International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 08 Issue: 03 | March - 2024 SJIF Rating: 8.176 ISSN: 2582-3930

Network Type Recognition Using Machine Learning Techniques

Author - Debmalya Ray

Email - debmalya.ray9989@gmail.com

Abstract:

The telecom industry is going through a massive digital transformation with the adoption of ML, AI, feedback-based automation and advanced analytics to handle the next generation of applications and services. AI concepts are not new; the algorithms used by Machine Learning and Deep Learning are being currently implemented in various industries and technology verticals. With growing data and an immense volume of information over 5G, the ability to predict data proactively, swiftly and with accuracy, is critically important.

Data-driven decision-making will be vital in future communication networks due to the traffic explosion and Artificial Intelligence (AI) will accelerate the 5G network performance.

Mobile operators are looking for a programmable solution that will allow them to accommodate multiple independent tenants on the same physical infrastructure and 5G networks allow for end-to-end network resource allocation using the concept of Network Slicing (NS).

Network Slicing will play a vital role in enabling a multitude of 5G applications, use cases, and services. Network slicing functions will provide end-to-end isolation between slices with an ability to customize each slice based on the service demands (bandwidth, coverage, security, latency, reliability, etc).

Index Terms:

Supervised Learning, Feature Engineering, Python, Network Slicing, Telecom, Classification Problems, EDA – Exploratory Data Analysis.

1. Introduction:

At present, few case studies focus on solving the network slicing problem using ML techniques and IOT data. For business applications with stringent latency specifications that need to be built on a common infrastructure, network slicing is an essential need to fill up the demand for network resources for various applications.

Our motivation is to take one such task to build a Machine Learning model that will be able to proactively detect and eliminate threats based on incoming connections thereby selecting the most appropriate network slice, even in case of a network failure.

Dataset Description

LTE/5g - User Equipment categories or classes to define the performance specifications

Packet Loss Rate - number of packets not received divided by the total number of packets sent.

Packet Delay - The time for a packet to be received.

Slice type - network configuration that allows multiple networks (virtualized and independent)

GBR - Guaranteed Bit Rate

Healthcare - Usage in Healthcare (1 or 0)

Industry 4.0 - Usage in Digital Enterprises (1 or 0)

IoT Devices - Usage

Public Safety - Usage for public welfare and safety purposes (1 or 0)

Smart City & Home - usage in daily household chores

Smart Transportation - usage in public transportation Smartphone - whether used for smartphone cellular data

2. Methods

Network slicing is a technique for managing the increasing complexity of manufacturing networks in the field of industrial communication. The process of creating and deploying network slices has been the topic of a significant amount of research. In this section, we summarize recent works related to slice creation through ML techniques and research on slice deployment in 5G.

1. Creating a Smart 5G Network

Network Slicing is set to be a prominent feature of 5G to allow connectivity and data processing tailored to specific customers' requirements. Mobile communications provided by smart networks will enhance the efficiency and productivity of business processes and will open up opportunities for operators to address different business requirements.

2. Building Business Opportunity

Different business verticals are seeking opportunities to improve productivity by leveraging the use of technology related to 5g. Network Slicing plays a critical role here in developing low-latency/high-performance applications that will support the transformation of digital services.

3. Understanding the customers' needs

The possibility of tailoring mobile network properties to the needs of the business through the configuration of a large set of parameters offers unsurpassed flexibility. However, with such a diverse range of possible requirements from verticals, operators need to manage risks from excessive complexity in the service offerings.

4. Building Correct Hypothesis

Simply put, a hypothesis is a research question that also includes the predicted or expected result of the research. Without a hypothesis, there can be no basis for a scientific or research experiment. As such, it is critical that you carefully construct your hypothesis by being deliberate and thorough, even before you set pen to paper. Unless your hypothesis is clearly and carefully constructed, any flaw can have an adverse, and even grave, effect on the quality of your experiment and its subsequent results.

5. Data pre-processing

The dataset itself is not always perfect. Some datasets have different data types, such as text, numbers, time series, continuity and discontinuity. It is also possible that the quality of the data is not good, there is noise,

there are anomalies, there are missing, the data is wrong, the dimensions are different, there are duplicates, the data is skewed, and the amount of data is too large or too small. For the data to fit the model and match the needs of the model, the Moore dataset needs to be pre-processed, detected from the data, corrected or deleted, inaccurate or inappropriate records for the model.

