

## Development of a Machine Learning Based Fault Detection Model for Received Signal Level in Telecommunication Enterprise Infrastructure



Kennedy Okokpujie<sup>1,2,3\*</sup>, Innocent O. Nwokolo<sup>1</sup>, Akingunsoye V. Adenugba<sup>4</sup>, Morayo E. Awomoyi<sup>5</sup>

<sup>1</sup> Department of Electrical and Information Engineering, Covenant University, Ota 112101, Nigeria

<sup>2</sup> Africa Centre of Excellence for Innovative & Transformative STEM Education, Lagos State University, Ojo 102003, Nigeria

<sup>3</sup> Informatics & Communication African Centre of Excellence (CApIC-ACE), Covenant University, Ota 112101, Nigeria

<sup>4</sup> Directorate, OVA Foundation, Millington 21651, USA

<sup>5</sup> US School of International Service, American University, Tenleytown 20016, USA

Corresponding Author Email: [kennedy.okokpujie@covenantuniversity.edu.ng](mailto:kennedy.okokpujie@covenantuniversity.edu.ng)

Copyright: ©2024 The authors. This article is published by IETA and is licensed under the CC BY 4.0 license (<http://creativecommons.org/licenses/by/4.0/>).

<https://doi.org/10.18280/ijse.140302>

### ABSTRACT

**Received:** 26 June 2023

**Revised:** 18 January 2024

**Accepted:** 29 January 2024

**Available online:** 24 June 2024

#### Keywords:

*Machine learning, enterprise wireless, telecommunication, received signal levels (RSL)*

This research develops a machine-learning fault detection model for received signal levels in telecommunication infrastructure. The methodology involves modeling an enterprise point-to-multipoint wireless network using pathloss 5.0 software. Data from the simulated network, including free space pathloss, transmit power output, transmit antenna gain, transmitter loss, miscellaneous loss, and receiver loss, is used to train three regression models: gradient boosting regression (GBR), random forest regression (RFR), and K-Nearest Neighbor (KNN). The algorithm compares the received signal levels (RSL) of new data with a threshold value, triggering a "Fault" or "No-fault" condition. A "Fault" indicates a deviation in the RSL, prompting maintenance by the field support team. A "No-fault" means the RSL is within the accepted range, requiring no maintenance. Performance evaluation metrics such as mean absolute error (MAE), mean square error (MSE), R-squared, and root mean square error (RMSE) were compared to select the optimal model. Experimental results show that the RFR model outperforms GBR and KNN with MAE: 0.007101, MSE: 0.000610, R-squared: 0.999992, and RMSE: 0.024697. Leveraging these machine learning-based fault detection models enables telecom service providers to optimize network performance, reduce downtime, and increase customer satisfaction.

## 1. INTRODUCTION

Wireless telecommunication infrastructure can fail without notice for maintenance action. However, these failures may not always result in a complete downtime, but rather degraded performance that can be difficult to pinpoint without specialized tools. Received signal level (RSL) is an important metric of the quality of a wireless connection and can be used to determine the strength of the signal from the transmitting to the receiving device. It is important to regularly monitor the RSL to ensure that the network is operating at peak performance and to proactively address any issues that may arise.

As businesses rely more and more on the networks to increase operational effectiveness and foster long-term business growth, telecommunications are a crucial component of today's business world. Through the power of telecommunication, businesses have experienced improved collaboration, enhanced communication and maximum productivity. Telecommunications networks have evolved into an important medium that provides the necessary platform for this electronic data exchange. Yet, in the current digital ecosystem, organizations can use mobile communication to

speed up workflow and productivity while allowing employees to use their devices to access particular applications, reply to emails, work on presentations, and take part in teleconference calls. This is made possible through the telecommunication service providers infrastructure.

The telecommunications network is responsible for carrying all internet data and can be comprised of various technologies, including satellites, microwaves, and mobile networks like 5G. Efficient communication relies on telecommunication infrastructure, which enables individuals and organizations to communicate via wired and wireless connections, phone, internet, and other mediums. As demand for connectivity increases globally, customers and end-users expect modern telecommunication networks to operate with minimal downtime, making carriers that can offer nearly 100% uptime the preferred choice for many consumers.

Telecommunication network failure of enterprise customers' equipment are avoidable, with careful planning, constant monitoring and maintenance in addition with spare parts available at all times. To circumvent the problem of constant outages which may lead to reputational damage, telecommunication service providers started searching for better ways to improve service delivery by applying the right

maintenance strategies.

The majority of telecommunication service providers began to use reactive or preventive maintenance procedures [1]. Reactive or breakdown maintenance is the strategy whereby no maintenance is carried out until machine breaks down. While predictive maintenance involves performing maintenance based on predictions derived from the analysis of key degradation parameters. The downside of both strategies is that reactive maintenance allows failure to occur before actions are taken [2] while preventive maintenance acts irrespective of the current state of the equipment to be maintained [3]. This means that the customers network will be shutdown to perform the maintenance whether the service is fine or not. Usually, customers are always upset with this form of maintenance decisions as their network will be interrupted when they are at the middle of a major task. This could also lead to loss of revenue for bigger businesses. Predictive maintenance (PdM) technology on the other hand has been widely used to manage the health of equipment and predict equipment failure so that organizations can schedule maintenance in advance to avoid unplanned equipment downtime and improve customer service [3-5]. In the light of this, the predictive maintenance strategy seems more appropriate for the telecommunications provider as it relies on the data to make decisions when equipment will be maintained before a customer complains. Maintenance actions can then be initiated to replace or repair the faulty components so that the associated unit can continuously perform its intended function throughout its useful life. Consequently, equipment failures must be discovered and corrected to avoid service disruption [6] and should be performed proactively to reduce maintenance costs and increase equipment uptime to the maximum extent practicable. This fact necessitates a shift in maintenance techniques from diagnostics to a prognostics approach.

The problem statement of this research is that the telecommunications industry is always looking for ways to maintain its infrastructure and provide reliable service while keeping costs to a minimum. Many researchers have engaged in reviewing ways to improving maintenance system for about twenty years now [7]. The reactive maintenance approach is only suitable when a failure occurs causing a sudden shutdown and end users are always unhappy with the interruption of service. The preventive maintenance (PvM) technique emerged as one of the prevalent strategies used by most telecommunication businesses. However, the annual cost is considerably high since more replacement spare must be purchased and stored. The success of this strategy hinges on having all replacement spares on hand for scheduled maintenance tasks on time. Predictive maintenance strategies work without a schedule for servicing or parts replacement. The main idea is to predict a failure and then replace the spares before their lifespan expires. If there is any evidence of deterioration, the maintenance personnel will perform the replacement appropriately. Predictive maintenance reduces downtime, optimizes spare parts inventory, and extends the longevity of the equipment [8, 9]. In a traditional approach, Network Operations Center (NOC) personnel manually monitor the receive signal levels of the wireless enterprise customers and then engage a field support personnel to investigate any deviation in the signal parameters. However, with the increasing complexity of telecommunication systems and the rapid growth of the network, this manual approach becomes time-consuming. PdM applications are increasingly

using intelligence, and model-based approaches due to the fact of it's more efficient to data-driven decision-making and real-time applications. The science of artificial intelligence has seen the emergence of machine learning method in the field of predictive maintenance. This problem can be mitigated by developing of a machine learning based fault detection model for received signal level in telecommunication enterprise infrastructure that can automate the process of monitoring and alert network operators of any deviations from optimal signal levels. Therefore, the problem to be addressed in this research work is identify system metric that indicate an impending system failure and develop a machine learning model that can accurately monitor the system metric and provide timely alerts to network operators for effective telecommunication maintenance purposes.

## 2. LITERATURE REVIEW

Traditionally, a team of specialists in the Network Operations Center (NOC) troubleshoots, locates, and resolves problems by examining the alarms accumulated in various network segments. As networks become more complex due to increasing demand, there is a need to have proactive solutions for automatic reporting of failures [10]. These solutions notify operators of both present and future potential issues. However, such alarms provide limited relevant data to network administrators, with only a small subset of them being relevant to current operational issues [11]. Recent advancements in information technology have allowed big data approaches to handle significant volumes of data. This is seen as a cost-effective information asset that can enhance decision-making and process automation, according to Gartner [12].

There are various steps to the process of maintaining a telecommunication network's faults, including fault detection, reporting of faults, diagnosis of faults, and resolution of faults [11]. These steps are crucial for figuring out whether the system is operating normally or if a malfunction has occurred. Detecting faults at an early stage is crucial for optimizing mobile networks, according to Rezaei et al. [13], who also claim that identifying these faults is the first step towards implementing any system that can make decisions. The system can automatically recover or the field engineer can step in if a fault is detected. The fault alarms follow a specific format established by the equipment vendor, containing information such as the device that caused the fault, a clear and concise explanation of the fault, the level of alarm severity, and additional details related to the fault logs, such as the node identifier, start time of the fault, and other relevant information. After identifying the critical alerts and creating a ticket to record the fault's history, the level 2 team takes charge of investigating the fault. Their primary objective is to identify the underlying cause of the failure and develop a suitable solution to rectify the problem [14].

Depending on the severity, it may be possible to resolve the fault remotely without taking any more on-site steps to restore service. Yet, in other circumstances, a physical intervention—such as a field engineer's on-site visit—might be necessary for fault resolution. When this event occurs, a new set of tickets will be generated and sent to the field team. The on-site support team acknowledges the dispatch notification and takes necessary corrective action to repair the failed hardware component. The support team then updates the ticket with information about the actions taken to resolve the failure [11].

Avoiding system failures and network downtime is crucial in meeting the Service Level Agreement [10]. With increasing network complexity, effective fault management becomes more critical, and network operators need to anticipate faults in advance to make timely repair decisions [15]. The objective of developing advanced maintenance solutions is to minimize maintenance costs. In order to achieve this objective for the best maintenance method to maintain telecommunication infrastructure.

## 2.1 Machine learning

The science of artificial intelligence has seen the emergence of machine learning methods in the field of predictive maintenance. Machine learning, among other things, is a complex combination of algorithms based on AI that are frequently used within knowledge discovery to assist systems in discovering patterns and structures from training examples [16]. This technique involves creating a model based on historical input data and its output behavior, which enables accurate forecasting of outcomes [17]. By capturing knowledge in data and uncovering hidden patterns, ML algorithms provide a more efficient approach to data-driven decision-making. Machine learning can execute complex algorithms by learning from data instead of relying on the pre-programmed instructions [18, 19]. The idea stems from the fact that computers can solve problems that require them to determine the relationship between the output of a vector of input variables ( $x$ ). To achieve this, a training data set consisting of  $N$  input and output value samples is used, and various learning methods are employed to create a function  $y(x)$  that predicts the output variable's value for a new input variable value.

The machine learning workflow are of two stages, consisting of training and decision-making. In the training phase, machine learning techniques are utilized to develop a model by analyzing the training dataset. During the decision-making phase, the system can apply the trained model to obtain an estimated output for each new input. Supervised learning algorithms which are one of the various categories of machine learning technique require a labeled training dataset to build a model that describes the relationship between input and output. A supervisor is needed to inform the system about the expected output for each input in supervised learning. There are multiple supervised learning algorithms available, each with its own set of requirements and uses [20, 21].

There are various machine learning algorithms available for different types of problems. For example, Carvalho et al. [22] investigated different studies on predictive maintenance published between 2009 and 2018, which utilized vibration signal data generated from PdM devices. They investigated various machine learning techniques to solve the prediction problem. The PdM applications utilized specific equipment such as turbines, motors, compressors, pumps, and fans. The research paper examined various machine learning techniques, the equipment used in maintenance prediction, and the type of data used in the machine learning algorithm. According to the authors' reviews, Random Forest is the most commonly used machine learning algorithm with 33%, then the neural network-based algorithms with 27%, support vector machines (SVM) with 25% while k-means is the least with 13%. The authors stressed the significance of selecting an appropriate machine learning method to achieve optimal performance in predictive maintenance applications. To demonstrate this, they

conducted a comprehensive review of the literature on machine learning methods used in predictive maintenance. They focused on the techniques currently being investigated in the field and evaluated the effectiveness of state-of-the-art machine learning approaches. Furthermore, they discussed the impact of artificial intelligence (AI) on future predictive maintenance, which is a vital aspect of advanced production systems. Specifically, they discussed the reasons for the interest in applying deep learning technology in predictive maintenance strategies, but cautioned that it may not be suitable for every problem as it often requires large datasets for training.

Çinar et al. [23] conducted a comprehensive literature review to identify existing machine learning (ML) applications in order to guide researchers and practitioners in selecting appropriate ML techniques, data size, and data types for feasible ML applications. Their paper provides an exhaustive review of ML techniques applied in predictive maintenance over a ten-year period (2010-2020). According to their analysis, SVM, RF, and ANN are the most commonly used ML algorithms in the review literature. However, it has been observed that RF is the most widely used ML technique in PdM and has been used in various industrial equipment, components and systems.

Research papers, such as those by Jardine et al. [8], Lei et al. [24], and Uddin et al. [25], show that significant efforts have been made over the past two decades to improve Predictive Maintenance systems, with knowledge-driven and data-driven approaches being the most common [26-29]. The data-driven approach relies on data mining techniques to create models directly from historical records, making it an ideal tool for Predictive maintenance applications [29]. Machine learning, a subset of artificial intelligence, is increasingly important in this approach as it teaches computers to learn directly from data without relying on a given equation as a model. As more data becomes available for learning, machine learning algorithms can adjust their performance to develop powerful predictive model.

The reason behind this conclusion is that machine learning techniques can effectively deal with complex and multivariate data by uncovering underlying relationships. Machine learning algorithms are designed to learn and improve their performance by analyzing input data and making predictions with minimal or no human intervention, as noted by Calabrese et al. [30], Susto et al. [31], Verhagen and Boer [32].

## 2.2 Frameworks for building machine learning systems

Three data mining frameworks widely used by machine learning professionals: Knowledge Discovery in Database (KDD), Cross-Industry Standard Process for Data Mining (CRISP-DM), Sample, Explore, Modify, Model, and Assess (SEMMA) [12]. KDD, which was coined by Gregory Piatetsky-Shapiro in 1989, is a process of extracting valuable knowledge from data. The CRISP-DM model is an iterative process, which means that many of the actions taken will revisit prior steps and repeat procedures in order to achieve clarity [33, 34]. SEMMA is another popular data mining framework that comprises of five stages. However, most researchers and data mining experts prefer to use KDD and CRISP-DM process models because of their comprehensive and precise nature. It is however clear that CRISP-DM is more complete as it provides a well-defined iterative flow of knowledge across and between stages. Furthermore, it covers

all the crucial aspects of creating a dependable machine learning system from a business standpoint.

### 3. MATERIALS AND METHODS

This section presented the conceptual framework, the materials and methods employed in achieving the research goals.

#### 3.1 Conceptual framework

The conceptual framework is made up of three modules: a data collection module, data analytics module and a machine learning management module. The Data Analytics Module is responsible for collecting the received signal level data from the telecommunication network and verifying that the data is correctly formatted for further modeling. The machine learning module extracts characteristics from the preprocessed dataset to train and assess the model's performance as shown in Figure 1, while Table 1 shows the Material and software applications used in carryout this research.

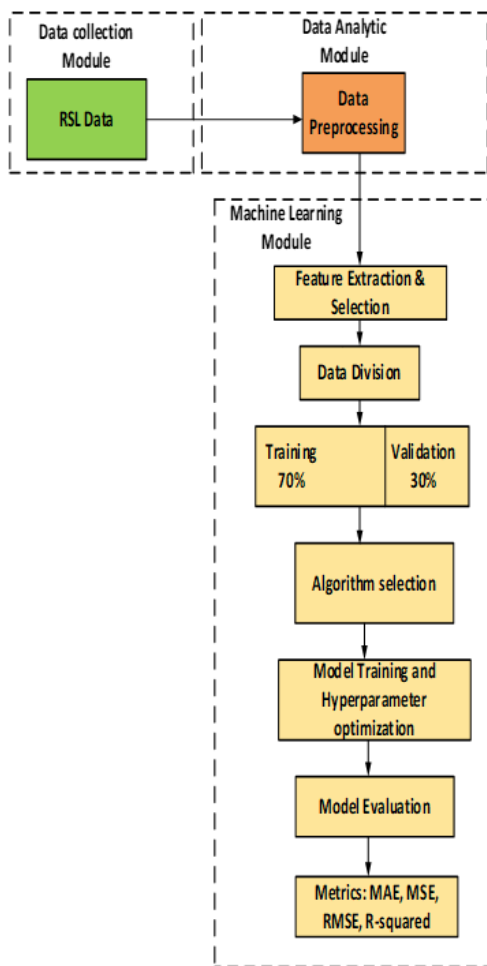


Figure 1. Conceptual framework

#### 3.2 Data generation approach

To simulate the received signal level data, the following methods were investigated and in this study a physics-based simulation method using the Path Loss 5.0 software was utilized. Table 2 contains the parameters for the Transmitter (Tx) and Receiver (Rx) which are utilized in the simulation of

a vector element for a functional telecommunication link using the pathloss 5.0 software. The table provides specific information on the transmit and receiver antenna gain, transmit power output, frequency, polarization, coordinates, and antenna height which are used to determine the path length. These parameters were supplied by the microwave radio manufacturer. On the other hand, the parameters in Table 3 were employed to generate data that could be used for testing the machine learning model. Therefore, the values in both tables are distinct since they were generated using parameters from different base stations.

Table 1. Material and software applications

Resource	Details
CPU	16GB, Core i5 computer
Pathloss 5.0	For microwave wireless link design and planning
Jupyter Notebook	Open-source machine learning and data analysis platform
Ipython	Evolved into Jupyter with an interactive command-line interface
Scikit-learn	Full featured library for ML algorithms.
Pandas	open-source data manipulation and analysis library.
NumPy	open-source library with collection of mathematical functions.
Matplotlib	plotting library for Python.
Seaborn	python library used for plotting and creating a wide variety of statistical plots.
Pickle	Python library for serialization and deserialization of an object's state to a byte stream and vice versa.

Table 2. Parameters for simulating training datasets

Transmitter Details	Receiver Details
Transmitter power output (21dBm)	Receiver power output (-100dBm)
Antenna height (36.0m)	Antenna height (9.0m)
Transmitter Antenna Gain (16dBi)	Receiver Antenna Gain (33.9dBi)
Vertical polarization	Vertical polarization
Frequency (10GHz)	Frequency (10GHz)
Latitude 06 38 57.19 N	Latitude 06 39 04.54 N
Longitude 003 21 54.94 E	Longitude 003 20 59.73 E
Path length = d(km) = 1.71km	

Table 3. Parameters for simulating testing datasets

Transmitter Details	Receiver Details
Transmitter power output (18dBm)	Receiver power output (-100dBm)
Antenna height (36.0m)	Antenna height (9.0m)
Transmitter Antenna Gain (13.3dBi)	Receiver Antenna Gain (33.9dBi)
Vertical polarization	Vertical polarization
Frequency (10GHz)	Frequency (10GHz)
Latitude 06 38 57.19 N	Latitude 06 39 04.54 N
Longitude 003 21 54.94 E	Longitude 003 20 59.73 E
Path length = d(km) = 1.71km	

Simulating data for received signal level (RSL) involves generating values for the signal strength that represents a real-world scenario and the following steps were followed:

- a) Determine the frequency of transmission.
- b) Select a propagation model to determines how the

signal strength changes over the distance.

- c) Select the transmit power according to the manufacturer's specifications.
- d) Choose the antenna heights and define the environments.
- e) Generate data using pathloss 5.0.
- f) Validate the data to ensure the simulated data accurately represents the real-world scenario.

### 3.2.1 Pathloss 5.0

Pathloss 5.0 is a robust software program that is useful for designing, optimizing, and planning radio networks. It offers an array of tools to evaluate and simulate radio propagation and to anticipate coverage and signal strength which allows users to input the base station (transmitter) and users (receiver) end parameters such as the base station location, antenna height, transmit power, frequency band, modulation scheme. Once the transmitter and receiver parameters have been set up, Pathloss 5.0 can be used to simulate the propagation of radio signals between the two ends and predict the expected signal strength, quality, and coverage area of the link.

The software uses advanced algorithms to model the effects of various factors such as terrain, building clutter, and atmospheric conditions on radio propagation, allowing users to optimize the placement and configuration of transmitters and receivers for maximum performance. In addition, Pathloss 5.0 provides tools for analyzing link performance, including link budget analysis and interference analysis, which allow users to fine-tune the transmitter and receiver parameters and improve link performance.

To simulate the wireless links, the pathloss software was launched on a computer and a new project environment was created. The transmitter and receiver parameters were set up by specifying locations, antenna height, antenna gain, transmitter power, and frequency. Digital Terrain data was imported and the propagation model was defined. Finally, the pathloss was calculated and a link budget was generated for the wireless link.

### 3.2.2 Data validation

The free space path loss (FSPL) equation is employed to estimate the attenuation of a radio signal as it moves through space, whereas the received signal level (RSL) equation is utilized to determine the strength of a radio signal that a receiver antenna receives in a wireless communication system. The RSL is crucial in evaluating the dependability and quality of the communication link and is often compared to the minimum signal level required for the communication system to work properly. These equations can be represented as shown in Eq. (1) and Eq. (2) [35] and are used to generate the sample raw data in Table 4 and Table 5.

$$L_{FSL} = 92.45 + 20\log(f_{GHz}) + 20\log(d_{km}) \text{ [dB]} \quad (1)$$

where,

$f$  = the frequency of the signal in gigahertz frequency (GHz)

$d$  = the distance between the transmitter and receiver in(km)

Similarly,

$$\text{Received signal strength (RSL)} = P_o - L_{ctx} + G_{atx} - L_{crx} + G_{arx} - FSL - L_m \text{ [dBm]} \quad (2)$$

where,

$P_o$  = the transmitted power in (dBm)

$L_{ctx}$  = Transmitter antenna loss (dB)

$G_{atx}$  = the gain of the transmitting antenna (dBi)

$L_{crx}$  = Receiver antenna loss (dB)

$G_{arx}$  = the gain of the receiving antenna (dBi)

$L_m$  = system losses (dB)

FSL = free-space pathloss (dB)

Using the simulation parameters from Table 2 and Eqs. (1)-(2) to validate the parameters.

$$F = 10\text{GHz and } d = 1.71\text{km}$$

$$\text{Free space pathloss (FSPL)} = 92.45 + 20\log(10) + 20\log(1.71) \quad (1.71)$$

$$FSPL = 92.45 + 20 + 4.6599 = 117.11\text{dB}$$

Similarly, with Eq. (3)

$$\text{Received Signal Level (RSL)} = P_o - L_{ctx} + G_{atx} - L_{crx} + G_{arx} - FSL - L_m \quad (3)$$

where,

$$P_o = 21\text{dBm,}$$

$$L_{ctx} = 2.43\text{dB}$$

$$G_{atx} = 16\text{dBi}$$

$$L_{crx} = 0.00\text{dB}$$

$$G_{arx} = 33.9\text{dBi}$$

$$FSL = 117.11\text{dB}$$

$$L_m = 0.02\text{dB}$$

$$RSL = 21 - 2.43 + 16 - 0.00 + 33.9 - 117.11 - 0.02 = -48.66\text{dBm}$$

In general, the mathematical model for simulating the RSL values is shown in Eq. (4).

$$RSL = P_o - L_{ctx} + G_{atx} - L_{crx} + G_{arx} - (92.45 + 20\log(f_{GHz}) + 20\log(d_{km})) - L_m \quad (4)$$

**Table 4.** Sample generated training dataset

Node ID	Path Length (km)	Free Space Path Loss (dB)	Receive Signal Level (dBm)
100	0.1	92.45	-24
101	0.11	93.2778537	-24.8278537
102	0.12	94.03362492	-25.58362492
...	...	.....	.....
119	0.29	101.69796	-33.24795996
120	0.3	101.9924251	-33.54242509

**Table 5.** Sample generated test dataset

Node ID	Path Length (km)	Free Space Path Loss (dB)	Receive Signal Level (dBm)
100	0.1	92.45	-29.7
101	0.11	93.2778537	-30.5278537
102	0.12	94.03362492	-31.28362492
...	...	.....	.....
119	0.29	101.69796	-38.94795996
120	0.3	101.9924251	-39.24242509

### 3.2.3 Data understanding

To conduct the research, simulated data was generated using pathloss 5.0 software, and a mathematical model obtained from the RSL transmission equations was used to

verify the accuracy of the dataset. This additional step of validation using the RSL transmission equations helped ensure the reliability of the results before utilizing them in machine learning models. The RSL dataset comprises 5000 instances with 10 attributes, such as customer node IDs, distance (in kilometers), transmitting frequency (in GHz), free space path loss (in dB), transmit power (in dBm), transmit antenna gain (in dBi), transmitter loss (in dB), miscellaneous losses, remote antenna gain (in dBi), and received signal level (in dBm).

### 3.2.4 Data preprocessing

Data preprocessing is a fundamental technique aimed at reducing the complexity of data for easier processing. It covers the entire process of preparing data for analysis, including data wrangling and other tasks such as scaling and normalizing data, converting categorical variables to numerical variables, and dividing data into training and validation sets [36, 37].

### 3.3 Feature selection and extraction

It has been demonstrated that feature selection for high-dimensional data analysis is effective and advantageous by selecting a subset of important characteristics from a dataset using feature selection.

The objective of selecting features from a given set of data is to choose the most insightful and pertinent features for a specific issue while lowering the data's dimensions and computing complexity [38]. On the other hand, feature extraction typically transforms the raw data into easily recognizable features [39]. For this investigation, the six labelled input features indicated in Table 6 together with their data types have been chosen.

**Table 6.** Input features

Input Features	Data Types
Free Space Pathloss (dB)	float (64)
Transmit Power (dBm)	int (64)
Transmitter Antenna Gain (dBi)	int (64)
Transmitter Loss (dB)	float (64)
Miscellaneous Losses	float (64)
Remote Antenna Gain (dBi)	float (64)

### 3.4 Data normalization

Data standardization was performed on the dataset using Standard Scaler library in Python before incorporating the input into the ML algorithms. The main idea is to standardize the data by adjusting its scale and distribution so that each value has a mean of 0 and a standard deviation of 1 as shown in Eq. (5):

$$z = \frac{X - \mu}{\sigma} \quad (5)$$

where,

Z: the standard score

X: the raw score or observation being standardized

$\mu$ : the mean of the data

$\sigma$ : the standard deviation of the data

#### 3.4.1 Data division

Nguyen et al. [40] did a study where the Performance evaluation of several machine learning algorithms is done by partitioning the dataset into different proportions for training

and testing. The RMSE, MAE, and R-squared metrics were used to assess the model's effectiveness and gauge each metric's predictive potential. It was observed that the training/testing ratios of 70/30 outperformed other ratios. The train-test split's aim is to ensure that the model is trained and tested on various data sets in order to avoid overfitting [41].

### 3.5 Model selection

The act of choosing the most appropriate statistical or machine learning model for a specific problem or dataset is known as model selection. It is an important step in the machine learning modeling process, as the choice of models can greatly impact the accuracy and interpretability of the results. Received signal level prediction is a supervised learning regression problem because the predicted output is continuous-valued [39]. Hence, neural networks, support vector regression (SVR), K-NN, and random forest regression have shown substantial performance in received signal level predictions [42-46].

#### 3.5.1 Hyperparameter tuning

A very common approach for hyperparameter tuning is grid search, where a range of hyperparameters are specified for model training and evaluation with each hyperparameter combined together. Random search is another approach, where a random subset of hyperparameters is chosen and the model is trained and evaluated with those hyperparameters. There are also more advanced methods such as Bayesian optimization and gradient-based optimization. However, the process to tune the hyperparameter can be time-consuming and computationally expensive, especially for large and complex models [47-50].

#### 3.5.2 Model development

The Python and machine learning libraries were imported into the Jupyter notebook environment and stored in the same directory as the 'RSL\_Data.csv' file for preprocessing. The RSL data was split into features and a target variable, with input values comprising the features and the RSL values as the target (output) variable. The features were standardized by scaling them to have a mean of 0 and a standard deviation of 1.

The data was split into two subsets, using the `train_test_split` function. One subset was used for training the machine learning model, and the other subset was used for evaluating the performance of the model. The split was random and involved using 70% of the data for training and the remaining 30% for validation. A random seed value was also set, which ensured that the dataset was split consistently each time the code was executed. This level of consistency is crucial for reproducibility since it guarantees that the same random split is used during each execution.

For each classifier, an instance was created, and the models were fitted to the provided training data. After being trained via the `fit` method, the model was employed to make predictions on new data using the `predict` method. