuracy for class 3 and 4 with micro average 93% and macro average of 79%. In future studies this study can be implemented on 6G and can be transformed to deep learning in real time slicing prediction. The limitations associated with the use of Boosted Machine Learning models, including computational complexity and the interpretability of results. We recognize that these factors play a crucial role in the practical implementation and deployment of machine learning models, and their consideration is vital for a comprehensive understanding of the proposed methodology. Future work should focus on mitigating computational complexities associated with Boosted Machine Learning models, exploring methods for performance optimization without compromising accuracy. Enhancing the interpretability of machine learning models is pivotal. Research efforts should be directed towards developing methodologies that provide insights into the decision-making processes of complex models.

ACKNOWLEDGEMENTS

The authors wish to thank the editors and anonymous reviewers for their valuable comments and helpful suggestions which greatly improved the paper's quality. This work was supported in part by the National Key R&D Program of China (SQ2023YFB3100028), in part by the Natural Science Foundation of China (61972306, 62302371), and in part by SongShan Laboratory (YYJC012022005).

REFERENCES

[1] C. Ssengonzi, O. P. Kogeda, and T. O. Olwal, “A survey of deep reinforcement learning application in 5G and beyond network slicing and virtualization,” Array, vol. 14, no. January, p. 100142, 2022, doi: 10.1016/j.array.2022.100142.

[2] C.V. Nahum et al., “Testbed for 5G Connected Artificial Intelligence on Virtualized Networks,” IEEE Access, vol. 8, no. Ml, pp. 223202–223213, 2020, doi: 10.1109/ACCESS.2020.3043876.

[3] G. P. Koudouridis, Q. He, and G. Dán, An architecture and performance evaluation framework for artificial intelligence solutions in beyond 5G radio access networks, vol. 2022, no. 1. 2022.

[4] W. Package and D. Level, “5G Mobile Network Architecture Testbed setup and 5G-MoNArch technologies demonstrated,” no. 761445, 2016.

[5] K. Saeedi, “Machine Learning for Ddos Detection in Packet Core Network for IoT,” Comput. Sci. Eng., 2019.

[6] Y. Dai, D. Xu, S. Maharjan, G. Qiao, and Y. Zhang, “Artificial Intelligence Empowered Edge Computing and Caching for Internet of Vehicles,” IEEE Wirel. Commun., vol. 26, no. 3, pp. 12–18, 2019, doi: 10.1109/MWC.2019.1800411.

[7] A. A. Al-habob and O. A. Dobre, “Mobile Edge Computing and Artificial Intelligence: A Mutually-Beneficial Relationship,” 2020, [Online]. Available: http://arxiv.org/abs/2005.03100.

[8] S. M. Kumar and D. Majumder, “Healthcare Solution based on Machine Learning Applications in IOT and Edge Computing,” Int. J. Pure Appl. Math., vol. 119, no. 16, pp. 1473–1484, 2018.

[9] M. Ayaz, M. Ammad-Uddin, Z. Sharif, A. Mansour, and E. H. M. Aggoune, "Internet-of-Things (IoT)-based smart agriculture: Toward making the fields talk," IEEE Access, vol. 7, pp. 129551–129583, 2019, doi: 10.1109/ACCESS.2019.2932609.

[10] Y. Tsai and D. Chang, "applied sciences Edge Computing Based on Federated Learning for Machine Monitoring," 2022.

[11] F. Ali et al., "An intelligent healthcare monitoring framework using wearable sensors and social networking data," Futur. Gener. Comput. Syst., vol. 114, pp. 23–43, 2020, doi: 10.1016/j.future.2020.07.047.

[12] Y. Wang, H. Zen, M. F. M. Sabri, X. Wang, and L. C. Kho, "Towards Strengthening the Resilience of IoV Networks—A Trust Management Perspective," Futur. Internet, vol. 14, no. 7, pp. 1–21, 2022, doi: 10.3390/fi14070202.

[13] C. K. Leung, Y. Chen, S. Shang, and D. Deng, "Big Data Science on COVID-19 Data," Proc. - 2020 IEEE 14th Int. Conf. Big Data Sci. Eng. BigDataSE 2020, pp. 14–21, 2020, doi: 10.1109/BigDataSE50710.2020.00010.

[14] Y. Dong and Y. D. Yao, "IoT platform for covid-19 prevention and control: A survey," IEEE Access, vol. 9, pp. 49929–49941, 2021, doi: 10.1109/ACCESS.2021.3068276

AUTHORS

Hafiz Bilal Ahmad received the B.S. degree from Mirpur University of science and technology, Azad Kashmir, Pakistan in 2017 and M.S. degrees from North University of China, Taiyuan, China, in 2022.Currently He is pursuing Ph.D. degree in computer science and technology from Xidian University, Xi'an, China. His research interests include Network security, ML security, and privacy protection.

Haichang Gao (Member, IEEE) received the Ph.D. degree in computer science and technology from Xi'an Jiaotong University, Xi'an, Shaanxi, China, in 2006. He is currently a Professor with the School of Computer Science and Technology, Xidian University, Xi'an. He has published more than 75 papers. He is currently in charge of a project of the National Natural Science Foundation of China. His current research interests include Captcha, computer security, and machine learning.

Fawwad Hassan Jaskani has done his engineering degree from Islamia University of Bahawalpur then switched to MS Engineering from Islamia university of Bahawalpur. He has published more than 40 papers. Currently he is defending his thesis of PhD from UTHM Malaysia. His current research interests include Machine Learning, Neural Networks, image processing and Security.