DECISION TREE-BASED RADIO LINK FAILURE PREDICTION FOR 5G COMMUNICATION RELIABILITY

Nethraa Sivakumar[1*], Pooja Srinivasan[1*], Nikhil Viswanath[1*], Venkateswaran N[2]
[1]Department of Electronics and Communication, SSN College of Engineering, Kalavakkam, Tamil Nadu, India, [2]Professor, Department of Electronics and Communication, SSN College of Engineering, Kalavakkam, Tamil Nadu, India

NOTE: Corresponding authors: Nethraa Sivakumar, nethraa18096@ece.ssn.edu.in; Pooja Srinivasan, pooja18112@ece.ssn.edu.in; Nikhil Viswanath, nikhil18098@ece.ssn.edu.in
* These authors contributed equally

Abstract – *Stable and high-quality Internet connectivity is mandatory for 5G mobile networks. Network disruption may occur due to unexpected variations in environmental conditions such as weather, wind, and natural or man-made surroundings, and the influence of the defect is quite severe. Prediction of such undesirable events at a low cost can boost 5G communication reliability, massive network capacity, and decreased latency. This research work makes use of novel preprocessing and feature engineering techniques, followed by a trained decision tree model to predict the occurrence of Radio Link Failure (RLF). This system is designed to predict RLF for not just the next day, but also any of the next 5 days. This prediction supports reliance and increasing demand for good Internet connectivity. In order to achieve accurate RLF prediction, comprehensive data has been used which undergoes preprocessing. To account for the influence of surrounding weather conditions on radio links, the proposed system makes use of information from the past i.e., previous RLFs, and the information from the future i.e., the weather forecast from the weather station around the radio link station. The decision tree model was trained with the integration of feature engineering. A macro-averaged F1-score of 70% and 77% were obtained for RLF prediction for the next day and RLF prediction for the next 5 days, respectively. The results show improvement in performance after the incorporation of feature engineering in the pipeline. Further, an additional metric termed G-Mean is introduced in the paper. Owing to the high imbalance in the dataset, this metric was found to provide a more realistic representation of the results. The G-Mean score was found to be 98.69% and 92.89% for RLF prediction for the next day and RLF prediction for the next 5 days, respectively.*

Keywords – 5G communication, decision tree, feature engineering, G-Mean metric, preprocessing, radio link failure

1. INTRODUCTION

Mobile devices and their modes of communication, whether visual or audio or video, have become increasingly reliant on high-speed, around-the-clock Internet access. The 5th generation of wireless networking technology, also known as 5G, was introduced to meet this ever-increasing demand. 5G technology has been and is conceived to be employed in a variety of ways in the future, taking advantage of its high data transfer speeds, high network capacity, and low latency. As a result of reduced waiting times and ease of access, as well as every remote action being made slicker and far more exact owing to 5G technology, the economic structure of the business could potentially be heightened.

With 7 billion people in this world, it has been predicted that by 2025 there will be about 12 billion Internet-connected devices with over 13 times more data under use than today. This radical digital transformation has been proclaimed to be the next big revolution. Maintaining connectivity to crucial equipment that regulates our safety and security has become increasingly important. Avoiding loss of data transfer is imperative and central to this is the network reaction times or latency that is inter- linked to high-speed Internet connectivity.

Part of a larger communication network, a two-way radio communication system is the Radio Link (RL). It plays a very important role when it comes to connection and ensuring the transmission of signal from the transmitter to the receiver. Stable and high-quality Internet connectivity as mentioned before is mandatory to 5G mobile networks, but certain defects or obstructions to the channel can deter the quality of transmission heavily. When the key parameters (such as signal strength and frequency offsets) of a radio channel are poor to continue with the desired application, it is said to be a radio link failure [1]. This failure is highly undesirable and needs to be avoided following its prediction at no cost for 5G communication reliability, massive network capacity and decreased latency.

Most radio links are exposed to various weather-related phenomena like wind, rain or snow [2]. The impact is more of concern particularly with respect to the backhaul links that operate in the GHz frequency range. When radio links are affected by such phenomena, there are chances for the communication system to fail in continuing its operation in the days following the effect. This becomes an issue considering the massive reliance on them for data transfer and uninterrupted connection. Thus, the prediction of a radio link becomes necessary.

In this paper, we propose a machine learning-based Radio Link Failure (RLF) prediction model for 5G communication reliability. It helps to predict the failure in radio links in any of the next five days given the weather forecast conditions surrounding the link. Considering that there are more and more radio stations built almost every year to address the increase in data consumption for high-speed Internet connectivity, the key features and specifications of the radio link and the weather stations to capture accurate data is ever increasing. This makes the data more and more complex to ensure a uniform user experience. To facilitate such complex data, the proposed dynamic model incorporates two features: (i) extensive preprocessing techniques and (ii) feature engineering methodologies. Following this, a decision tree model was used to obtain the prediction results. The proposed model makes use of information and data from the past, i.e., failure of radio links previously over the years and the data from the future, i.e., the weather forecast from the weather stations taking into account that weather conditions affect the performance of a link. F1-scores were used to evaluate the performance of the model before and after the implementation of feature engineering. Further, another metric called the G-Mean was introduced in order to account for the considerable imbalance present in the dataset.

The following parts of this paper are organized as follows: Section 2 compiles the literature survey to facilitate the approach as related work. Section 3 describes the data source, while decision tree concepts are explained in Section 4. The proposed approach is thoroughly analysed in Section 5, with the results obtained being summarised in Section 6. Finally, we conclude on the findings and talk about the future work in Section 7.

2. RELATED WORK

The phenomenon of radio link failure occurs very commonly, causing disruption in high-speed, low-latency data transfer through wireless channels. With 5G applications multiplying by the day, it is essential to predict and forecast the possibilities of failure in radio links to employ measures to solve such disruptions in connectivity.
Existing research work for radio link failure prediction considers multiple factors individually, such as signal strength [3], weather phenomenon [4], mobility, real-time data, etc. [5]. While they prove successful for limited datasets and duration for a short period into the future, the current studies fail to consider multiple factors combined. We performed literature reviews on preprocessing, feature engineering, and training the artificial intelligence model to construct and support their approach for a robust RLF prediction system.

Shayea et al. [2] investigated the impact of weather phenomena, particularly rain, on 5G communications. The work highlights the importance of monthly variability in weather impact on wireless links. This paper considers rain rate and rain attenuation for the worst month as parameters, which is stated as an accurate method to observe the variability in rainfall and its effect. Through such analysis, the authors discovered that the effect of rain on wave propagation is highest for tropical climates. By observing an increase in rain attenuation with rain rate, the importance of rainfall intensity and thereby the effect of the time of the year and the surroundings are presented. But, the authors of this paper stick to a specific set of frequency bands and a limited geographical region. Additionally, there are other weather factors besides rainfall which they miss to consider, such as humidity and temperature.

The model proposed in this paper is aimed to predict radio link failure for the next day and the next five days, hence requiring more than one label. Here comes the need for multi-label classification as for this problem statement, the values would be related to more than one label, with the labels depending on each other as well. Shi and Wu [6] describe preliminary research to construct forecasting models of type selection on the basis of historical data of mobile base stations. Bi and Kwok [7] propose a randomized-sampling based approach to classify examples into multiple labels. Through such literature work, we perform classification for the labels which are based on the number of days to predict.

Robust feature engineering translates into simpler Machine Learning (ML) model requirements and efficient training. For large and unbalanced datasets, sufficient techniques applied to the available features would reduce the dimensionality and provide the necessary features which contain information of proper value as training data. Backward elimination is a feature engineering approach that improves failure prediction. True to its name, the technique starts with all features and removes the least significant features. This occurs in a loop fashion until there is no more improvement in the parameters considered for evaluation. Aggrawal and Pal [8] surveyed on the right selection of p-value for backward elimination and combined such feature engineering to ML models such as K-Nearest Neighbour (KNN) and random forests for the prediction of heart diseases.

The studies for the predictive model are spread across different classes based on the available dataset. Qin et al. [9] use multi-class support vector machines to predict handover-based radio link failures. Tom and Vasudevan [10] studied cellular link failure and developed rudimentary ML models such as logistic regression, naive bayes,

Table 1 – Influence of key performance indicators on radio links

KPI	Description	Influence on RL
Type	RL equipment vendor	Different manufacturers, quality difference
Date time	Date and timestamp of data collection	Sequential data over all months (climate change and seasons)
end-point	Binary-valued, saying if link end-point is near/far	Far (Long distances gives more signal strength attenuation) Near (Short distances gives less signal strength attenuation)
Mini Link ID	A unique ID given for each Minilink (most deployed microwave transmission system)	Affects transmission of radio link
Connection No.	Unique internal connection ID for each connection	Indicates path of communication between two radio stations
Site ID	Unique ID given for a site, in the form of RL_xxxxx (alphanumeric)	Does not affect
Polarization	Binary-valued, says if antenna is vertical/horizontal polarized	Vertical: Field strength perpendicular to ground Horizontal: Field strength in relation to ground
Card Type	Modem card type	The data transmission speed of different modems vary (measured in megabytes per second). Hence, card type affects the signal strength of RL
Adaptive Modulation	Whether adaptive modulation is employed or not	Allows radio to change its speed as conditions in radio network change
Frequency Band	5 Frequency bands in the GHz range	Increased frequency causes higher attenuation
Link length	Distance between 2 sites (LOS)	Inversely proportional to attenuation
Severely Errored Second	Count of 1 second periods with error that covers >= 30 % of the frame	Even one uncorrected bit error is enough to cause the loss of a data packet, whether that packet loss was caused by a single bit error or hundred-bit-long error burst is irrelevant
Errored Second	Count of 1 second periods with error	Same as ”Severely Errored Second”
Unavailable seconds	The unavailable operation duration in seconds	More means more probable connection failure
Available time	The active duration in seconds	More means more probable connection success
Max. received power level	Maximum transmitter output power (the actual amount of power of radio frequency energy that a transmitter produces at its output)	Effective radiated power is reduced when maximum transmitter output power is reduced due to loss/resistance from the feed line.
RL capacity	Physical transmission bit rate of source	Stronger signal strength is correlated with higher bit rates
Modulation deployed	Modulation type used	Higher order means higher bandwidth efficiency, used for higher frequency

and KNNs. The best accuracy was achieved through KNNs but required extensive data and large computational expenditure. Prone to indoor climate conditions, wireless sensor networks were studied and failures were detected using a newly proposed supervised learning network [4]. Equipped with temperature sensors, this system estimates the temperature in a remote location using the novel model designed. Yet, it fails to consider surrounding factors (impact of the environment around the receiver, humidity, rain, and distance from the transmitter).

A reinforcement learning approach [5] to handle RL failures was proposed by Karjee et al. This work focused on a restricted space, an elevator, as the transceiver point, using cellular data from a real elevator environment. Another research work concentrated on executing handovers before the complete disruption of connection due to failure using machine learning-based prediction of optimal time and destination of such handovers and to trigger them based on radio conditions [11]. The mentioned research work broadens the study on handling radio link failure in real-world scenarios, post the prediction phase.

Through a rigorous literature survey of similar work existing in the fields of failure prediction and handling, this paper formulates a machine learning model which is enhanced through elimination-based feature engineering post data preprocessing performed on the vast dataset. This approach considers multiple factors relevant to predicting failure. We process the weather factors and Key Performance Indicators (KPIs) through ranking and selecting only the necessary training samples by evaluating the distance metric between weather stations and RL stations. In this data, the main contributing metrics are retained by removing features that lower the evaluating parameter based on backward elimination and embedded method. By undergoing such data science techniques, the training of the model becomes easier.

3. DATA SOURCE

The data that is used for the ML model pipeline is provided by Turkcell in the context of the AI for 5G Challenge organized by International Telecommunication Union (ITU) [12]. This data consists of six files, each with different sets of information that needs to be processed and modelled for RLF prediction.

1. **Relative distances**: The distance between each weather station and RL station, between 2 weather stations, and between 2 RL stations is present.

2. **Weather forecast data**: This data consists of the weather forecast for every 5 days from 2nd January 2018 to 23rd December 2020.

3. **Meteorology historic realizations**: It consists of various meteorological parameters (temperature, humidity, wind speed, wind direction) as measured by the weather station in the surrounding area.

4. **Meteorology station information**: The information regarding the location surrounding the weather station, i.e., dense tree, low-medium urban, airport, low-sparse urban, sea, sparse tree, open land, open in urban along with station number and height of the weather station is given here.

5. **Radio link Key Performance Indicators (KPIs) and configuration parameters**: Key performance indicators include date, time, site ID, type, frequency band, polarization, radio link capacity, error statistics and link length (distance between 2 sites). The KPIs are depicted in Table 1.

6. **Radio link site information**: The information regarding the location surrounding the RL station, i.e., open land, dense tree, open in urban, average-medium urban, low-medium urban, sparse tree, high-dense urban, average-sparse, industrial and commercial, average-dense urban, low-dense urban, high-sparse urban along with the RL ID and height of the RL station.

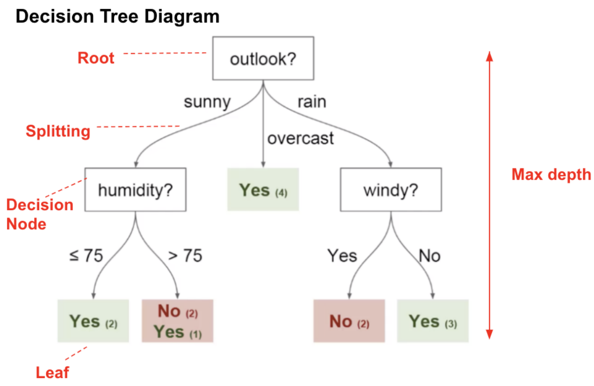

Fig. 1 – Example of decision tree diagram with labelled components [14]

The data that is used for the model is comprehensive yet highly imbalanced with just 0.061% failure weather forecast data. Accurate preprocessing steps along with feature engineering techniques to remove redundant data are carried out on this model for increased performance in RLF prediction.

Algorithm 1 Decision tree algorithm [15]

1: Begin the tree with the root node, says S, which contains the complete dataset.
2: Find the best attribute in the dataset using Attribute Selection Measure (ASM).
3: Divide the S into subsets that contain possible values for the best attributes.
4: Generate the decision tree node, which contains the best attribute.
5: Recursively make new decision trees using the subsets of the dataset created in Step 3.
6: Continue this process until a stage is reached where you cannot further classify the nodes and call the final node as a leaf node. Fig. 1 depicts the steps with labelled components.

---

4. DECISION TREE

A decision tree is a flowchart-like structure in which each internal node represents a test on a feature (e.g. whether a coin flip comes up heads or tails), each leaf node represents a class label (decision taken after computing all features), and branches represent conjunctions of features that lead to those class labels [13]. The paths from the root to the leaf represent classification rules. It is a supervised method. Algorithm 1 explains the decision tree algorithm.

Algorithm 2 Preprocessing approach

1: For each unique *site_id* and *mlid* pair, Day-1 & Day-5 radio link failure predictions are found for each date. This is present in a data frame named 'Intermediate Data'.
2: 'Intermediate Data' is merged with original radio link configuration data to give the merged dataset which contains two additional parameters. Hence, the original 19 parameters become 21 parameters after merging.
3: Next, weather data is separated into parts corresponding to each day's data. In the first day's part (containing 8 parameters), *report_time* and *weather_day1* are ignored while only numerical data fields are retained.
4: Whenever there is a repetition of a station no. and date time pair, the mean of the numerical values for the repeated samples is found and hence eradicate the repeated samples. The data is then merged with *report_time* and *weather_day1*.
5: The nearest weather station (obtained through Algorithm 3) for each sample is added as a parameter to the 'Merged Radio Link Configuration' data. Now it contains one additional parameter and the total number of parameters becomes 22.
6: Using nearest station, 7 parameters are obtained, which correspond to the minimum and maximum temperatures, humidity, and wind direction for the first day, along with the weather for that day.
7: This is done by obtaining the weather parameters for the weather station that is nearest to the RL site under consideration.
8: The weather parameters are present in the 'Modified Weather Data'.
9: After combining the 'Merged Radio Link Configuration data' with the corresponding weather parameters, totally there are 29 parameters (7 new parameters from weather data)
10: The final set of columns is added to 'Merged Radio Link Configuration Data'.
11: With the aim to exploit prior information, the radio link parameters of the previous day corresponding to the particular *mlid* and *site_id* are appended.
12: Hence, the number of parameters now increases from 29 to 51.

Advantages of using decision trees
They can handle both categorical and numerical data. The data present with us is high dimensional and has a fair number of categorical and numerical features. Hence, decision trees are ideal to handle this diversity.

Algorithm 3 Algorithm for finding nearest weather station for each RL site

---

1: List of weather stations, distance between each weather station, radio link site pair are present in the database. (Obtained from dataset)
2: A *site_id* 'x' is fed as input. Aim is to obtain the nearest weather station for the RL site 'x', as shown in Fig. 3.
3: For 'x', the list of distances corresponding to each weather station - RL site 'x' is obtained.
4: The above list is ranked and sorted based on distance values in ascending order.
5: The first weather station in this list is the nearest weather station for RL site 'x'.

---

5. PROPOSED APPROACH

The decision tree-based model involves the integration of extensive preprocessing, feature engineering and model building from the dataset to account for the imbalance dataset. The steps involved are depicted in Fig. 2.

5.1 Preprocessing

The dataset labels are split in a highly imbalanced ratio, 19,87,625 values corresponding to success and 1,233 failure. Here, 'success' means no RL failure, and 'failure' indicates the occurrence of RL failure. Such a narrow split with highly scarce failure values (0.061%) would act as inefficient training data, as sufficient failure label examples are the rate of failure prediction should be maximum.

Hence, we propose a preprocessing approach to remove redundant parameters and retain maximal information. The approach (explained below as a sequence of steps in Algorithm 2) includes dealing with the date time parameter, finding the nearest weather station, handling null values, and encoding categorical data.

Handling NULL values
Deleting the samples in the dataset containing null values for certain parameters is not advisable as it would lead to a loss of information. Instead, we go for the following approach:

1. For numerical parameters: Mean is found from all samples for each *mlid* (excluding NULL) for that parameter. Then replace NULL values for that parameter with the computed mean. The mean is substituted based on the *mlid* under consideration.

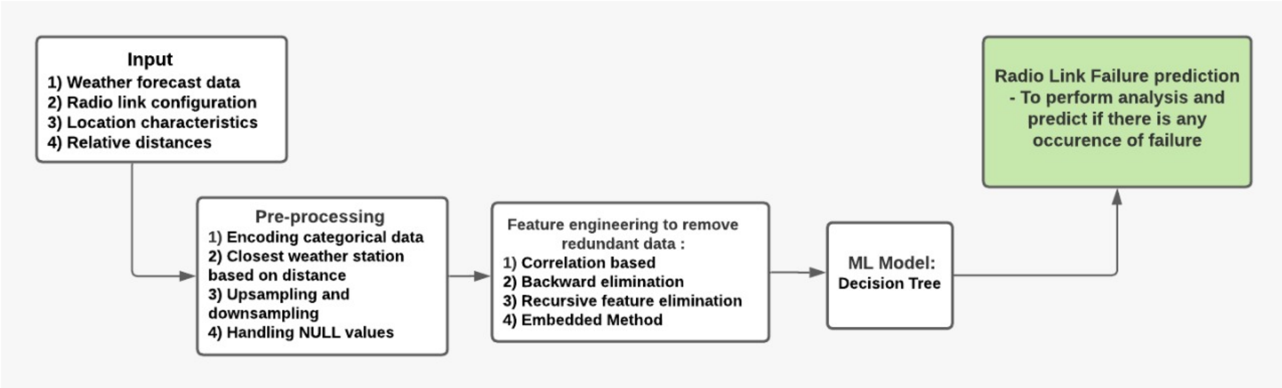

Fig. 2 – Block diagram of proposed pipeline

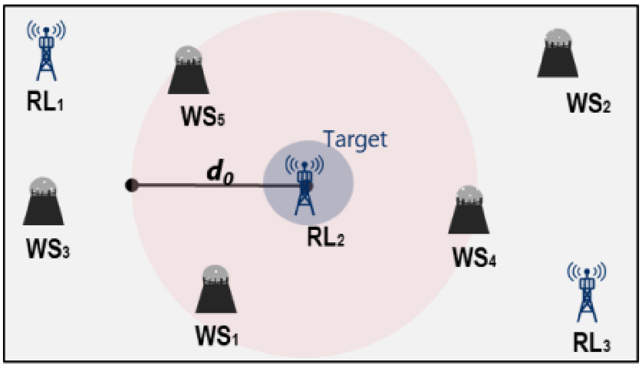

Fig. 3 – Weather stations in vicinity of radio station. RL1, RL2, RL3 represent the radio stations. WS1, WS2, WS3, WS4, WS5 represent the weather stations.

2. For categorical parameters: Mode is found from all samples for each *mlid* (excluding NULL) for that parameter. Then replace NULL values for that parameter with the computed mode. The mode is substituted based on the *mlid* under consideration.

The NULL values are depicted in Fig. 4.

```
type                      0
datetime                  0
tip                       0
mlid                      0
mw_connection_no          0
site_id                   0
card_type                 0
adaptive_modulation       0
freq_band             23316
severaly_error_second     0
error_second              0
unavail_second            0
avail_time                0
bbe                       0
rxlevmax                  0
capacity                164
modulation                0
rlf                       0
1-day-predict         25916
5-day-predict             0
forecast_datetime         0
nearest_station           0
```

Fig. 4 – The number of samples containing NULL value for some of the parameters of the dataset are shown.

Algorithm 4 Handling imbalance across months

1: The months with the number of samples close to the median are found. In this case, months 6, 8, 9, 10 are closest to the median. The rest need to be upsampled or downsampled.
2: Mean is found for the months close to the median. In this case, the mean for months 6, 8, 9, 10 was found to be 167920.
3: The months 2, 4, 12 are upsampled to match it to the mean computed in the previous step. Next, the months are downsampled 1, 3, 5, 7, 11 to match it to the computed mean.

Handling imbalance across months

To eliminate any bias that may arise due to an imbalance in the number of samples available for each month, upsampling and downsampling is carried out.

Algorithm 4 explains the process to handle imbalance of samples across months. The distribution of samples before and after upsampling and downsampling is shown in Table 2.

Algorithm 5 Encoding categorical data

1: Convert 'True' and 'False' from string to Boolean form.
2: Leave numerical data as it is after checking if it is in float form and not string.
3: Encode categorical data using label encoding.

Encoding categorical data

The accumulated dataset given as input to the system contains multiple fields which are numerical or categorical. To aid the successful training of machine learning models, it is important to perform encoding techniques to uniformly label the fields as numerical features.

Table 2 – Distribution of samples after upsampling and downsampling

Month	Before	After
1	172159	167920
2	141998	167920
3	176131	167920
4	141854	167920
5	176615	167920
6	165888	167920
7	180034	167920
8	141998	167920
9	170555	167920
10	164402	167920
11	174675	167920
12	153699	167920

The 'one hot encoding' approach is not considered as it contributes to increased dimensionality. For high dimensional data, observations become harder to cluster, causing every sample in the dataset to appear equidistant from all other observations. As clustering uses a distance measure, Euclidean distance, to quantify the similarity between samples, this phenomenon of points being equidistant results in no meaningful clusters being formed. On certain occasions, this increase in dimensionality might also cause overfitting.

To overcome these, we go for label encoding to encode the categorical data (refer to Algorithm 5). Ordinal encoding or its counterpart without numerical values, label encoding, does not add dimensionality, yet effectively encodes the categorical data even while non-binary [16]. As the data is extensive, label encoding will ensure lower time complexity during model training.

5.2 Feature engineering

The data used is highly imbalanced with 19,87,625 (99.93%) success and 1,233 (0.061%) failure data, with some redundant information that contributes to decreased accuracy. The predictive model that is built is directly influenced by the features in the data. There might be some features that are larger in quantity and may influence the model to produce misleading results. Thus to overcome this, feature engineering techniques are used to identify redundant data that do not give accurate radio link failure prediction. Feature engineering is a preprocessing step that moulds the raw data into features that better represent the underlying unseen

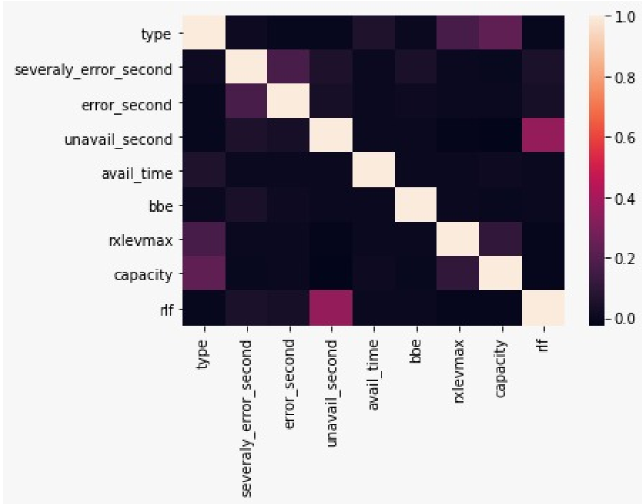

Fig. 5 – Correlation matrix

Table 3 – p-values obtained for the parameters using backward elimination method

S.No.	Parameter Name	p-value
1	severally_error_second	2.22172581e-007
2	error_second	0.00000000e+000
3	unavail_second	6.10771950e-183
4	avail_time	0.00000000e+000
5	bbe	7.65626684e-002
6	rxlevmax	7.10625958e-002
7	capacity	1.62893325e-012

information. In this paper, three methods were used to perform feature elimination namely, correlation-based, backward elimination, and embedded methods.

Correlation-based approach

This approach first filters and selects only a subset of features. After choosing the features, the model is produced. The correlation matrix is used to filter the data, and Pearson correlation is the most frequent method which has been utilized as well. First, a Pearson correlation heatmap is created and then examined to enable finding the relationship between the independent factors and the output variable. Only features with a correlation of more than 0.5 (in absolute value) with the output variable will be considered. If the correlation coefficient value is close to 0 it implies a weaker correlation; If the correlation coefficient is closer to 1 then it implies stronger positive correlation; And if the correlation coefficient is closer to -1 then it implies a stronger negative correlation.

Table 4 – Coefficient values obtained for the parameters using embedded method

S.No.	Parameter Name	Co-eff value
1	severally_error_second	5.471792e-05
2	error_second	2.870396e-05
3	unavail_second	3.697294e-04
4	avail_time	0.000000e+00
5	bbe	2.711982e-09
6	rxlevmax	-0.000000e+00
7	capacity	-9.559263e-08

There was no such pair with high correlation value, providing no conclusive set of features to be removed and hence this method is not suitable for indicating redundant columns. The same is observed in Fig. 5.