Data pre-processing methods include removing unique attributes, processing missing values, attribute coding, data standardization regularization, feature selection, principal component analysis and so on.

6. Appling ML algorithm

Based on the data being processed, a series of ML algorithms are used to extract the best results. Based on the problem statement discussed earlier, we are dealing with a classification problem and the algorithm used will help us to derive the best results.

The procedures used for calculating performance metrics are accuracy (AUC), precision, recall, and F1 Score.

Accuracy = TP + TN / TP + FP + FN + TN

Precision = TP / TP + FP

Recall = TP / TP + FN

3. Algorithms used

Logistic Regression

Logistic regression is a classification technique used in machine learning. It uses a logistic function to model the dependent variable. The dependent variable is dichotomous, i.e. there could only be two possible classes.

A cost function is a mathematical formula used to quantify the error between the predicted values and the expected values. Put simply, a cost function is a measure of how wrong the model is in terms of its ability to estimate the relationship between x and y. The value returned by the cost function is referred to as cost, loss or error. For logistic regression, the cost function is given by the equation:

$$\text{Cost}(h_\theta(x), Y(\text{actual})) = -\log(h_\theta(x)) \text{ if } y=1$$
$$-\log(1-h_\theta(x)) \text{ if } y=0$$

Mathematical Derivation: Logistic Regression

K-Nearest Neighbor

The KNN algorithm is a classic instance of a non-parametric classification approach because it does not presuppose anything about the training data. Using a labelled training dataset, where points have been assigned to several classes, it is possible to predict the class of unlabelled data. The unknown tuple can be classified using K's fixed value.

When KNN discovers a novel unlabelled tuple in the dataset, it performs two actions. First, it finds its K-nearest neighbors or points that are immediately adjacent to the new data point.

The second part of KNN is that it uses neigh boring data to figure out what category the new information belongs in. The Euclidean distance should be used to calculate the distance between the test sample and the specified training samples. The Euclidean distance function is as follows:

$$\sqrt{\sum_{i=1}^{m}(x_i - y_i)^2}.$$

Mathematical Derivation: KNN

Support Vector Machine

With proper training, support vector machines (SVMs) can detect the hyperplane that effectively divides a dataset into two classes, allowing for the classification of data points into one of two groups. The SVM creates a model from the data in the training set.

The model is then utilized to classify specific instances within the test data after it has been constructed. Consequently, the SVM locates the hyperplane that divides the data into its
two classes in the best possible way. The SVM hyperplane is denoted as:

$$h_{w,b}(x) = g(w^2x + b),$$

where w, x, b refer to weights, input, and bias, respectively.

Mathematical Derivation: SVM

Using AUTO ML

`*AutoML*` provides tools to automatically discover good machine learning model pipelines for a dataset with very little user intervention. It is ideal for domain experts new to machine learning or machine learning practitioners looking to get good results quickly for a predictive modelling task.

Open-source libraries are available for using AutoML methods with popular machine learning libraries in Python, such as the scikit-learn machine learning library.

EVAL ML

There are a variety of ML problem types. Supervised learning describes the case where the collected data contains an output value to be modelled and a set of inputs with which to train the model. EvalML focuses on training supervised learning models. EvalML supports three common supervised ML problem types. The first is regression, where the target value of the model is a continuous numeric value.

Next are binary and multiclass classification, where the target value of the model consists of two or more discrete values or categories. The choice of which supervised ML problem type is most appropriate depends on domain expertise and on how the model will be evaluated and used.

