Computer Networks and Communications
https://ojs.wiserpub.com/index.php/CNC

UNIVERSAL WISER PUBLISHER

Research Article

Mutual Exploration for Missing Data Imputation, QoS Parameter Selection, and QoS Prediction in 5G Networks Using a Novel Skewness Driven Distribution Imputation Algorithm, Pearson Correlation, and XGBoost

Saifullah Khan[1], **Onel Luis Alcaraz López**[2], **Abdul Basit Khattak**[2*]

[1] Department of Electrical and Computer Engineering, COMSATS University Islamabad, Attock Campus, Islamabad, Pakistan
[2] Centre for Wireless Communications (CWC), University of Oulu, Oulu, Finland
E-mail: Abdul.khattak@oulu.fi

Received: 20 August 2024; **Revised:** 8 November 2024; **Accepted:** 19 November 2024

Abstract: Pre-processing is a key stage in the Machine Learning (ML) pipeline. In such a stage, data is prepared and organized for feeding it to the ML models for a prediction task. One of the problems that might incur in this stage is that data may have missing values, requiring that either the data is deleted or imputed with data points that resembles/ correlate with the original one. Imputation is desirable, as having more data to be fed to ML models means the model will have more context and thus better prediction results. From an imputation perspective, since the goal is that the imputed data faithfully relates with the original data, the result of correlation metric is desirable to show that the imputed data and original data are closely correlated. Herein, we present a novel imputation algorithm of Skewness Driven Distribution Imputation (SDDI), and evaluate its efficacy compared to multiple state-of-the-art methods including, K-Nearest Neighbors (KNN), Mean, Mode, Forward and Backward Fill (F&B-Fill) imputation methods. The comparison is done using accuracy metrics, Root Mean Square Error (RMSE), correlation and computation time. Furthermore, a correlation analysis is conducted on the subject 5G Vehicle-to-Everything (V2X) Quality of Service (QoS) dataset, aiming to enhance the understanding of parameter selection in assessing the QoS for 5G networks by providing the comparison regarding the significance in terms of correlation of various parameters in influencing 5G network's QoS. The state-of-the-art Pearson correlation method is used for said purpose. Moreover, we exploit an Extreme Gradient Boosting or XGBoost algorithm which is an ensemble of other techniques and not as complex as deep learning algorithms, to predict QoS, given certain conditions relevant to the 5G network. The comparative analysis of various imputation methods revealed average correlation values (as a measure of faithful data imputation) to be relatively close for Mean, Mode, F&B-Fill, SDDI, and K-Nearest Neighbors imputation methods at 0.161, 0.176, 0.143, 0.143, and 0.196, respectively. In terms of accuracy, all methods achieved high rates, with Mean and Mode at 93%, F&B-Fill at 90%, and both SDDI and KNN at 92%. Notably, in the second part of the research, when only the 15 most correlated features were used, we observed a substantial 60.5% reduction in the amount of data affected, with only a minimal impact of 3% on accuracy, achieving an impressive 93% accuracy. These results highlight the effectiveness of targeted feature selection of parameters for QoS of 5G networks and underscore the potential of our novel SDDI method in maintaining high data integrity while efficiently handling missing data, thereby enhancing the predictive reliability of the XGBoost algorithm.

***Keywords*:** skewness driven distribution imputation, vehicle to everything network, 5G QoS, machine learning

DOI: https://doi.org/10.37256/cnc.2220245534

Nomenclature

eMBB	Enhanced Mobile Broadband
F&B-Fill	Forward and Backward Fill
KNN	K-Nearest Neighbors
ML	Machine Learning
mMTC	Massive Machine-Type Communications
MSE	Mean Squared Error
NA	Not Available
QoS	Quality of Service
RMSE	Root Mean Squared Error
SDDI	Skewness Driven Distribution Imputation
URLLC	Ultra-Reliable and Low Latency
V2X	Vehicle-to-Everything
XGBoost	Extreme Gradient Boosting

1. Introduction

Data imputation is a critical step in the pre-processing phase of a Data Science or Machine Learning (ML) workflow [1], taking 80% time of the whole ML project [2]. The fundamental motivation for data imputation is that many models cannot handle datasets containing missing values [3], e.g., the continuity of a time series is affected by missing values since date and time would be missing [4]. Among various techniques used for filling the missing values of data, Mean imputation [5] involves inserting the mean value throughout the dataset at missing positions, while Mode imputation uses the mode of the data for the same purpose. In certain cases, deletion of missing parts may be appropriate for datasets with less than 5% missing values [6], contingent on factors like resultant accuracy and project objectives. However, when dealing with a substantial percentage of missing data, ML algorithms such as K-Nearest Neighbors (KNN) [7] or Decision Trees [8] are often employed, though this introduces complexity and increases computational processing time compared to statistical methods like mean/mode imputation. Additionally, when examining scientific trends over time, it is apparent that the domain of data imputation is often overshadowed by the flourishing interest in new models pertaining to deep learning and, more broadly, machine learning. This trend is highlighted in [9], where Google Trends analysis for the period from November 2023 to November 2024 shows that searches related to data imputation and associated keywords average at 8 points, whereas searches for machine learning models peak at 83 points. While the development of new models is crucial, the significant portion of time (80%) spent on preprocessing in an ML project suggests that this area also deserves considerable attention. Typically, the imputation techniques employed range from basic and less computationally intensive methods such as mean or mode imputation and outright data deletion, to more advanced, but resource heavy deep learning based models. For instance, recent studies like [10] have explored complex imputation methods such as the Bi-ConvRNN, which combines Convolutional Neural Networks (CNN) and Bidirectional Recurrent Neural Networks (Bi-RNN), or [11], which employs Conditional Generative Adversarial Networks (cGANs) for addressing imbalances and incompleteness in time-series datasets. The statistical-based imputation methods serves as a middle ground, enhancing efficiency without incurring high computational costs, and is the motivation for this research to contribute to such underrated yet essential domain of imputation in ML pipeline.

This article introduces Skewness Driven Distribution Imputation (SDDI), an innovative algorithm crafted to tackle the challenge of integrating natural distribution concept of data into imputation process, avoiding the intricacies associated with ML methods. In contrast to traditional mean/mode imputation approaches, SDDI takes into account the distribution characteristics of the dataset while maintaining a complexity level comparable to statistical techniques. While the primary goal of this novel algorithm is to offer flexibility and comparable outcomes, it's crucial to emphasize that SDDI is not alternative to ML algorithms. Instead, it presents an additional streamlined option for data scientists and researchers to impute and enhance results without delving into the complexities of intricate ML workflows. This enables the utilization of the traditional and simple mean/mode imputations while effectively addressing imputation in the dataset through the statistical consideration of distribution characteristics.

Additionally, this research involves correlation analyses on a 5G network Quality of Service (QoS) dataset. Firstly, subject correlation analysis assesses the correlation between the imputed data generated by the SDDI algorithm and the original dataset, and subsequent comparisons with other state-of-the-art methods. Secondly, this correlation analysis contribute to showing the significance of each of the input parameters that effects the QoS of a 5G network.

The fifth-generation cellular communication standard, known as 5G, offers significant broadcasting capacity of up to Gigabit units, accommodating up to 65,000 connections simultaneously [12]. Additionally, 5G technology enables the use of a single universal device, providing access to various cellular technologies simultaneously based on the desired application [12]. From a technical standpoint, 5G technology encompasses three distinct scenarios [13]: (i) Enhanced Mobile Broadband (eMBB), characterized by high-speed communication and improved performance compared to earlier technologies; (ii) Ultra-Reliable and Low Latency (URLLC), dedicated to minimal latency, real-time and reliable communication applications; (iii) Massive Machine-Type Communications (mMTC), prioritizing connectivity between humans and machines, as well as inter-machine communication [14].

In this paper, the utilized data originate from Vehicle-to-Everything (V2X) network measurements and pertain to the URLLC domain. We focus on the QoS dimension of 5G, a critical factor whose determination can contribute to not only the dynamic allocation of network resources based on QoS conditions but also define the expectations of network's service providers, and customers [15], since, unlike 4G, the 5G network is characterized by those feature elements which are distinct from traditional QoS network parameters. These elements are related to user satisfaction, aspects such as video buffering and latency in online gaming etc. [16]. This study contributes in identifying the relevance of 5G QoS parameters for V2X by examining the correlation between various factors. The analysis includes dataset with parameters related not only to network architecture but also to location and physical scenarios, ultimately influencing the determination of QoS [17]. Correlation analysis is employed to identify the parameters that significantly influence downlink throughput, and thus can help researchers and industrial players to focus on these in future studies. Consequently, there will be savings in time, logistical expenses associated with sensor and equipment measurements, computation costs, and overall efficiency in both the measurement and QoS determination processes. Though, the downlink data rate is emphasized, the input features, includes factors that affect traffic congestion, reliability and strength of the signal, which are all very important in determining the QoS and will be discussed later in Section 2 [18].

This research utilizes an ensemble of low complex techniques, i.e., Extreme Gradient Boosting (XGBoost) ML algorithm to predict the QoS in terms of throughput of the 5G V2X network. This specific model was selected for this research due to several considerations. Advanced methods like deep learning algorithms have been applied in similar studies, such as in [19]; where cascading Neural Networks (NN) were used instead of a single fully-connected NN such that one part of the cascade finds optimal Bandwidth allocation and the second cascade finds the transmit power for optimal 5G QoS; and in [20], where a sophisticated resource allocation architecture for TV multimedia services in the 5G wireless cloud network (C-RAN) is developed, employing a combination of LSTM (Long Short-Term Memory) for dynamic traffic modeling and deep reinforcement learning with convex optimization for minimizing energy usage in Remote Radio Heads (RRHs) while adhering to QoS constraints. However, the preference for ML algorithms arises from their faster processing speed and lower resource requirements [21]. In particular, the use of XGBoost, an ensemble of computationally inexpensive models [22], aligns with the subject research goal of achieving accurate predictions with minimal computational resources.

It should be emphasized that the central focus of this research is primarily on proposing and evaluating the novel SDDI algorithm, comparing its effectiveness with other commonly used imputation methods. Consequently, it was deemed more appropriate to assess the outcomes of each imputation technique using a single model with the same configuration for all such techniques, XGBoost, as a representative use case. There were various other options of using ML models, including promising latest research of [23], where a novel approach of NNBoost was proposed for the application of network management in Software Defined Networking and it outperformed methods like XGBoost, LSTM, Random Forest etc. Another interesting novel research [24], proposes CLpred paradigm which uses Contrast Learning approach to implement a transformer encoder based architecture to infer QoS data time-awarely, with implications of data augmentation application as well.

However, the focus of this research is on the imputation domain and evaluating imputation methods, which differs from the objectives of NNBoost and CLpred that target enhancements in network and data prediction architectures. Consequently, XGBoost serves as a consistent use-case to evaluate subject research's employed imputation methods.

The key contributions of this work are three-fold:

- Introduction of the SDDI algorithm for imputation of missing data in the pre-processing stage of an ML workflow. The novel SSDI utilizes the distribution feature of skewness, of the analyzed dataset, and then imputes missing data according to the identified distribution.
- The comparison of SDDI with state-of-the-art Mean, Mode, KNN and Forward and Backward Fill (F&B-Fill) imputation techniques in terms of correlation, Root Mean Square Error (RMSE), Accuracy and computation time of imputation has been implemented with SDDI getting 0.0499 RMSE, 0.143 correlation, 92% accuracy and 10 s in computation time for execution. The results corroborate that SDDI offers comparable results with respect to the counterparts.
- Correlation analysis of input parameters/features of subject data set and consequent conclusion of which parameters holds relevance, subsequently helping future V2X QoS determination processes to only consider the meaningful features for measurement and dynamic resource allocation.

The paper is organized as follows. In Section 2, an overview of the pre-processing procedures is outlined. Furthermore, a comprehensive overview is provided for the XGBoost ML algorithm employed in the study, along with the implementation of Pearson correlation and the assessment metrics of Accuracy and RMSE utilized. In Section 3, numerical experimentations along with their respective interpretations are shown. Lastly, the inferences of the research are summarized in Section 4.

2. Methodology

Herein, we first present the dataset and pre-processing methods, specifically related to the imputation, adopted in this research work. Then, we discuss the implementation of Pearson correlation method and the ML algorithm of XGBoost. Finally, we discuss the evaluation metrics used to evaluate the model's performance.

Figure 1 gives an overview of the proposed research methodology. More specifically, it illustrates the methodology employed in our study to analyze 5G Quality of Service (QoS) using V2X (Vehicle to Everything) data from Berlin which features 37 input variables and downlink throughput as the target variable.

In the preprocessing phase, two different paths are taken for achieving two different objectives of this research. The path where correlations study takes place (orange arrows) and non-correlated features are dropped (features with zero correlation) and further passed through the imputation stage where five imputation techniques are applied, has the objective of evaluating the novel SDDI technique in terms of how much faithful reproduction of missing data has resulted in (evaluated by average correlation) and how such imputation leads to effect accuracy when an ML algorithm is applied (in this case XGBoost applied and evaluated using RMSE, MSE, accuracy and compute time). This path is basically used to evaluate the efficacy of novel SDDI compared to other existing methods.

The secondary route (green arrows) aims to identify the essential input features necessary for general 5G QoS assessment. By isolating only the correlated features, our methodology can reduce costs and save time by suggesting that future research focus solely on these critical features. This streamlined approach ensures that only relevant data is collected and utilized, saving costs and time in terms of sensor deployments for recording data of the non-essential parameters. This is specifically done here by dropping weather related features and features with less than equal to 0.1 correlation (only 15 input features remain out of original 37, after dropping low correlated features). To test whether the choice of dropping criteria is suitable and low correlated features indeed had negligible impact on the final outcome of QoS prediction, the data is passed through ML implementation (in this case XGBoost) and results are evaluated via RMSE, MSE, and accuracy. The results are elaborated with interesting outcomes in Section 3.

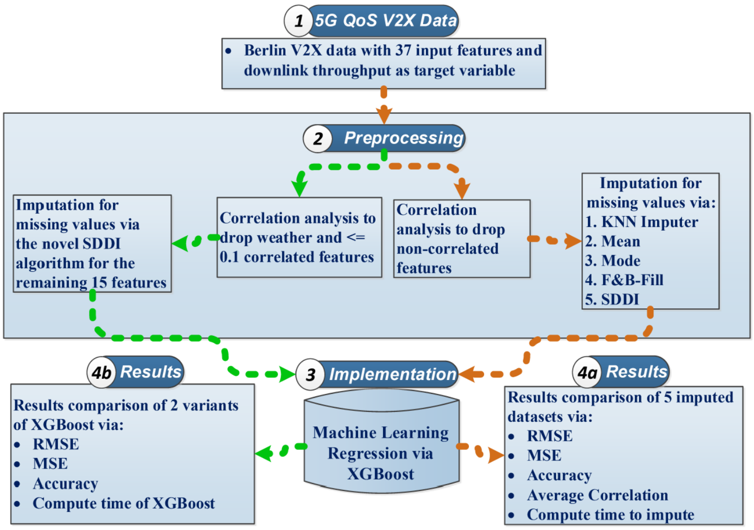
1 5G QoS V2X Data
Berlin V2X data with 37 input features and downlink throughput as target variable
2 Preprocessing
Imputation for missing values via the novel SDDI algorithm for the remaining 15 features
Correlation analysis to drop weather and <= 0.1 correlated features
Correlation analysis to drop non-correlated features
Imputation for missing values via: 1. KNN Imputer 2. Mean 3. Mode 4. F&B-Fill 5. SDDI
4b Results
Results comparison of 2 variants of XGBoost via: RMSE MSE Accuracy Compute time of XGBoost
3 Implementation
Machine Learning Regression via XGBoost
4a Results
Results comparison of 5 imputed datasets via: RMSE MSE Accuracy Average Correlation Compute time to impute

Figure 1. Proposed methodology of the subject research

2.1 *Pre-processing*

The 5G V2X dataset [25] corresponds to the city of Berlin, Germany and includes 37 input features and 1 target variable with a total of 34,274 data points or rows for each of the 38 total columns. The data had a total 131,506 missing data points from all the columns. A short description for each column is given in Table 1. The investigation employs identical column names as those present in the original dataset for ease of reference, and these names will be consistently used to denote these columns in subsequent discussions. It is to be noted that the study further includes research in the scenario of dropping parameters related to weather, and less than equal to 0.1 correlation features (as shown in Table 2 with further results oriented insights given in Section 3.2). Hence, the resultant scenario is using parameters that not only employs to V2X but to 5G network in general. Therefore, any conclusions of this study are usable in terms of 5G QoS description as well.

Table 1. Short description of the subject dataset

Input parameters	Description
PCell_RSRP_max	Maximum reference signal received power in dBm from the primary cell
PCell_RSRQ_max	Maximum reference signal received quality in dB from the primary cell
PCell_RSSI_max	Maximum received signal strength indicator in dBm from the primary cell
PCell_SNR_max	Maximum signal to noise ratio in dB from the primary cell
PCell_Downlink_Num_RBs	Aggregated number of received resource blocks in downlink from the primary cell
PCell_Downlink_TB_Size	Aggregated transport block size in downlink from the primary cell
PCell_Downlink_Average_MCS	Average modulation and coding scheme weighted by received resource blocks from the primary cell
PCell_Downlink_bandwidth_MHz	Downlink bandwidth in MHz from the primary cell
PCell_Cell_Identity	Cell ID of the primary cell
PCell_freq_MHz	Carrier frequency in MHz of the primary cell
SCell_RSRP_max	Maximum reference signal received power in dBm from the secondary cell
SCell_RSRQ_max	Maximum reference signal received quality in dB from the secondary cell
SCell_RSSI_max	Maximum received signal strength indicator in dBm from the secondary cell
SCell_SNR_max	Maximum signal to noise ratio in dB from the secondary cell
SCell_Downlink_Num_RBs	Aggregated number of received resource blocks in downlink from the secondary cell
SCell_Downlink_TB_Size	Aggregated transport block size in downlink from the secondary cell
SCell_Downlink_Average_MCS	Average modulation and coding scheme weighted by received resource blocks from the secondary cell
SCell_Downlink_bandwidth_MHz	Downlink bandwidth in MHz from the secondary cell
SCell_Cell_Identity	Cell ID of the secondary cell
SCell_freq_MHz	Carrier frequency in MHz of the secondary cell
operator	ID of the operator
Latitude	GPS latitude of the vehicle
Longitude	GPS longitude of the vehicle
Altitude	GPS altitude of the vehicle

Table 1. *Cont.*

Input parameters	Description
speed_kmh	GPS-based speed of the vehicle in km/h
COG	GPS-based course over ground of the vehicle
precipIntensity	Precipitation intensity
precipProbability	Precipitation probability
temperature	Temperature
apparentTemperature	Apparent temperature
dewPoint	Dew point
humidity	Humidity
pressure	Atmospheric pressure
windSpeed	Wind speed
cloudCover	Cloud cover
uvIndex	Ultraviolet index
visibility	Visibility due to weather conditions
Traffic Jam Factor	Factor of traffic congestion (a higher factor corresponds to heavier traffic)
area	Categorical urban area among Avenue/Park/Residential/Highway/Tunnel
target	Prediction target: downlink data rate in bits per second

Table 2. Parameters related to Weather and less than equal to 0.1 correlation

Parameters	Parameters
SCell_Downlink_bandwidth_MHz	apparentTemperature
SCell_freq_MHz	dewPoint
Latitude	humidity
Longitude	pressure
Altitude	windSpeed
speed_kmh	cloudCover
COG	uvIndex
precipIntensity	visibility
precipProbability	Traffic Jam Factor
temperature	area

2.2 *Adopted imputation algorithms*

In this research work, the novel method of Skewness Driven Distribution Imputation (SDDI), the KNN imputer, Mean, Mode, and Forward and Backward Fill (F&B-Fill) imputation methods are used for filling any missing values in the original dataset. These are explained in subsequent sub-sections.

2.2.1 *Skewness driven distribution imputation (SDDI)*

The SDDI has mainly 3 components: Skewness, Normal Distribution, and Gamma Distribution, which are explained below.

Skewness is a measure of the asymmetry of a probability distribution. It quantifies the extent and direction of skew (departure from horizontal symmetry) in the data. A positive skewness indicates a longer right tail, while negative skewness indicates a longer left tail. Mathematically, skewness is calculated as

$$Skewness = \frac{\sum_{i=1}^{N} (X_i - \overline{X})^3}{(N-1)\ \sigma^3} \tag{1}$$

where N is number of variables in the distribution, X_i is the i-th data point, $\overline{X}$ is the sample mean of the distribution and σ is the sample standard deviation.

A *Normal Distribution*, is symmetric around its mean, forming a bell-shaped curve. It is fully characterized by its mean and standard deviation. The code uses the *numpy.random.normal* function to generate random numbers from a normal distribution with a mean and standard deviation specific to the subject data calculated by *numpy.log(data_column.mean())* and *numpy.log(data_column.std())* functions, respectively. Mathematically, the probability density function (PDF) of the normal distribution is given by

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{x-\overline{X}^2}{2\sigma^2}} \tag{2}$$

The 99.7% of all the values lie within the first 3 standard deviations as seen in the area under the curve of the graph in Figure 2.

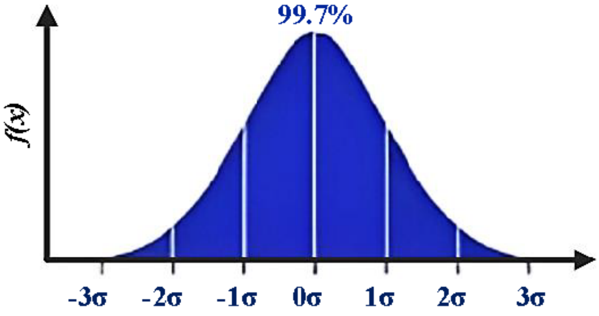

Figure 2. PDF of a normal distribution

A *Gamma Distribution* is used in this research for both right and left skewed data. Mathematically, the PDF of the gamma distribution is given by

$$f(x) = \frac{1}{\Gamma(k)\theta^k} x^{k-1} e^{-\frac{x}{\theta}} \tag{3}$$

where, k denotes shape and influences the form of the probability distribution, e.g., higher k would give more peaked and less skewed shape), θ is the scale and captures the compression or stretch of distribution, e.g., larger θ values gives wider distribution), and $\Gamma(k)$ is the gamma function.

Figure 3 illustrates the shape of data according to 3 types of distribution. The algorithm of SDDI checks for skewness of data. If the data is skewed, returning either left skewed or a right skewed distribution, the Gamma Distribution is fitted on the data based on data's distribution's shape, scale and location parameters. Then, using its PDF, a pool of data points are formed from using the gamma distribution model as described in Equation (3). These generated data points are equal to the number of data points as in the original dataset and have a gamma distribution (in case the original data was skewed). As a result, at places where the original dataset have data missing, the gamma distributed generated values are imputed. Alternatively, if the data is normally distributed, then using the PDF of Normal Distribution, a pool of data points that adheres to this distribution are formed for imputing missing values. The values from the Normal Distribution or Gamma Distribution ensures that the generated values align with the natural distribution of that column since data's own statistical parameters like mean, skewness, shape, location, standard deviation and scale parameters has been used. This means that even in the presence of missing values, the filled values from the distribution obtained will closely resemble the overall natural distribution of the column.

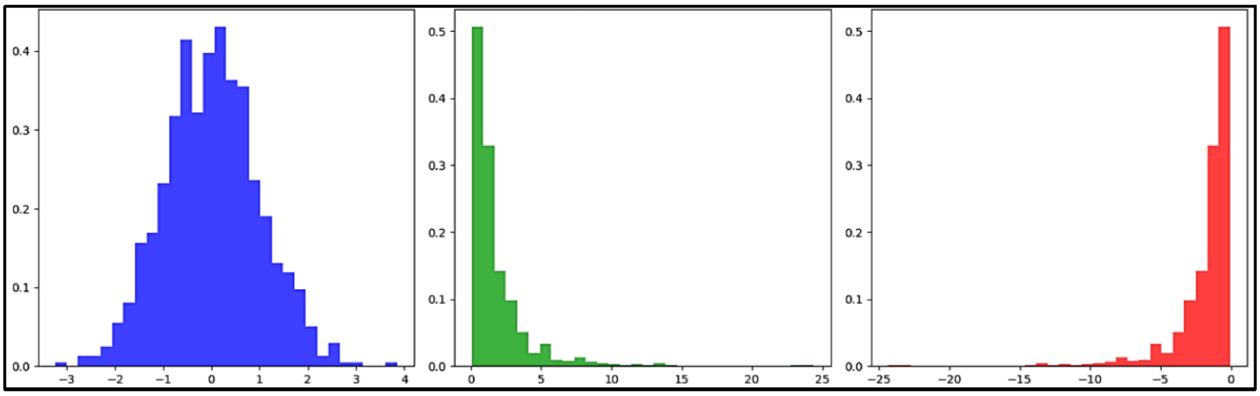

Figure 3. An illustration of normal distribution (No skewness, left image), right skewed (middle image), and left skewed (right image)

It is of importance to note that this method can be used in time-series data. Its non-seasonal counterparts Mean and Mode imputation [26, 27] can be used as well but their usage depend on the use-case, specifically the computation time, complexity requirements of a project and the amount of prediction accuracy needed. In general, imputation methods, including Seasonal Mean Imputation [28], are better suited for time-series data. This research involves the use of seasonal methods of ML based KNN imputation and statistical method of F&B-Fill for comparison to the SDDI, in addition to non-seasonal methods of Mean and Mode imputation such that the SDDI is evaluated with both seasonal and non-seasonal state-of-the-art. This comprehensive evaluation allows for a robust comparison of SDDI against a wide range of techniques when applied to non-time series data.

The pseudo code in Table 3 examines the skewness of columns containing missing values in the dataset. If the skewness exceeds 0.05 or below -0.05, it fits a gamma distribution to the non-null values of the column using parameters of *a_fit* (shape), *loc_fit* (location), and *scale_fit* (scale). The code creates a linear space (x) spanning the range of the column values and calculates the PDF of the fitted gamma distribution at each point in the linear space. Imputed values are generated by randomly sampling from the fitted PDF. Finally, the missing values in the original column are replaced with the generated imputed values, aligning the imputation with the natural distribution characteristics inferred from the fitted gamma distribution. At last, if the skewness is in the range of -0.05 to 0.05, the normal distribution of the particular column is generated.

Table 3. Pseudo code for SDDI implementation

```
for each column col in df_Train.columns:
    if df_Train[col] has missing values:
        calculate skewness = skew(dropna values of df_Train[col])
        if absolute value of skewness > 0.05:
            # Impute using Gamma Distribution
            [a_fit, loc_fit, scale_fit] = fit_gamma_distribution(dropna values of df_Train[col])
            x = create_linear_space(minimum value of df_Train[col], maximum value of df_Train[col], length of df_Train[col])
            calculate PDF values using gamma distribution parameters (x, a_fit, loc=loc_fit, scale=scale_fit)
            generate imputed values using random choice and PDF values
            fill missing data points in df_Train[col] with imputed values
        else:
            # Impute using Normal Distribution
            generate random seed using default random number generator
            calculate mean mu and standard deviation sd of df_Train[col]
            create filler values from normal distribution (size=len(df_Train))
            fill missing data points in df_Train[col] with filler values
```

2.2.2 *Mean imputation*

Taking a simple average or mean of the whole data as shown in the Figure 4A (gives an example of data named *x*), and then using the resultant value to fill any NA (not available) values in the data, is defined as Mean imputation.

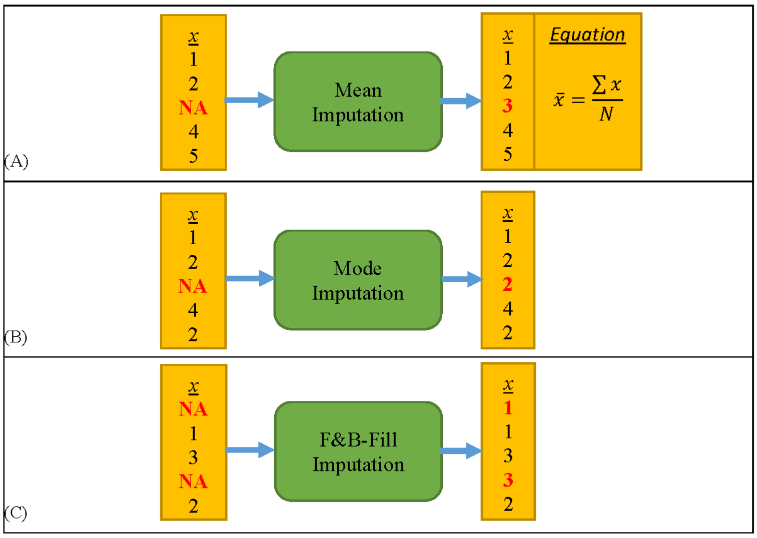

Figure 4. Example of mean, mode, and F&B-Fill imputation working

2.2.3 *Mode imputation*

Taking a value repeated most time in the whole data as shown in the Figure 4B, and then using the resultant value to fill any NA (not available) values in the data, is defined as Mode imputation. However, discrete datasets having repeated values are compatible with this method, such as the subject data utilized in this research.

2.2.4 *Forward and backward fill imputation*

Forward filling involves filling a missing value in a dataset by taking the nearest non-missing value that precedes it. This method utilizes the value immediately before the target missing value to populate it. While, backward filling addresses the gaps left by forward filling, particularly when data points are positioned at the extremes of the dataset. In backward filling, missing values are filled by using the nearest non-missing values that succeed the target missing values. The Figure 4C shows first NA value backward filled while the second NA value forward filled.

2.2.5 *K-nearest neighbors imputation*

In KNN, the label or parameter for which a result is required and the variables or features or input data points that effect such a label are considered on a 2D plane. Depending on their values, the features are scattered over the plane. The algorithm works by calculating distance of label with each feature points through Euclidean distance as

$$Euclidean\ distance(a,b) = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2} \tag{4}$$

As represented in Figure 5, the features that are being observed are then ranked by how far they are from the label parameter. The 'K' number is allotted by the user, which represents, e.g., how many neighbors or feature points come inside the specified group defined by K number. Then the average value of each feature is taken, and the result of this average is the prediction value.

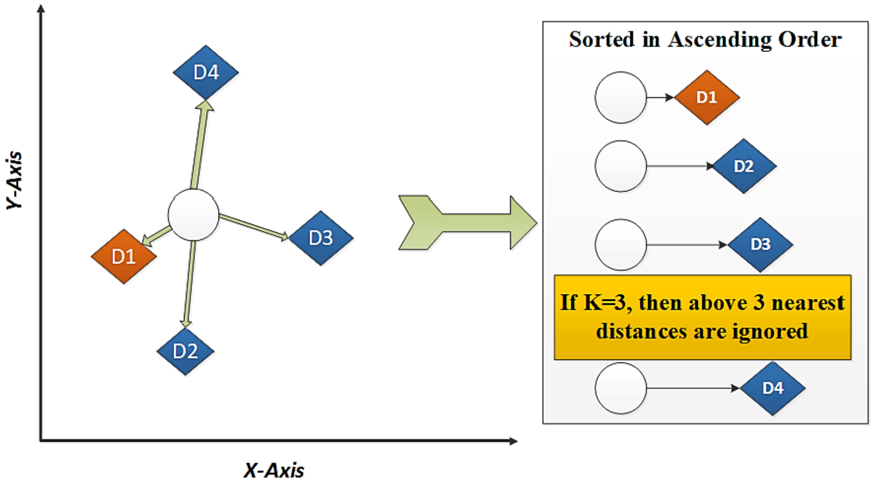
Y-Axis
X-Axis
D1
D2
D3
D4
Sorted in Ascending Order
D1
D2
D3
If K=3, then above 3 nearest distances are ignored
D4

Figure 5. Visualization of basic working of KNN

2.3 *XGBoost ML algorithm*

In this section, we delve into the XGBoost algorithm, a scalable ML system designed for tree boosting. Boosting serves as a ML technique applicable to both regression and classification problems. It involves generating a weak learner at each step and incorporating it into the overall model. If, at each step, the weak learner is constructed based on the gradient direction of the loss function, the technique is referred to as Gradient Boosting Machines (GBM) [29].

A key distinction between Random Forest (RF) and Gradient Boosted Machines lies in that trees are constructed independently of each other in RF, while in GBM, a new tree is introduced to complement those already built [30]. XGBoost outperforms the other tree boosting methods for three main reasons: (i) the incorporation of a regularized loss function, (ii) the ability to scale down the weights of each new tree by a specified constant η, thereby reducing the influence of a single tree on the final score, and (iii) column-sampling, which operates similarly to random forests [31, 32]. The XGBoost ML model is conveniently implemented in the ‘xgboost’ Python package [33].

2.4 *Pearson Correlation*

Pearson correlation is a measure of the linear relationship between two variables. It is commonly denoted by the symbol r. For two data sequences X and Y with N data points, the Pearson correlation coefficient is calculated as

$$\boldsymbol{r} = \frac{\sum_{i=1}^{N}(\boldsymbol{X}_i - \overline{\boldsymbol{X}})(\boldsymbol{Y}_i - \overline{\boldsymbol{Y}})}{\sqrt{\sum_{i=1}^{N}(\boldsymbol{X}_i - \overline{\boldsymbol{X}})^2 \sum_{i=1}^{N}(\boldsymbol{Y}_i - \overline{\boldsymbol{Y}})^2}} \tag{5}$$

where X_i and Y_i are individual data points of X and Y respectively. $\overline{X}$ and $\overline{Y}$ are the means of X and Y respectively. The numerator represents the covariance between X and Y, which measures how much X and Y change together. The denominator involves the standard deviations of X and Y, and the correlation is essentially the normalized covariance. The Pearson correlation coefficient ranges from -1 to 1 with $r = 1$ indicating a perfect positive linear relationship, while $r = -1$ indicates a perfect negative linear relationship and $r = 0$ indicates no linear relationship. The correlation coefficient is a measure of the strength and direction of the linear relationship between two variables.

2.5 *Evaluation metrics*

For performance evaluation of the proposed model, evaluation metrics of Root Mean Square Error (RMSE) and Accuracy are used. The RMSE is one of the popular standards in the context of ML [34]. The RMSE is given by

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(\boldsymbol{K}-\boldsymbol{\check{K}})^2}{\boldsymbol{N}}} \tag{6}$$

where $\boldsymbol{K}$ shows the actual observed values and $\boldsymbol{\check{K}}$ shows the predicted output values, respectively. $\boldsymbol{N}$ is the total number of records. The accuracy in percentage is given by point wise utilization of the equation

$$Accuracy\ (\%) = \left(\frac{\boldsymbol{No.\ of\ correct\ predictions}}{\boldsymbol{Total\ predictions}}\right) \times 100 \tag{7}$$

3. Numerical experimentations

In this section, details regarding implementation of imputation techniques of Mean, Mode, KNN, F&B-Fill and SDDI are presented and discussed. Furthermore, correlation analysis is done on the dataset to find which parameters are relevant in determining 5G QoS. In addition, XGBoost is applied to predict QoS with the research focusing on lowering computation and time costs for 5G QoS V2X parameter selection.

3.1 *Pertinent implementation of the imputation techniques*

Firstly, the correlation of subject dataset is computed. Figure 6 offers a full overview of what columns or input parameters has correlation with the target label (downlink throughput in bits per second). The variables of $SCell\_freq\_MHz$ and $windSpeed$ with zero correlation are consequently discarded. Similarly, the variable of *visibility* shows a white strip because it had a constant value throughout the dataset on inspection. Hence, this parameter is also discarded. Parameters exhibiting correlations as low as 0.02 were retained, as the intention is to initially consider every minor aspect of significance to discern its potential impact on accuracy results. However, further correlation analysis is discussed later in detail.