Backward elimination
In this method, all possible feasible features are first fed to the model, then the model's performance is evaluated. The worst performing feature is eliminated one by one until the model's overall performance is within acceptable bounds. The p-value performance metric was employed to assess feature performance in this case. If the p-value is greater than 0.05, the feature is removed; otherwise, it is kept. The higher the p-value, the lower is the significance of the feature. The ordinary least square model is used in backward elimination to obtain the final set of features.

It is seen that using this method the feature numbers 5 and 6 have a p-value greater than 0.05 as shown in Table 3. Hence, these features are considered redundant and can be removed. The final set of features namely *bbe* and *rxlevmax* are obtained using this iterative process, which could be dropped to improve performance.

Embedded method
In embedded approaches, meticulous extraction of the features that contribute the most to the training for each iteration of the model training process is done continuously. Regularization methods, which make use of a feature based on a coefficient threshold, are the most widely used embedding approaches. Least absolute shrinkage and selection operator (Lasso) regularization is put into use in this case to choose features. If a feature is unimportant, Lasso distinguishes it by setting its coefficient to 0. As a result, the features with coefficient = 0 are deleted, leaving the rest. The lower the coefficient value, the lower is its significance and higher is its redundancy.

Table 5 – Consolidated results of feature engineering

S.No.	Feature Engineering Method	Predicted redundant features
1	Correlation based approach	Inconclusive
2	Backward elimination	bbe and rxlevmax
3	Embedded method	bbe, capacity, avail_time and rxlevmax
	Combining all 3 methods	bbe and rxlevmax

Table 6 – Labels and class

Label Name	Class Names and Class Description	Label Description
RLF for one day	1.0 : Occurrence of RLF 0.0 : Absence of RLF	Radio Link Failure occurring on the next day.
RLF for five days	1.0 : Occurrence of RLF 0.0 : Absence of RLF	Radio Link Failure occurring on any of the next 5 days.

Table 7 – Number of samples present in the sets before and after Train-test split

Name of Set	Number of Samples
Merged Dataset after Preprocessing (Obtained in Section 5.1)	434570
Train Set	304193
Test Set	130377

	Label 1 (RLF for 1 day)	Label 2 (RLF for 5 days)
Without Feature Engineering	W/O FE for 1 DAY	W/O FE for 5 DAYS
With Feature Engineering	W FE for 1 DAY	W FE for 5 DAYS

Fig. 6 – Name convention for the four conditions

Using the embedded method it was observed that *bbe*, *capacity*, *avail_time* and *rxlevmax* have low coefficient values as depicted in Table 4. Thus, these features can be dropped as they are redundant. *bbe* and *rxlevmax* are the overlapping redundant features obtained by the backward elimination method and embedded method. Hence, these factors were dropped during training. It is also observed that these two methods give more accurate results as they are computationally expensive. However, the correlation-based approach, though used widely, did not give conclusive evidence of which features are redundant for the dataset used in this paper. Hence, the approach could not be leveraged. Table 5 demonstrates the same.

5.3 Training using decision tree

Labels and classes

The decision tree model is trained by feeding the pre-processed dataset obtained in the Section 5.1. This preprocessed dataset contains key parameter indicators and weather conditions merged together which are the input features. There are two labels for classification which are given in Table 6. Each label can be classified into one of the two classes given in Table 6.

Train-test split of dataset

The merged dataset after preprocessing contains the parameters and the two labels mentioned in the *Labels and Classes* subsection. This merged dataset is split into train and test sets in the ratio of 7:3, i.e., 70% of samples in the train set and 30% of the samples in the test set. The number of samples present in the sets before and after splitting is given in Table 7.

Incorporating results from feature engineering

This paper compares the performance of the model with and without feature engineering present in the pipeline. Before the train-test split, the redundant features obtained after carrying out feature engineering are dropped from the merged preprocessed dataset. Following this, train-test split is done. This version of the sets is referred to as 'Feature Engineered Dataset' in this paper. Training is carried out twice for each label, once on the 'Merged Preprocessed Dataset' and once again on the 'Feature Engineered Dataset'. This is done in the next Section. Following that, results obtained for with and without feature engineering are compared in the Section.

Training using decision tree model

The decision tree model was trained for four different conditions (illustrated in Fig. 6). The name conventions in Fig. 6 will be used in the paper henceforth.

For training under all four conditions, the tuned hyper-parameters pertaining to the decision tree model are given in Table 8.

```
|--- feature_44 <= 0.50
|   |--- feature_20 <= -4.00
|   |   |--- class: 0.0
|   |--- feature_20 >  -4.00
|   |   |--- class: 0.0
|--- feature_44 >  0.50
|   |--- feature_17 <= 0.50
|   |   |--- feature_21 <= 68.50
|   |   |   |--- class: 0.0
|   |   |--- feature_21 >  68.50
|   |   |   |--- feature_23 <= 357.50
|   |   |   |   |--- feature_37 <= 15.30
|   |   |   |   |   |--- class: 0.0
|   |   |   |   |--- feature_37 >  15.30
|   |   |   |   |   |--- feature_3 <= 1.50
|   |   |   |   |   |   |--- class: 0.0
|   |   |   |   |   |--- feature_3 >  1.50
|   |   |   |   |   |   |--- feature_25 <= 4.50
|   |   |   |   |   |   |   |--- class: 1.0
|   |   |   |   |   |   |--- feature_25 >  4.50
|   |   |   |   |   |   |   |--- class: 1.0
|   |   |   |--- feature_23 >  357.50
|   |   |   |   |--- class: 0.0
|   |--- feature_17 >  0.50
|   |   |--- feature_38 <= 78.50
|   |   |   |--- feature_25 <= 7.50
|   |   |   |   |--- feature_47 <= 7.50
|   |   |   |   |   |--- feature_31 <= 0.50
|   |   |   |   |   |   |--- class: 0.0
|   |   |   |   |   |--- feature_31 >  0.50
|   |   |   |   |   |   |--- feature_21 <= 84.50
|   |   |   |   |   |   |   |--- class: 0.0
|   |   |   |   |   |   |--- feature_21 >  84.50
|   |   |   |   |   |   |   |--- class: 1.0
|   |   |   |   |--- feature_47 >  7.50
|   |   |   |   |   |--- class: 0.0
|   |   |   |--- feature_25 >  7.50
|   |   |   |   |--- feature_35 <= 0.50
|   |   |   |   |   |--- class: 0.0
|   |   |   |   |--- feature_35 >  0.50
|   |   |   |   |   |--- class: 0.0
|   |   |--- feature_38 >  78.50
|   |   |   |--- feature_39 <= -39.65
|   |   |   |   |--- feature_23 <= 196.50
|   |   |   |   |   |--- feature_2 <= 0.50
|   |   |   |   |   |   |--- feature_4 <= 337.50
|   |   |   |   |   |   |   |--- class: 1.0
|   |   |   |   |   |   |--- feature_4 >  337.50
|   |   |   |   |   |   |   |--- class: 0.0
|   |   |   |   |   |--- feature_2 >  0.50
|   |   |   |   |   |   |--- class: 0.0
|   |   |   |   |--- feature_23 >  196.50
|   |   |   |   |   |--- class: 0.0
|   |   |   |--- feature_39 >  -39.65
|   |   |   |   |--- class: 0.0
```