	id	pipeline_name	search_order	ranking_score	mean_cv_score	standard_deviation_cv_score	percent_better_than_baseline	high_variance_cv	parameters
0	1	Random Forest Classifier w/ Label Encoder + Im...	1	4.440892e-16	4.440892e-16	0.000000e+00	100.000000	False	{'Label Encoder': {'positive_label': None}, 'I...
1	3	Extra Trees Classifier w/ Label Encoder + Impu...	3	4.440892e-16	4.440892e-16	0.000000e+00	100.000000	False	{'Label Encoder': {'positive_label': None}, 'I...
2	2	LightGBM Classifier w/ Label Encoder + Imputer...	2	1.137570e-06	1.137570e-06	2.102637e-11	99.999993	False	{'Label Encoder': {'positive_label': None}, 'I...
3	5	XGBoost Classifier w/ Label Encoder + Imputer ...	5	1.418487e-04	1.418487e-04	4.158776e-07	99.999153	False	{'Label Encoder': {'positive_label': None}, 'I...
4	4	Elastic Net Classifier w/ Label Encoder + Impu...	4	4.380961e-04	4.380961e-04	7.629913e-06	99.997383	False	{'Label Encoder': {'positive_label': None}, 'I...
5	6	Logistic Regression Classifier w/ Label Encode...	6	5.086381e-04	5.086381e-04	1.026568e-05	99.996962	False	{'Label Encoder': {'positive_label': None}, 'I...
6	0	Mode Baseline Multiclass Classification Pipeline	0	1.674177e+01	1.674177e+01	2.945292e-03	0.000000	False	{'Label Encoder': {'positive_label': None}, 'B...

code snippets: EVAL ML

H2O ALGORITHM

Automated Machine Learning (AutoML) is the process of automating tasks in the machine learning pipeline such as data pre-processing, hyperparameter tuning, model selection and evaluation. In this article, we will examine how to utilize an open-source automated machine learning package from H2O to accelerate a Data Scientist's model development process.

After the models are trained, we can compare the model performance using the leaderboard. H2O AutoML produces a leaderboard which ranks the trained model based on a predefined metric. By default, it ranks models by ascending order of log loss and rmse for classification and regression tasks respectively.

```
AutoML progress: |████████████████████████████████████████████████████| (done) 100%

Model Details
=============
H2ORandomForestEstimator : Distributed Random Forest
Model Key: DRF_1_AutoML_2_20240221_95129

Model Summary:
| number_of_trees | number_of_internal_trees | model_size_in_bytes | min_depth | max_depth | mean_depth | min_leaves | max_leaves | mean_leaves |
|---|---|---|---|---|---|---|---|---|
| 50.0 | 50.0 | 5461.0 | 2.0 | 6.0 | 2.94 | 3.0 | 7.0 | 4.02 |

ModelMetricsRegression: drf
** Reported on train data. **

MSE: 0.0
RMSE: 0.0
MAE: 0.0
RMSLE: 0.0
Mean Residual Deviance: 0.0

ModelMetricsRegression: drf
** Reported on cross-validation data. **

MSE: 0.0
RMSE: 0.0
MAE: 0.0
RMSLE: 0.0
Mean Residual Deviance: 0.0
```

code snippets: H2O AUTOML

4. Plots and Figures

Figure 1 shows the sizes of the trains and tests used for model training and evaluation.

```
print(train_dataset.shape, test_dataset.shape)

    (31583, 17) (31584, 16)
```

Figure 1 Train and Test Datasets

Figure 2 explains the correlation among the variables. Variables with high multicollinearity provide redundant information, similar to how correlated features do. However, multicollinearity is more problematic because it inflates standard errors and undermines the reliability of estimated coefficients. By examining correlation matrices and variance inflation factors, machine learning practitioners can identify cases of multicollinearity between input features. This allows them to address multicollinearity through techniques such as principal component analysis or ridge regression to improve model stability and interpretability. Understanding correlations is crucial for diagnosing and mitigating the adverse effects of multicollinearity on predictive modelling.

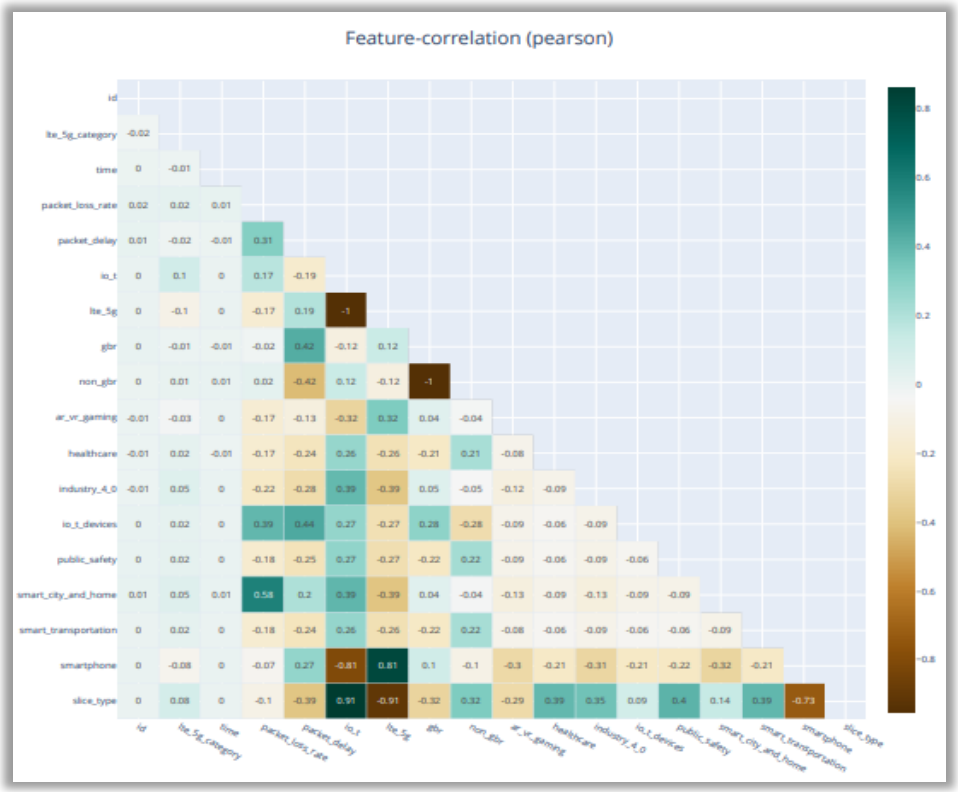


Figure 2 Correlation among the variables used for modelling

Figure 3 explains the numerical distribution of the data along an axis.

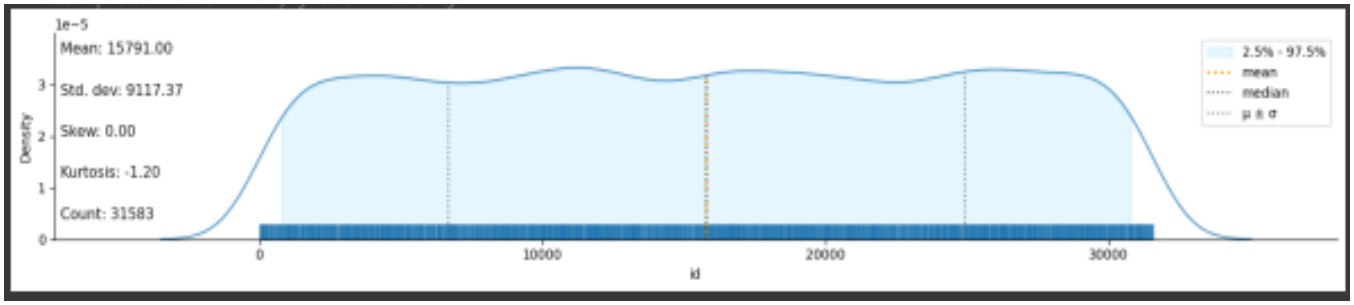


Figure 3 Numerical variables distributed across an axis

Figure 4 explains the feature selection techniques. The goal of feature selection techniques in machine learning is to find the best set of features that allows one to build optimized models of studied phenomena.

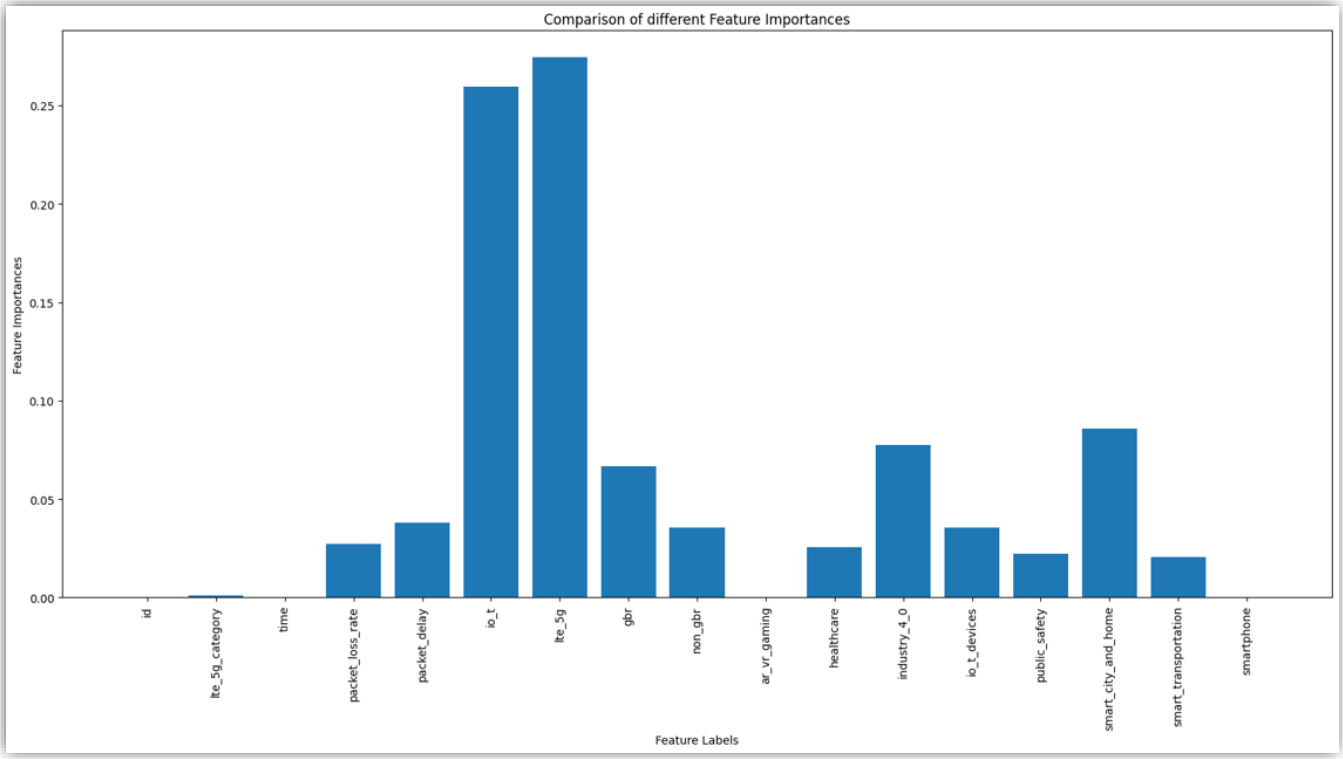


Figure 4 Feature Selection Techniques

Figure 5 explains the prediction of target variables as categorical types using a pie plot. Each categories are explained as a percentage type.

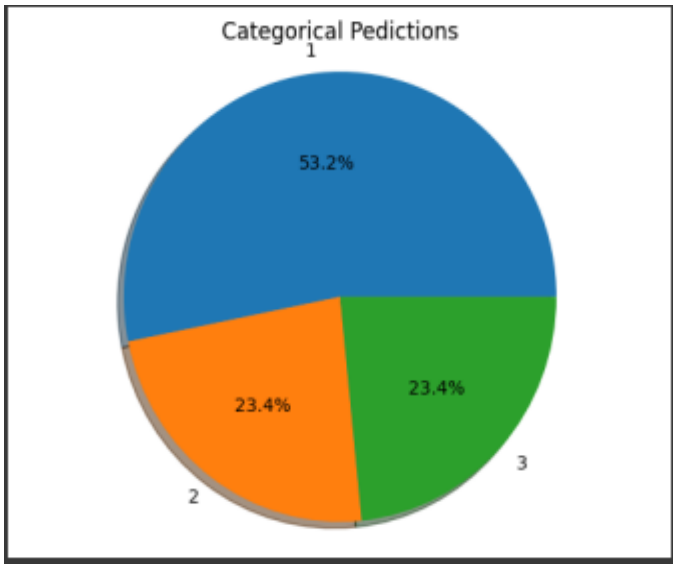


Figure 5 Pie - Chart

5. Results

The final results predicted from the dataset are as follows:

Network Slices	Count
CLASS 1	16800
CLASS 2	7392
CLASS 3	7392

Table 1: Final Results

6. Metrics

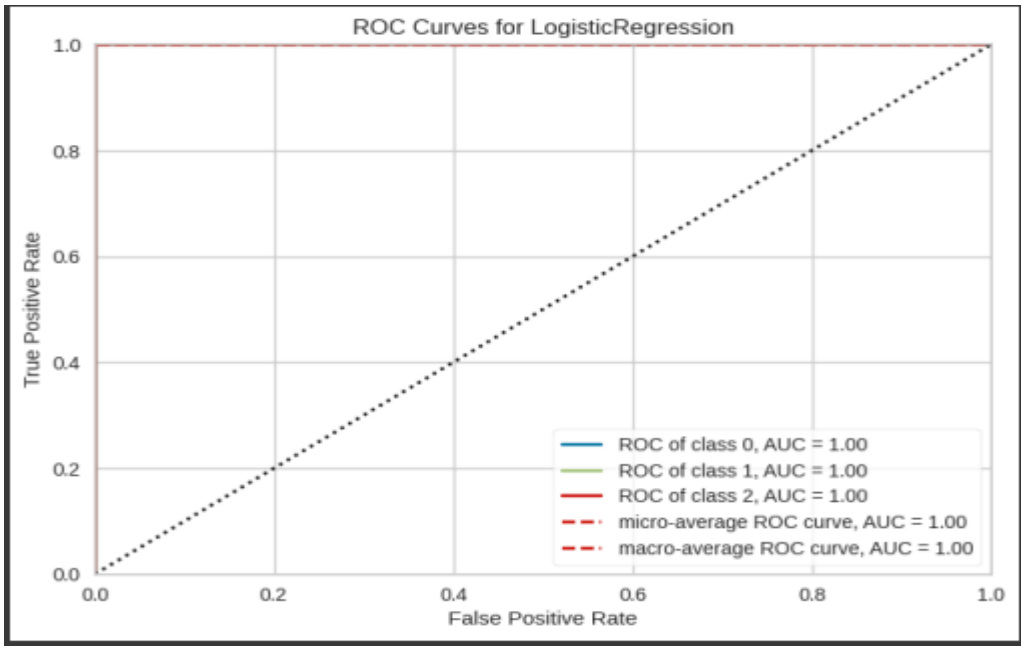
Before selecting the best result, we performed a comparative analysis using various algorithms. The evaluation used is the foundation metric computed based on a series of algorithms.

Algorithm Used	Precision	Recall	F1-Score	Accuracy
Logistic Regression	1.00	1.00	1.00	1.00
K Nearest Neighbors	1.00	1.00	0.94	1.00
Random Forest	1.00	0.90	0.95	1.00
SVM	0.93	0.93	0.9149	0.90

Table 2: Comparative Metrics

7. Plots derived from metrics

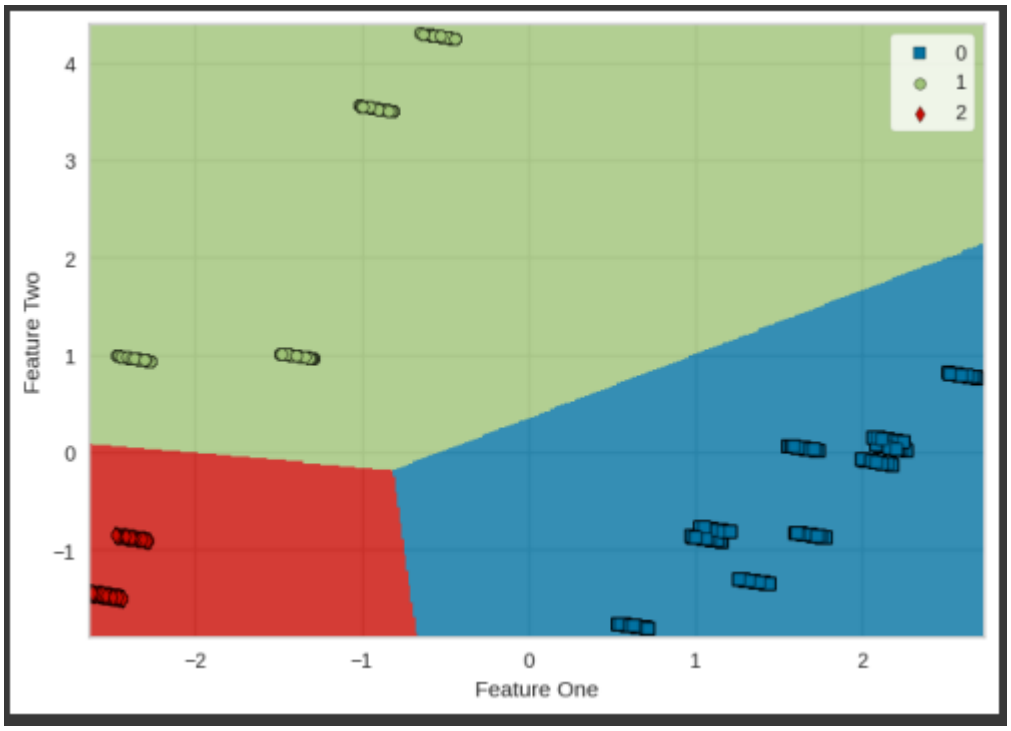
A *ROC curve* (receiver operating characteristic curve) is a graph showing the performance of a classification model at all classification thresholds.



ROC Curves for Classification Algorithm

A *decision boundary* is the region of a problem space in which the output label of a classifier is ambiguous.

If the decision surface is a hyperplane, then the classification problem is linear, and the classes are linearly separable. Decision boundaries are not always clear-cut.



Decision Boundary for Classification Algorithm

8. Data availability statement

Available online at:

Test Dataset

https://github.com/DebmalyaRay9989/networkslicing/blob/main/test_dataset.csv

Train Dataset

https://github.com/DebmalyaRay9989/networkslicing/blob/main/train_dataset.csv

9. Code availability

The codes for analysing the data from mapping the feature variables (Usage Class, Title, Subjects, Publisher etc) to the target variable (Network Class Type) using both Train and Test Data are available on GitHub:

https://github.com/DebmalyaRay9989/networkslicing/tree/main

10. Proof of Concept

The application is created and deployed in the **SAAS** platform. The URL of the application is:

https://networkslicing.onrender.com/

11. Discussion:

With the growth of emerging technologies, the requirement of 5G in the market will increase, since fast internet will be in demand to fulfil the needs of the market. More and more intelligent home appliances have entered the family.

Coupled with the frequent use of the Internet in life, the network slice is becoming more and more mature, and the data traffic has increased dramatically. By considering the various threats and challenges discussed in this paper regarding security concerns, we can see there is a need to minimize those threats in 5G network slicing. Since 5G is used in multiple sectors such as robotics, medicine, and automobile as well as applications such as IoT, Industry 4.0 and many others, these issues will impact the global market shortly, as discussed in the above sections.

Therefore, security in 5G network slicing is a primary concern. This paper also discussed and illustrated the taxonomies of security measures in terms of attack prevention and machine learning algorithms at various phases of network functions.

In this paper, the accuracy rate is almost 100%, which provides a good model reference for the recognition of network slicing.

12. Authors contributions:

All authors take part in the discussion of the work described in this paper. All authors read and approved the final manuscript.

13. Informed consent statement: Not applicable.

14. Conflicts of interest: The authors declare no conflicts of interest.

15. Funding: None

16. Abbreviations:

The following abbreviations are used in this manuscript:

ML	Machine Learning
LR	Logistic Regression
RF	Random Forest
SVM	Support Vector Machine
KNN	K Nearest Neighbor
XAI	Explainability AI
TP	True Positive
FP	False Positive
TN	True Negative
FN	False Negative

Table 3: Comparative analysis

17. References:

1. Babbar, H.; Rani, S.; AlZubi, A.A.; Singh, A.; Nasser, N.; Ali, A. Role of Network Slicing in Software Defined Networking for 5G: Use Cases and Future Directions. IEEE Wirel. Commun. 2022, 29, 112–118. [CrossRef]

2. Phyu, H.P.; Naboulsi, D.; Stanica, R. Machine Learning in Network Slicing—A Survey. IEEE Access 2023, 11, 39123–39153. [CrossRef]

3. Barakabitze, A.A.; Ahmad, A.; Mijumbi, R.; Hines, A. 5G network slicing using SDN and NFV: A survey of taxonomy, architectures and future challenges. Comput. Netw. 2020, 167, 106984. [CrossRef]

4. Everything You Need to Know about 5G. 2020. Available online: https://www.qualcomm.com/5g/what-is-5g# (accessed on 17 March 2022).

5. Zhang, H.; Liu, N.; Chu, X.; Long, K.; Aghvami, A.; Leung, V. Network Slicing Based 5G and Future Mobile Networks: Mobility. In Resource Management, and Challenges; IEEE Communications Magazine: 2017. Available online: https://ieeexplore.ieee.org/abstract/document/8004168 (accessed on 17 March 2023).

6. Foukas, X.; Patounas, G.; Elmokashfi, A.; Marina, M.K. Network slicing in 5G: Survey and challenges. IEEE Commun. Mag. 2017, 55, 94–100. [CrossRef]

7. Zhang, Q.; Liu, F.; Zeng, C. Online Adaptive Interference-Aware VNF Deployment and Migration for 5G Network Slice. IEEE/ACM Trans. Netw. 2021, 29, 2115–2128. [CrossRef]

8. Chirivella-Perez, E.; Salva-Garcia, P.; Sanchez-Navarro, I.; Alcaraz-Calero, J.M.; Wang, Q. E2E network slice management framework for 5G multi-tenant networks. J. Commun. Netw. 2023, 25, 392–404. [CrossRef]

9. Taleb, T.; Mada, B.; Corici, M.I.; Nakao, A.; Flinck, H. PERMIT: Network Slicing for Personalized 5G Mobile Telecommunications. IEEE Commun. Mag. 2017, 55, 88–93. [CrossRef]

10. Y. Sun, S. Qin, G. Feng, L. Zhang, M.A. Imran, Service provisioning framework for RAN slicing: user admissibility, slice association and bandwidth allocation. IEEE Trans. Mob. Comput. pp. 99 (2020)

11. Raghavendra Prasad, J.; Senthil, M.; Yadav, A.; Gupta, P.; Anusha, K. A comparative study of machine learning algorithms for gas leak detection. In Inventive Communication and Computational Technologies: Proceedings of ICICCT 2020; Springer: Berlin/Heidelberg, Germany, 2021; pp. 81–90

12. Archanaa, R.; Athulya, V.; Rajasundari, T.; Kiran, M.V.K. A comparative performance analysis on network traffic classification using supervised learning algorithms. In Proceedings of the 2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, Tamil Nadu, India, 6–7 January 2017; pp. 1–5.

13. Nair, M.R.; Ramya, G.; Sivakumar, P.B. Usage and analysis of Twitter during 2015 Chennai flood towards disaster management. Procedia Comput. Sci. 2017, 115, 350–358. [CrossRef]

14. Barabasi, A.L.; Albert, R. Emergence of Scaling in Random Networks. Science 1999, 286, 509–512. [CrossRef]

15. Thiruvenkadam, S.; Sujitha, V.; Jo, H.G.; Ra, I.H. A Heuristic Fuzzy Based 5G Network Orchestration Framework for Dynamic Virtual Network Embedding. Appl. Sci. 2022, 12, 6942. [CrossRef]

16. Y. Mu, G. Feng, J.H. Zhou, Y. Sun, Y.C. Liang, Intelligent resource scheduling for 5G radio access network slicing. IEEE Trans. Veh. Technol. 68, 7691–7703 (2019)

17. W. Wei, H. Song, W. Li, P. Shen, A. Vasilakos, Gradient-driven parking navigation using a continuous information potential feld based on wireless sensor network. Inf. Sci. 408(2), 100–114 (2017)

18. L. Yu, W. Wang, C. Runze, Zigbee-based IoT smart home system% the design of internet of things smart home system based on zigbee. Electron. Test. 000(005), 71–75 (2016)

19. Z. Zhang, J. Li, Y. Wang, Y. Qin, Direct detection of pilot carrier-assisted DMT signals with pre-phase compensation and imaginary noise suppression. J. Lightwave Technol. 39, 1611–1618 (2020)

20. Z.H. Wu, Research on the application of internet of things technology to digital museum construction. Acta Geosci. Sin. 38(2), 293–298 (2017)

21. Y. Wang, The innovation of computer internet of things technology in logistics feld. Log. Technol. 040(003), 41–42 (2017)

22. W.Q. Huang, M. Zhang, D. Wei, D.G. Sun, J. Shi, Efcient and anti-interference method of synchronising information extraction for cideo leaking signal. IET Signal Proc. 10(1), 63–68 (2016)

23. W. Wei, H. Song, W. Li, P. Shen, A. Vasilakos, Gradient-driven parking navigation using a continuous information potential feld based on wireless sensor network. Inf. Sci. 408(2), 100–114 (2017)