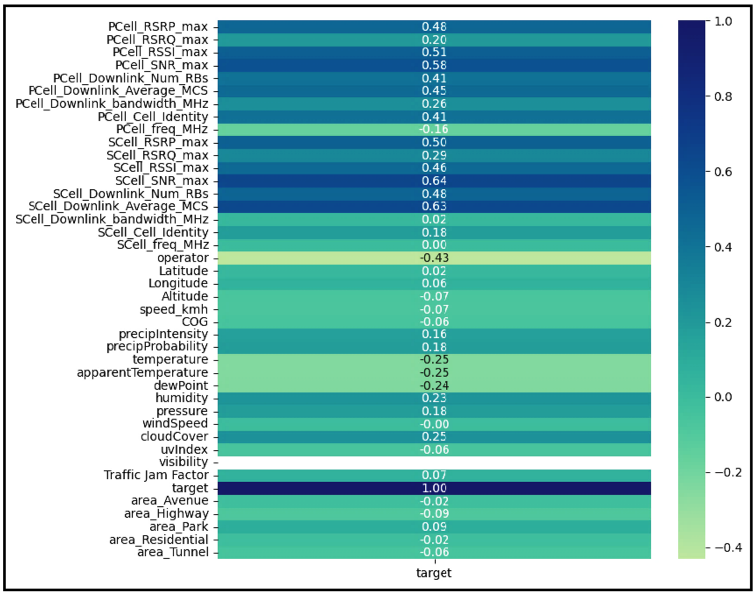

Figure 6. Correlation of the whole original data set with respect to target label column

After the initial correlation inspection and discarding absolutely non-correlated parameters the five imputation techniques are implemented to impute the original dataset and fill its missing values, then using the XGBoost (with tree depth = 6, learning rate = 0.01, estimators = 350), prediction of the target variable (downlink throughput in bits per second) is performed. The 0.75–0.25 train-test split ratio has been selected meaning that 75% data is used for training and 25% is used for testing. The training dataset contains 35 features including the target variable of data rate in bps and excludes the completely non-correlated features as discussed above. The testing dataset contains 34 columns that excludes the data rate variable and the ML model is given the task to predict this against the 34 input features.

Figures 7 and 8 depict the outcomes of curve fitting (original and predicted data points are compared to each other on the same chart) when the XGBoost algorithm is employed on five distinct versions of the identical dataset. These versions correspond to datasets that underwent various imputation techniques, including SDDI, F&B-Fill, KNN, Mean and Mode imputation. The y-axis represents the Data Rate, plotted against the x-axis, which corresponds to individual samples or data points.

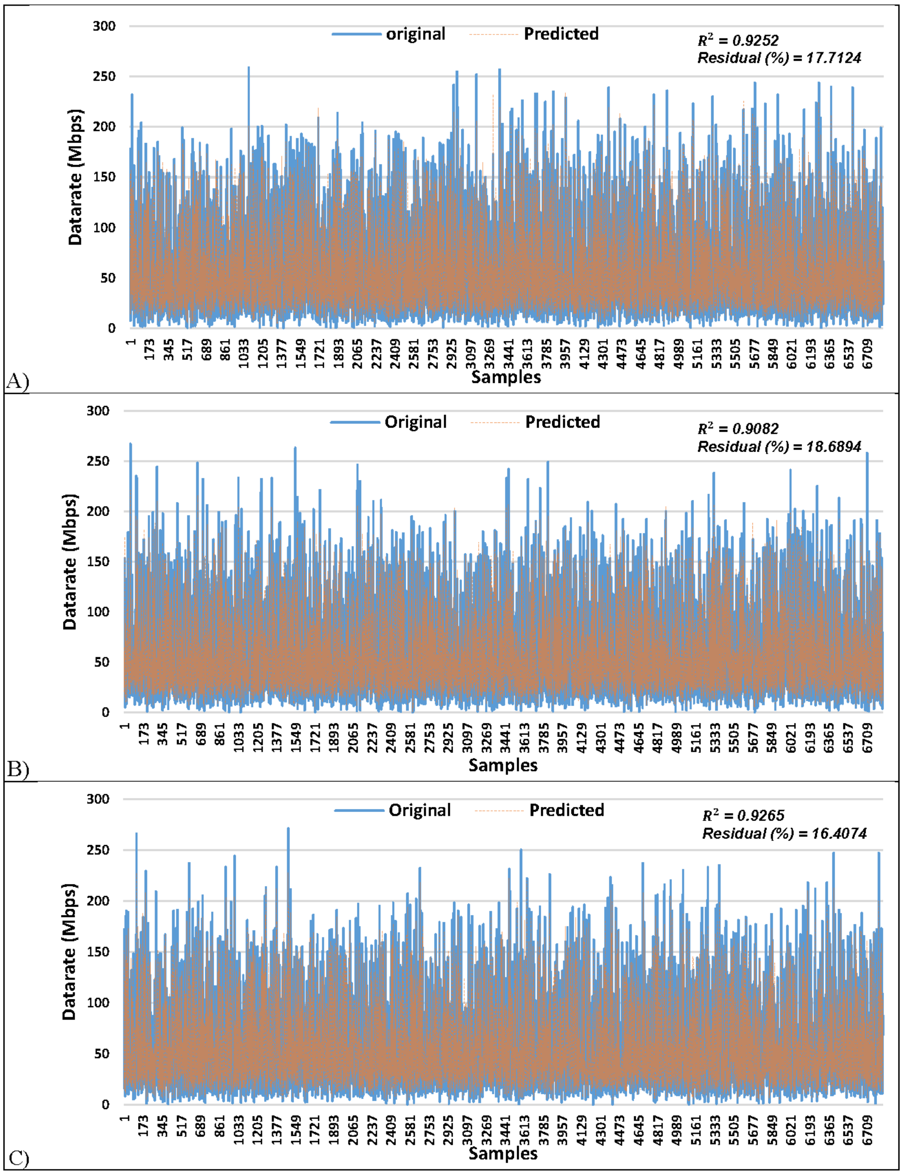

Figure 7. Curve fitting (Original vs. Predicted values) of XGBoost applied for imputation methods of (**A**) SDDI (**B**) F&B-FILL and (**C**) KNN

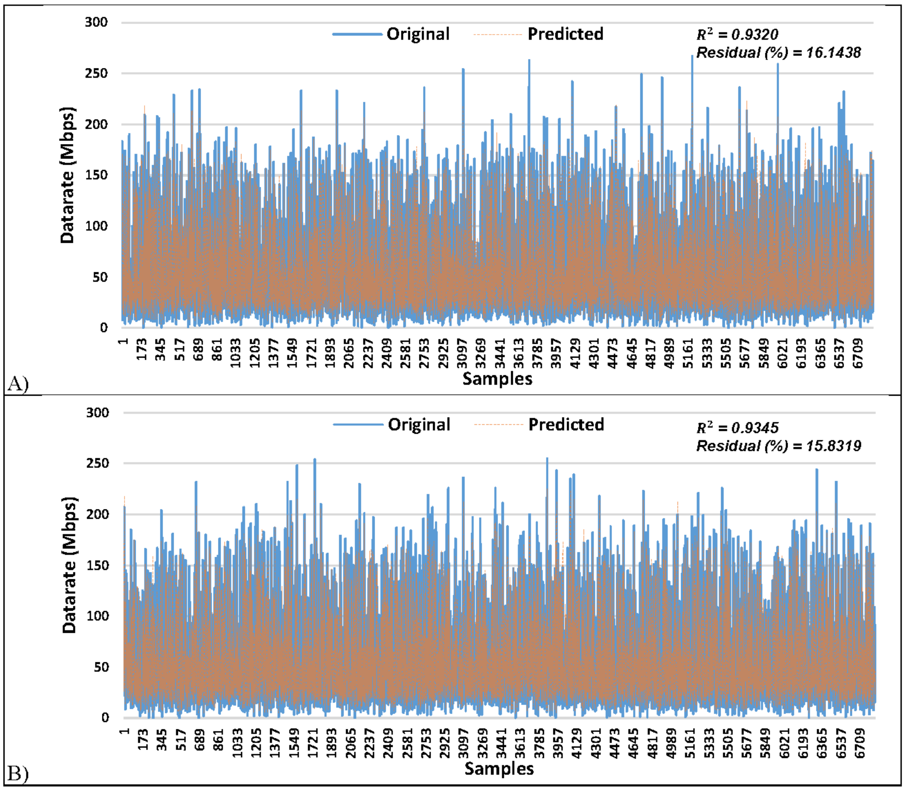

Figure 8. Curve fitting (Original vs. Predicted values) of XGBoost applied for imputation methods of (**A**) Mean and (**B**) Mode

There are two additional metrics shown specific to the Figures 7 and 8: Residual (%), which is an average of the difference between original and predicted values of every data point in the testing dataset; the r-squared evaluation metric or represented as R^2 in this study and given as

$$R^2 = \frac{\sum_1^N (\check{\boldsymbol{K}} - \widehat{\boldsymbol{K}})^2}{\sum_1^N (\boldsymbol{K} - \widehat{\boldsymbol{K}})^2} \tag{8}$$

where $\boldsymbol{K}$ shows the actual observed values, $\check{\boldsymbol{K}}$ shows the predicted output values and $\widehat{\boldsymbol{K}}$ shows the mean value. The more closer to value of 1 the R-squared is, the more precise is the prediction such that 1 means exact value match between predicted and observed value.

The definite results and comparison is shown in the Figures 9 and 10. Evaluation metrics of RMSE and average correlation obtained using the subject imputation methods are plotted in Figure 9. While the Figure 10, compares the five techniques with respect to Accuracy (%) and computation time (seconds).