Fig. 7 – Rules defined after training the decision tree for W FE for 1 day

Table 8 – Tuned hyper-parameters of the decision tree model [17]

Hyper-parameter Name	Hyper-parameter Description	Value	Reason for choice of value
max_depth	Maximum depth of the tree	7	Chosen to be 7 and not higher in order to avoid overfitting
random_state	Controls the randomness of the estimator. The features are always randomly permuted at each split, even if the splitter is set to ”best”.	1555	To obtain a deterministic behaviour during fitting, *random_state* has to be fixed to an integer.
class_weight	It is a dictionary that defines each class label (e.g. 0 and 1) and the weighting to apply in the calculation of group purity for splits in the decision tree when fitting the model.	{0:0.01,1:0.99}	The dataset is unbalanced and the number of class 1 (occurrence of RLF) samples is low, while the number of class 0 (non-occurrence of RLF) samples are extremely high (the ratio is 0.00061). Hence, class weights of 0.01 and 0.99 for class 0 and class 1 respectively are chosen.

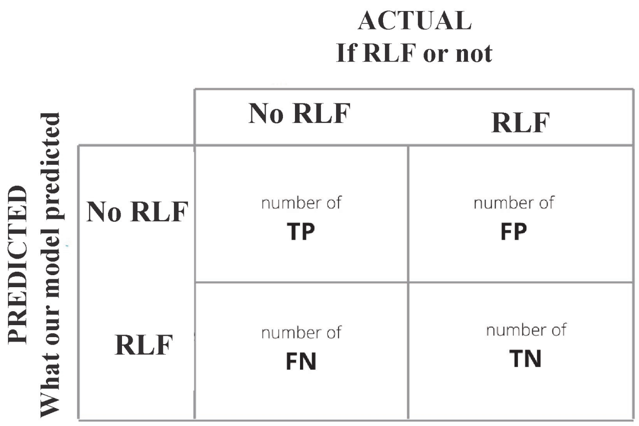

Fig. 8 – Confusion matrix

Training for all four conditions was carried out. Fig. 9 illustrates the schematic diagram of the decision tree obtained after training with feature engineering (W FE) for one day. The decision tree schematic has three main parts: root node, leaf nodes and branches. The root node is the starting point of the tree; both root and leaf nodes contain questions or criteria to be answered. Branches are arrows connecting nodes, showing the flow from question to answer. Each node typically has two or more nodes extending from it.

The legend shows that the colour of the schematic diagram's leaf nodes indicate the class to which it belongs along with the number of samples satisfying that branch rule. Blue indicates Class 1.0 (occurrence of RLF) and orange indicates Class 0.0 (absence of RLF).

Table 9 – Confusion matrix

Values	TP	FP
FN	129197	870
TN	6	304

Decision tree rules obtained after training for W FE for 1 day

The schematic diagram provided a visual representation of the decision tree. The rules defined after training the decision tree can be represented as in Fig. 7. It shows the sequence of rules used to decide the branches or paths chosen by each sample which finally results in the assignment of the class (0.0 or 1.0). Note that the schematic diagram and the rules for the trained decision tree can be obtained in the same method for the other three conditions as well.

6. RESULTS

The trained decision tree was validated using a test set. The classification report was generated after applying the trained decision tree model to the test set. This was carried out for all four conditions separately.

Components of the classification report

The confusion matrix is represented in terms of *True Positive, False Positive, False Negative and True Negative*, as seen in Fig. 8. The values in the confusion matrix are given in Table 9.

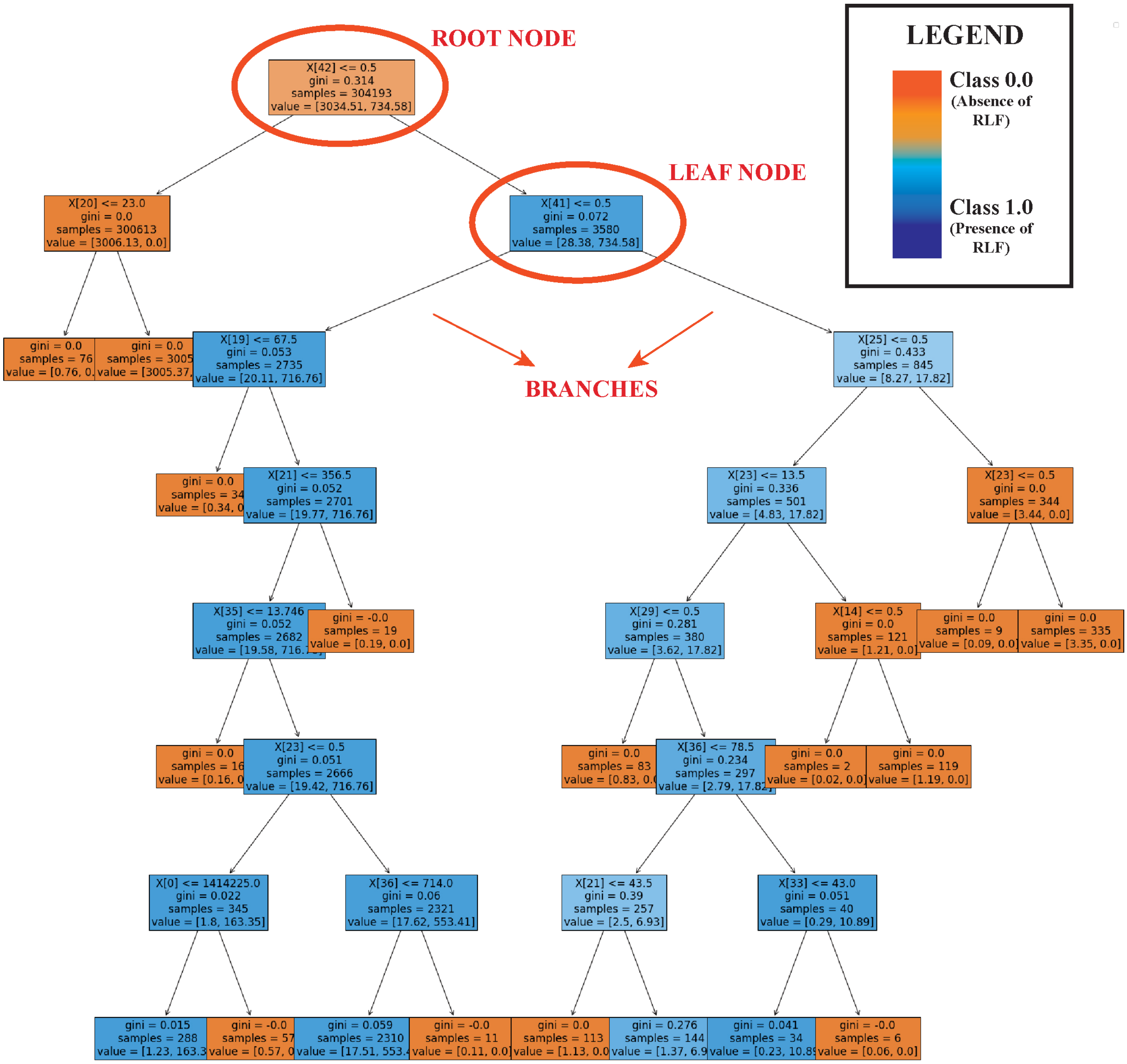

Fig. 9 – Schematic diagram of decision tree after training for W FE for 1 day

Table 10 – Classification report for W FE for 1 day

Class	Precision	Recall	F1-Score	Support
0.0	1.00	0.99	1.00	130067
1.0	0.26	0.98	0.41	310
Macro Average	0.63	0.99	0.70	130377
Weighted Average	1.00	0.99	1.00	130377

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive}$$

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative}$$

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall}$$

Formulae of metrics present in classification report [18][19]

Precision: Out of all the positives predicted, what percentage is truly positive.

Recall: Out of the total positives, what percentage are predicted positive. It is the same as True Positive Rate (TPR).

F1 - score: It is the harmonic mean of precision and recall. It takes both false positives and false negatives into account. Therefore, it performs well on an imbalanced dataset.

Support: Support is the number of actual occurrences of the class in the specified dataset. For example, there are 310 samples with class 1.0 (occurrence of RLF).

Macro-average: It is the simple mean of scores of all classes. So, macro-average precision is the mean of the precision of classes 0.0 and 1.0. In this case, it's the mean of 1.00 and 0.26 which gives 0.63 (as shown in the report in Table 10).

Weighted average: The sum of the scores of all classes after multiplying their respective class proportions. The weighted average obtained is high as it has the ability to account for an imbalance in data between the two labels.

The classification report for W FE for 1 day is given in Table 10. Here, the F1-score obtained for class 1.0 (presence of RLF) is 41%. The macro-averaged F1-score is 70%. The weighted average is high as it is able to balance the two labels. These results can be consolidated similarly for the other three conditions and are compared in the *Consolidated Results and Comparisons* subsection.

Additional evaluation metrics [20]

The dataset under consideration is highly imbalanced. Using accuracy (precision, recall and F1-score) does not truly represent the results after testing due to this imbalance. In the dataset, more than 90% of the samples belong to class 0.0 (majority class). A better-suited evaluation metric in order to get reliable results is the G-Mean metric [21]. Sensitivity refers to the true positive rate and summarizes how well the positive class was predicted. Specificity is the complement to sensitivity, or the true negative rate, and summarizes how well the negative class was predicted. Sensitivity and specificity can be combined into a single score that balances both concerns, called the geometric mean or G-Mean. The formulae for these metrics are:

$$Sensitivity = \frac{TruePositive}{TruePositive + FalseNegative}$$

$$Specificity = \frac{TrueNegative}{FalsePositive + TrueNegative}$$

$$G - Mean = \sqrt{Sensitivity * Specificity}$$

Table 12 shows the scores obtained for the additionally proposed metrics computed on the W FE for 1 day model.

Consolidated results and comparisons

The consolidated results for all four conditions are given in Table 11.

Upon comparing the results obtained for the four conditions, we can conclude that feature engineering enhances the performance of the trained model. The F1-score for class 1.0, macro-average F1-score, and G-Mean score show a significant increase when feature engineering is employed as compared to when it is not. Hence, it is evident that the preprocessing steps as well as the inclusion of feature engineering have helped the model perform efficiently and accurately.

7. CONCLUSION AND FUTURE WORK

This paper focused on improving 5G communication reliability by performing timely and accurate prediction of radio link failure in the next day or any of the next 5 days. The data was analyzed, preprocessed, merged and then trained using a decision tree. Results obtained by applying the trained decision tree to the test set have been discussed. The F1-scores obtained without feature engineering showed that there is scope to improve the accuracy and performance by introducing feature engineering in the pipeline. Feature engineering was successfully implemented and increased the F1-scores significantly. Further, the metrics used to analyze the results were extended. Another metric called the G-Mean was introduced in order to account for the considerable imbalance present in the dataset. G-Mean

Table 11 – Consolidated results and comparisons for the four conditions

Condition Name	**F1 Score for Class 0.0 (%)**	**F1-Score for Class 1.0 (%)**	**Macro Average - F1-Score (%)**	**Weighted Average - F1-Score (%)**	**G-Mean Score (%)**
W/O FE for 1 DAY	100	39	69	99	98.35
W FE for 1 DAY	100	41	70	100	98.69
W/O FE for 5 DAYS	99	43	71	98	92.71
W FE for 5 DAYS	99	55	77	99	92.89

Table 12 – Scores obtained for the additionally proposed metrics computed on the W FE for 1 day model

Metric Name	**Score (%)**
Sensitivity	99.33
Specificity	98.06
G-Mean	98.69

was used to provide a more realistic and compatible evaluation score. In this paper, a novel pipeline has been proposed and successfully implemented. The main components making this novel pipeline are:

- understanding the key performance indicators and their impact on radio links;
- systematic preprocessing to combine multiple datasets efficiently;
- handling an imbalanced and high-dimensional dataset;
- incorporating feature engineering to improve accuracy;
- introducing G-Mean as a metric for the highly imbalanced dataset.

As part of future work, other machine learning and deep learning models can be explored for modeling the dataset. The radio link failure prediction could be extended to predict the possibility of failure for a larger number of days (more than five days) or the frequency of prediction can be increased so that prediction is done twice or thrice in a single day.

REFERENCES

[1] Dibakar Das and S. Saha. "Evaluation of mobile handset recovery from radio link failure in a multi-RATS environment". In: Jan. 2009, pp. 1–6. DOI: 10.1109/IMSAA.2008.4753932.

[2] Ibraheem Shayea, Liyth Nissirat, mahdi nisirat mahdi, Aida Alsamawi, Tharek Rahman, Marwan Bin Azmi, Mohammad Abo-Zeed, and Issam Trrad. "Rain attenuation and worst month statistics verification and modeling for 5G radio link system at 26 GHz in Malaysia". In: *Transactions on Emerging Telecommunications Technologies* 30 (Dec. 2019). DOI: 10.1002/ett.3697.

[3] Yi-Wei Mal, Jiann-Liang Chen, and Hao-Kai Lin. "Mobility Robustness Optimization Based on Radio Link Failure Prediction". In: *2018 Tenth International Conference on Ubiquitous and Future Networks (ICUFN)*. 2018, pp. 454–457. DOI: 10.1109/ICUFN.2018.8436964.

[4] Sebastián Gutiérrez and Hiram Ponce. "An Intelligent Failure Detection on a Wireless Sensor Network for Indoor Climate Conditions". In: *Sensors* 19 (Feb. 2019), p. 854. DOI: 10.3390/s19040854.

[5] Jyotirmoy Karjee, Rajan Garg, Vaishal Tijoriwala, and Ashok kumar Reddy Chavva. "A Reinforcement Learning Approach to Handle Radio Link Failure in Elevator Scenario". In: *2020 IEEE 17th Annual Consumer Communications Networking Conference (CCNC)*. 2020, pp. 1–6. DOI: 10.1109/CCNC46108.2020.9045193.

[6] Wangyang Shi and Linlin Wu. "The Study of Application of Multi-label Classification Methods to Type Selection of Mobile Base-Station". In: *2012 Fourth International Conference on Computational and Information Sciences*. 2012, pp. 294–297. DOI: 10.1109/ICCIS.2012.352.

[7] Wei Bi and James T. Kwok. "Efficient Multi-Label Classification with Many Labels". In: *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*. ICML'13. Atlanta, GA, USA: JMLR.org, 2013, pp. III–405–III–413.

[8] Ritu Aggrawal and Saurabh Pal. "Elimination and Backward Selection of Features (P-Value Technique) In Prediction of Heart Disease by Using

Machine Learning Algorithms". In: *Turkish Journal of Computer and Mathematics Education (TURCOMAT)* 12 (Apr. 2021), pp. 2650–2665. DOI: 10.17762/turcomat.v12i6.5765.

[9] Wencong Qin, Yinglei Teng, Yi Man, Shuai Yu, and Yinghai Zhang. "A Detection Method for Handover-Related Radio Link Failures Based on SVM". In: Jan. 2014, pp. 476–486. ISBN: 978-3-319-09264-5. DOI: 10.1007/978-3-319-09265-2_49.

[10] A L Tom and Srinivasan Vasudevan. "Predicting Cellular Link Failures to Improve User Experience on Smartphones". In: 2021.

[11] Ahmed Masri, Teemu Veijalainen, Henrik Martikainen, Stephen Mwanje, Janne Ali-Tolppa, and Márton Kajó. "Machine-Learning-Based Predictive Handover". In: *2021 IFIP/IEEE International Symposium on Integrated Network Management (IM)*. 2021, pp. 648–652.

[12] *ITU AI/ML in 5G Challenge*. URL: https://aiforgood.itu.int/about-ai-for-good/aiml-in-5g-challenge/.

[13] P. Yadav. *"Decision Tree in Machine Learning", Towards Data Science (Online Resource)*. 2021. URL: https://towardsdatascience.com/decision-tree-in-machine-learning-e380942a4c96.

[14] C. Liu. *"Decision Tree Intuition: From Concept to Application" (Online Resource)*. 2021. URL: https://www.vebuso.com/2020/01/decision-tree-intuition-from-concept-to-application/.

[15] *"Decision Tree Classification Algorithm", Javatpoint (Online Resource)*. 2021. URL: https://www.javatpoint.com/machine-learning-decision-tree-classification-algorithm.

[16] Kedar Potdar, Taher Pardawala, and Chinmay Pai. "A Comparative Study of Categorical Variable Encoding Techniques for Neural Network Classifiers". In: *International Journal of Computer Applications* 175 (Oct. 2017), pp. 7–9. DOI: 10.5120/ijca2017915495.

[17] *"Decision Tree Classifier", Scikit Learn (Online Resource)*. 2021. URL: https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html.

[18] K. P. Shung. *"Accuracy, Precision, Recall or F1?", Towards Data Science*. 2018.

[19] *Churning the Confusion out of the Confusion Matrix", Medium (Online Resource)*. 2019.

[20] J. Brownlee. *"Tour of Evaluation Metrics for Imbalanced Classification", Machine Learning Mystery*. 2020. URL: https://machinelearningmastery.com/tour-of-evaluation-metrics-for-imbalanced-classification/.

[21] J. S. Akosa. *"Predictive Accuracy: A Misleading Performance Measure for Highly Imbalanced Data"*. 2017.

AUTHORS

Nethraa Sivakumar is a B.E student in electronics and communication engineering from SSN College of Engineering (2018-2022). She has accepted a position as a Member of Technical Staff (Incoming) at Adobe Systems, India. She has worked on research projects in the field of statistics, artificial intelligence and fuzzy logic. Currently, her research focuses on the application of natural language processing in a multilingual, knowledge graph based question answer system to assist farmers.

Pooja Srinivasan is a B.E student in electronics and communication engineering from SSN College of Engineering (2018-2022). She currently holds positions as chair of IEEE communications society chapter and treasurer of IEEE student chapter of her college. Her research works and interests are in the field of machine learning, natural language processing, statistics and fuzzy logic systems. She has also worked at Western University, Canada as a MITACS Globalink Research Internship 2021 scholar and will be pursuing her Master of Science in Data Science at Columbia University, USA.

Nikhil Viswanath is a B.E. student in electronics and communication engineering at Sri Sivasubramaniya Nadar College of Engineering, India (2018-2022). He currently holds the position of Student President of his college's department association. As an aspiring Signal Processing Engineer and a Machine Learning enthusiast, he will be pursuing his graduate studies majoring in the above-mentioned fields, at Georgia Institute of Technology. He pursued a Project-cum-research internship at Acrophase, a startup under the Healthcare Technology Innovation Centre of Indian Institute of Technology, Madras. Currently, he is working as a research associate for a healthcare start-up. Other research interests of his include speech technology, image processing, wearable technology for athletes, computer vision, and signal processing for wireless communication.

Dr. N. Venkateswaran, obtained his PhD from Anna University, Chennai in information and communication engineering for his thesis in the area image compression. He is a Fellow of IETE, Senior member of IEEE and Life member of ISTE. He is currently working in ECE department of Sri Sivasubramaniya Nadar college of Engineering Chennai. Earlier to this current position, he was a professor at Sri Venkateswara College of Engineering, Sriperumbudur. He has more than 23 years of experience in teaching, research and in addition 8 years of experience in industry. He has authored or co-authored over 100 papers in national and international journals and conferences. He has graduated 8 Ph.D. students. Also, he has supervised more than 30 Master's thesis. His area of interest are image and signal processing, machine learning, wireless communication and optimization techniques.