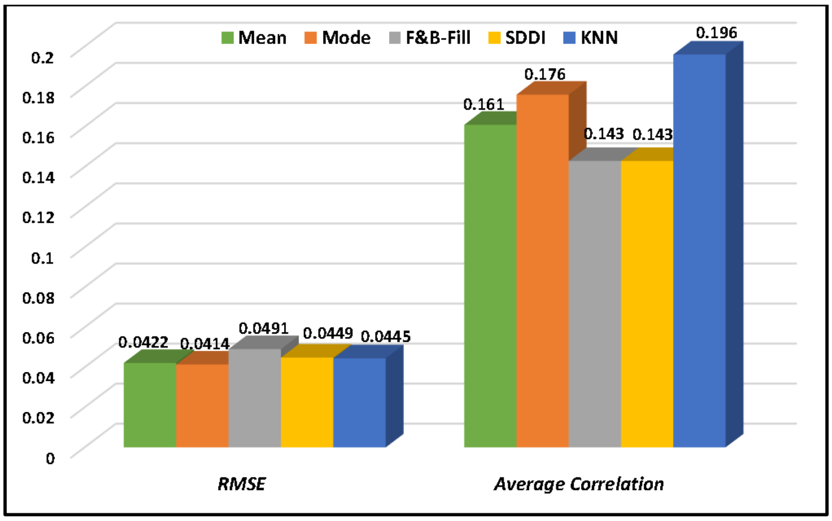

Figure 9. Comparison of various imputation methods' average correlation effect on data and RMSE performance achieved when XGBoost applied

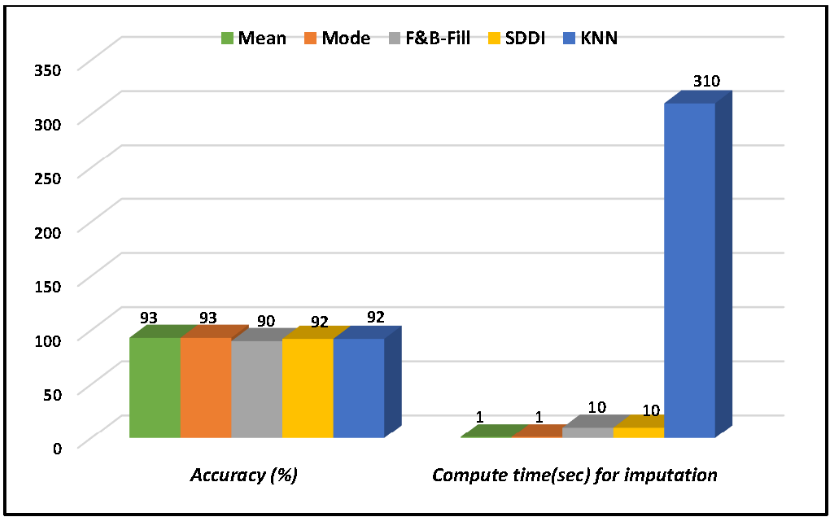

Figure 10. Comparison of various imputation methods' computation time for imputation in seconds and accuracy achieved when XGBoost applied

In Figure 9, RMSE values for various imputation techniques are presented, with the Mode imputation technique leading, having the lowest RMSE of 0.0414, closely followed by Mean imputation with an RMSE of 0.0422. In contrast, F&B-Fill lags behind with an RMSE of 0.0491. Interestingly, the advanced KNN imputer and the novel SDDI technique exhibit similar performance, both yielding an RMSE of 0.0445. Close to zero results are desirable, which is being acquired here. These results highlight that in ML or Data Science projects, no single algorithm consistently outshines the others. Success depends on the unique characteristics of datasets and research objectives. Despite these differences, the algorithms in this study performed similarly, and with very close to zero RMSE, emphasizing their comparable effectiveness.

The effect of imputation is evident from the observed correlation results for the imputed versions of the dataset, indicating the importance of understanding how a certain imputation technique affects the core pattern-related structure of the dataset, suggesting how faithfully a dataset is imputed if compared to the original dataset. In this situation, KNN surpasses other imputation techniques by producing a dataset with significantly higher correlation of 0.196. Following closely are the Mode and Mean imputed datasets, averaging correlations of 0.176 and 0.161, respectively. F&B-Fill and

SDDI exhibit comparable performance with an identical correlation of 0.143. These average correlation values has been obtained by the Table 4.

Table 4. Correlation of imputed data sets with respect to the target label column

S No.	Input features	Employed imputation methods				
		Mean	**Mode**	**F&B-Fill**	**SDDI**	**KNN**
1	PCell_RSRP_max	0.476838	0.476838	0.476838	0.476838	0.476838
2	PCell_RSRQ_max	0.199297	0.199297	0.199297	0.199297	0.199297
3	PCell_RSSI_max	0.509748	0.509748	0.509748	0.509748	0.509748
4	PCell_SNR_max	0.578254	0.578254	0.578254	0.578254	0.578254
5	PCell_Downlink_Num_RBs	0.407318	0.407318	0.407318	0.407318	0.407318
6	PCell_Downlink_Average_MCS	0.453623	0.453623	0.453623	0.453623	0.453623
7	PCell_Downlink_bandwidth_MHz	0.25083	0.252155	0.248252	0.247159	0.258138
8	PCell_Cell_Identity	0.405961	0.391666	0.400921	0.399899	0.416747
9	PCell_freq_MHz	−0.163348	−0.163348	−0.163348	−0.163348	−0.16335
10	SCell_RSRP_max	0.396969	0.566138	0.295311	0.29746	0.589953
11	SCell_RSRQ_max	0.23106	0.563075	0.171816	0.173502	0.456744
12	SCell_RSSI_max	0.362379	0.499563	0.269224	0.271616	0.472675
13	SCell_SNR_max	0.511597	0.505822	0.377273	0.383103	0.68511
14	SCell_Downlink_Num_RBs	0.37531	−0.023593	0.271604	0.276286	0.585413
15	SCell_Downlink_Average_MCS	0.487332	0.647561	0.356428	0.358503	0.665922
16	SCell_Downlink_bandwidth_MHz	0.013356	0.007616	0.011586	0.007559	0.073391
17	SCell_Cell_Identity	0.117944	0.354039	0.083687	0.073209	0.333047
18	operator	−0.431571	−0.431571	−0.431571	−0.431571	−0.43157
19	Latitude	0.018893	0.018893	0.018893	0.018893	0.018893
20	Longitude	0.063113	0.063113	0.063113	0.063113	0.063113
21	Altitude	−0.065053	−0.064986	−0.065059	−0.065159	−0.06509
22	speed_kmh	−0.072416	−0.072416	−0.072416	−0.072416	−0.07242
23	COG	−0.056743	−0.056743	−0.056743	−0.056743	−0.05674
24	precipIntensity	0.158196	0.158196	0.158196	0.158196	0.158196
25	precipProbability	0.179002	0.179002	0.179002	0.179002	0.179002
26	temperature	−0.248663	−0.248663	−0.248663	−0.248663	−0.24866
27	apparentTemperature	−0.248663	−0.248663	−0.248663	−0.248663	−0.24866
28	dewPoint	−0.244297	−0.244297	−0.244297	−0.244297	−0.2443
29	humidity	0.230256	0.230256	0.230256	0.230256	0.230256
30	pressure	0.180679	0.180679	0.180679	0.180679	0.180679
31	cloudCover	0.246949	0.246949	0.246949	0.246949	0.246949
32	uvIndex	−0.060957	−0.060957	−0.060957	−0.060957	−0.06096
33	Traffic Jam Factor	0.065516	0.065844	0.065587	0.065448	0.065333
34	area_Avenue	−0.018871	−0.018871	−0.018871	−0.018871	−0.01887
35	area_Highway	−0.088661	−0.088661	−0.088661	−0.088661	−0.08866
36	area_Park	0.091808	0.091808	0.091808	0.091808	0.091808
37	area_Residential	−0.015681	−0.015681	−0.015681	−0.015681	−0.01568
38	area_Tunnel	−0.055163	−0.055163	−0.055163	−0.055163	−0.05516
	Average	**0.161**	**0.176**	**0.143**	**0.143**	**0.196**

Nevertheless, the results in Figure 9, shows minimal deviation from each other, and there doesn't appear to be a discernible leader. This suggests that SDDI is a promising candidate as an alternative method for imputation when compared to both simple statistical techniques and ML-based approaches.

In Figure 10, the observed accuracy does not vary significantly. The mean/mode achieves 93%, while KNN and SDDI exhibit 92%, and F&B-Fill records 90%. Despite the perception that KNN, as a more advanced algorithm with a ML-based approach, would yield superior results, it underscores the idea that in data science projects, determining the best algorithm is subjective and highly dependent on the specific characteristics of the dataset. The computation time, shows that simplest techniques of Mean and Mode imputation took 1 s to execute and impute the whole dataset while 10 s each were spent by SDDI and F&B-Fill. The highest computation time of 310 s was taken by KNN as it is an ML based algorithm.

The insight derived from these results indicates that SDDI has demonstrated its efficacy, showcasing comparable performance to both statistical and ML-based algorithms.

3.2 *Parameter relevance and ML prediction for 5G QoS*

Examining the correlation results from Section 4.1, the investigation explores whether satisfactory QoS predictions can be achieved by omitting low-correlation parameters, specifically, those with correlations of 0.1 or lower and excluding weather-related parameters. If the QoS predictions are satisfactory and exhibit no significant deviation from previous results obtained using the same XGBoost ML model variant, it implies that these selected parameters can be safely utilized. This approach can be advantageous for substantial cost and time savings in measurement equipment, as well as computational efficiency, especially in real-time production tasks where the deployed model needs to process lower data volumes. Additionally, this study will contribute in signifying what parameters are needed to give 5G QoS parameters, not just for V2X but also generally for the 5G network. The Table 5 gives the names of those columns which are selected after discarding the parameters with 0.1 or lower correlation and the weather parameters.

Table 5. Columns selected after considering only greater than 0.1 correlation parameters and dropping weather parameters

No.	Parameters	Correlation	No.	Parameters	Correlation
1	PCell_RSRP_max	0.4768	11	SCell_RSRQ_max	0.1735
2	PCell_RSRQ_max	0.1992	12	SCell_RSSI_max	0.2716
3	PCell_RSSI_max	0.5097	13	SCell_SNR_max	0.3831
4	PCell_SNR_max	0.5782	14	SCell_Downlink_Num_RBs	0.2763
5	PCell_Downlink_Num_RBs	0.4073	15	Cell_Downlink_Average_MCS	0.3585
6	PCell_Downlink_Average_MCS	0.4536	16	target	1.0
7	Cell_Downlink_bandwidth_MHz	0.2471			
8	PCell_Cell_Identity	0.3998			
9	PCell_freq_MHz	−0.1633			
10	SCell_RSRP_max	0.2974			

For comparison purposes, two variants named for convenience as XGBoost V1 (depth = 6, learning rate = 0.01, estimators = 350) and XGBoost V2 (depth = 8, learning rate = 0.01, estimators = 400) have been used. The XGBoost V1 is the same as used in the Section 4.1 to get results for imputed datasets. While the XGBoost V2 is a bit more complex with slightly more depth and estimators used. More complexity is not desired since the study is focusing on faster results with low computing power. The SDDI imputed dataset is used for this section. The results for RMSE were observed to be 0.0484 and 0.0436 for XGBoost V1 and XGBoost V2, respectively. The accuracy of 90% and 93% were observed in Figure 11 for XGBoost V1 and XGBoost V2, respectively. However, the 3% accuracy was gained at more than 50% computation cost with computation time of 24 and 60 s for XGBoost V1 and XGBoost V2, respectively.

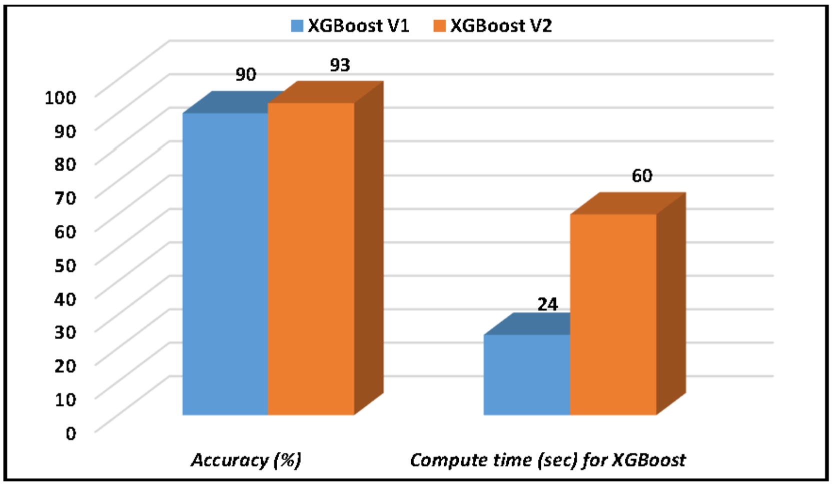

Figure 11. Comparison of XGBoost V1 with XGBoost V2 for accuracy and compute time taken on smaller correlation prioritized dataset

Hence, the results do show that obtained result from the same XGBoost V1 are satisfactory, especially considering the fact that only 15 input parameters are considered compared to original 38 original input columns in Section 4.1. Meaning, a 60.5% reduction in data affected only 3% accuracy, which is in itself an evidence that non-correlated features were definitely not of a significant value to measure with or use in ML pipeline.

This is even further realized in the Figure 12 by comparing r-squared metric results for XGBoost V1 and XGBoost V2, with 0.9114 and 0.9285 values which are very close in the context of 60.5% data reduction.

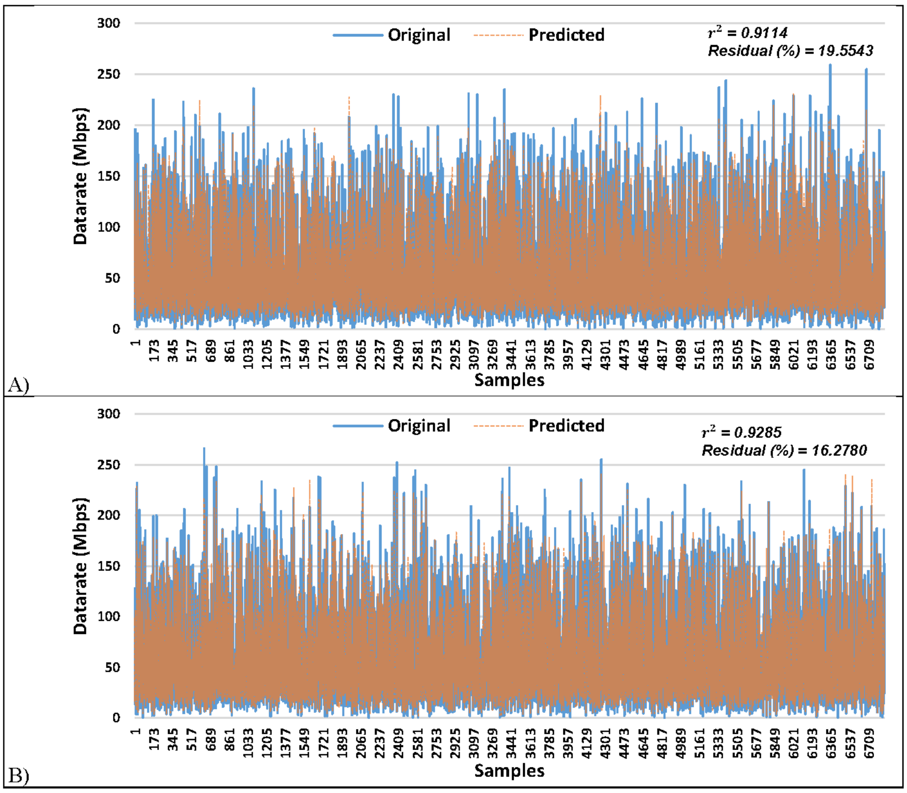

Figure 12. Curve fitting (original vs. predicted values) of XGBoost applied on smaller correlation prioritized dataset with variation of (**A**) XGBoost V1, and (**B**) XGBoost V2

Consequently, it can be concluded that these 15 parameters can be confidently selected, given their minimal impact on accuracy, thereby streamlining the data and contributing to more efficient and resource-effective analyses.

4. Conclusions

This work introduced a novel imputation method utilizing the distribution of the data itself to fill the missing values. This alternative approach, was shown to be computationally simple but strategic enough to be on par with other state-of-the-art methods. The research also contributed to the 5G QoS parameter selection in both general and V2X contexts by utilizing Pearson correlation and XGBoost implementations.

With data collected in Berlin, Germany, in the initial stages of the research, Pearson correlation method was used to identify the outliers that had no correlation at all, and then imputation methods were applied including the novel SDDI algorithm, KNN, F&B-Fill, Mean and Mode methods. The resulting imputed data was trained on XGBoost algorithm and evaluated via RMSE, accuracy (%), computation time, and average correlation. The results showed that the novel SDDI algorithm was on par with the other methods.

The study also identified the most significant parameters in determining 5G QoS and therefore, to evaluate the potential of selected features for QoS prediction suitability, XGBoost's two variants were applied with excellent accuracies of 90% and 93%. RMSE results were observed to be both very close to zero and therefore, successful.

The results from this research hold promising implications for the wireless communication sector, particularly in the context of 5G QoS parameter selection and prediction. The methodological advancements presented in this study offers alternative imputation method in imputing datasets, offering valuable insights into application of statistical, low compute cost and strategic algorithms in the Data Science and ML domains.

Future research could focus on further application of the proposed imputation method (SDDI method) by testing it on various other datasets and comparing its performance across various domains. Investigating its adaptability to different types of missing data scenarios, will open up even more insights of the efficacy of the said technique.

Conflict of interests

There is no conflict of interest declared by the authors.

References

[1] C. R. Padgett, C. E. Skilbeck, and M. J. Summers, "Missing data: the importance and impact of missing data from clinical research," *Brain Impair.*, vol. 15, no. 1, pp. 1–9, May 2014.

[2] G. Press, "Cleaning Big Data: Most Time-Consuming, Least Enjoyable Data Science Task, Survey Says," *Forbes*, 2016. Accessed: Sep. 21, 2023. [Online]. Available: https://www.forbes.com/sites/gilpress/2016/03/23/data-preparation-most-time-consuming-least-enjoyable-data-science-task-survey-says/.

[3] S. Kumar, "7 Ways to Handle Missing Values in Machine Learning," *Towards Data Sci.*, 2020. Accessed: Sep. 21, 2023. [Online]. Available: https://towardsdatascience.com/7-ways-to-handle-missing-values-in-machine-learning-1a6326adf79e#.

[4] I. Pratama, A. E. Permanasari, I. Ardiyanto, and R. Indrayani, "A review of missing values handling methods on time-series data," in *Proc. 2016 Int. Conf. Inf. Technol. Syst. Innov. (ICITSI)*, Bandung, Indonesia, Oct. 24–27, 2016, pp. 1–6.

[5] T. Aljuaid and S. Sasi, "Proper imputation techniques for missing values in data sets," in *Proc. 2016 Int. Conf. Data Sci. Eng. (ICDSE)*, Cochin, India, Aug. 23–25, 2016, pp. 1–5.

[6] "Working with Missing Data," University of Prince Edward Island. Accessed: Sep. 21, 2023. [Online]. Available: https://pressbooks.library.upei.ca/montelpare/chapter/working-with-missing-data/.

[7] J. Sessa and D. Syed, "Techniques to deal with missing data," in *Proc. 2016 5th Int. Conf. Electron. Dev. Syst. Appl. (ICEDSA)*, Ras Al Khaimah, United Arab Emirates, Dec. 6–8, 2016, pp. 1–4.

[8] S. Gavankar and S. Sawarkar, "Decision tree: Review of techniques for missing values at training, testing and compatibility," in *Proc. 2015 3rd Int. Conf. Artif. Intell. Modelling Simul. (AIMS)*, Kota Kinabalu, Malaysia, Dec. 2–4, 2015, pp. 122–126.

[9] Google Trends, "Interest Over Time," Google, 2024. Accessed: Nov. 2, 2024. [Online]. Available: https://trends.google.com/trends/explore?cat=174&q=data%20imputation,machine%20learning%20models, deep%20learning%20models&hl=en-GB.

[10] N. Liu, Y. Li, Z. Zang, Y. Hu, X. Fang, and S. Lolli, "A deep learning-based imputation method for missing gaps in satellite aerosol products by fusing numerical model data," *Atmos. Environ.*, vol. 325, p. 120440, 2024.

[11] M. D. Hssayeni and B. Ghoraani, "Deep Regression Modeling for Imbalanced and Incomplete Time-Series Data," *IEEE Trans. Emerg. Top. Comput. Intell.*, vol. 8, pp. 3767–3778, 2024.

[12] T. Chrysikos, P. Georgakopoulos, I. Oikonomou, and S. Kotsopoulos, "Measurement-based characterization of the 3.5 GHz channel for 5G enabled IoT at complex industrial and office topologies," in *Proc. 2018 Wireless Telecommun. Symp. (WTS)*, Phoenix, AZ, USA, Apr. 17–20, 2018.

[13] GVA Mission Briefing 5G, 28 Sept 2016. Accessed: Apr. 19, 2019. [Online]. Available: https://www.itu.int/en/membership/Documents/missions/GVA-mission-briefing-5G-28Sept2016.pdf.

[14] A. Hikmaturokhman, K. Ramli, and M. Suryanegara, “Spectrum Considerations for 5G in Indonesia,” in *Proc. 2018 Int. Conf. ICT Rural Dev. (IC-ICTRuDev)*, Badung, Indonesia, Oct. 17–18, 2018.

[15] M. Singh and G. Baranwal, “Quality of service (QoS) in Internet of Things,” in *Proc. 2018 3rd Int. Conf. Internet Things: Smart Innov. Usages (IoT-SIU)*, Bhimtal, India, Feb. 23–24, 2018, pp. 1–6.

[16] R. D. Mardian, M. Suryanegara, and K. Ramli, “Measuring quality of service (QoS) and quality of experience (QoE) on 5G technology: A review,” in *Proc. 2019 IEEE Int. Conf. Innov. Res. Dev. (ICIRD)*, Jakarta, Indonesia, Jun. 28–30, 2019, pp. 1–6.

[17] M. M. Nasralla and M. G. Martini, “A downlink scheduling approach for balancing QoS in LTE wireless networks,” in *Proc. 2013 IEEE 24th Annu. Int. Symp. Personal, Indoor, Mobile Radio Commun. (PIMRC)*, London, UK, Sep. 8–11, 2013, pp. 1571–1575.

[18] T. Daengsi and P. Wuttidittachotti, “Quality of Service as a Baseline for 5G: A Recent Study of 4G Network Performance in Thailand,” in *Proc. 2020 IEEE Int. Conf. Commun., Netw. Satellite (Comnetsat)*, Batam, Indonesia, Dec. 17–18, 2020, pp. 395–399.

[19] R. Dong, C. She, W. Hardjawana, Y. Li, and B. Vucetic, “Deep learning for radio resource allocation with diverse quality-of-service requirements in 5G,” *IEEE Trans. Wireless Commun.*, vol. 20, no. 4, pp. 2309–2324, Dec. 2020.

[20] P. Yu, F. Zhou, X. Zhang, X. Qiu, M. Kadoch, and M. Cheriet, “Deep learning-based resource allocation for 5G broadband TV service,” *IEEE Trans. Broadcast.*, vol. 66, no. 4, pp. 800–813, Feb. 2020.

[21] M. Urwin, “4 Disadvantages of Neural Networks,” *Built In*, 2023. Accessed: Nov. 11, 2023. [Online]. Available: https://builtin.com/data-science/disadvantages-neural-networks.

[22] C. Wade, “Getting Started with XGBoost in scikit-learn,” *Towards Data Sci.*, 2020. Accessed: Sep. 25, 2023. [Online]. Available: https://towardsdatascience.com/getting-started-with-xgboost-in-scikit-learn-f69f5f470a97.

[23] W. Jiang, H. Han, M. He, and W. Gu, “ML-based pre-deployment SDN performance prediction with neural network boosting regression,” *Expert Syst. Appl.*, vol. 241, p. 122774, 2024.

[24] Y. Yin, Q. Di, J. Wan, and T. Liang, “Time-Aware Smart City Services based on QoS Prediction: A Contrastive Learning Approach,” *IEEE Internet Things J.*, vol. 10, no. 21, pp. 18745–18753, 2023.

[25] “QoS Prediction Challenge AI/ML in 5G Challenge,” *Kaggle*, Jun. 2023. Accessed: Oct. 14, 2023. [Online]. Available: https://www.kaggle.com/datasets/gauravduttakiit/qos-prediction-challenge-aiml-in-5g-challenge/.

[26] S. Moritz, A. Sardá, T. Bartz-Beielstein, M. Zaefferer, and J. Stork, “Comparison of different methods for univariate time series imputation in R,” *arXiv*, 2015, preprint arXiv:1510.03924.

[27] S. Moritz and T. Bartz-Beielstein, “imputeTS: time series missing value imputation in R,” *R J.*, vol. 9, no. 1, p. 207, 2017.

[28] S. Khan, O. L. Lo’pez, A. B. Khattak, “Mutual Exploration for Missing Data Imputation, Qos Parameter Selection, and Qos Prediction in 5g Networks Using a Novel Skewness Driven Distribution Imputation Algorithm, Pearson Correlation, and Xgboost,” *Int. J. Data Sci. Big Data Anal.*, vol. 3, no. 2, pp. 51–58, 2023. https://doi.org/10.51483/IJDSBDA.3.2.2023.51-58.

[29] J. H. Friedman, “Stochastic gradient boosting,” *Comput. Stat. Data Anal.*, vol. 38, no. 4, pp. 367–378, 2002, https://doi.org/10.1016/S0167-9473(01)00065-2.

[30] M. Luckner, B. Topolski, M. Mazurek, “Application of XGBoost algorithm in fingerprinting localisation task,” *IFIP International Conference on Computer Information Systems and Industrial Management*. Cham, Switzerland: Springer International Publishing, 2017, pp. 661–671.

[31] A. B. K. Didavi, R. G. Agbokpanzo, and M. Agbomahena, “Comparative study of Decision Tree, Random Forest and XGBoost performance in forecasting the power output of a photovoltaic system,” in *Proc. 4th Int. Conf. Bio-Eng. Smart Technol. (BioSMART)*, Paris, France, Dec. 8–10, 2021, pp. 1–5. https://doi.org/10.1109/BioSMART54244.2021.9677566.

[32] Restack, “XGBoost vs Deep Learning in MLflow,” *restack.io*. Accessed: Jan. 23, 2024. [Online]. Available: https://www.restack.io/docs/mlflow-knowledge-xgboost-vs-deep-learning-mlflow.

[33] Scikit Learn, “Gradient Boosting Classifier,” *scikit-learn*. Accessed: Jan. 23, 2024. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html.

[34] D. S. K. Karunasingha, “Root mean square error or mean absolute error? Use their ratio as well,” *Inf. Sci.*, vol. 585, pp. 609–629, 2022.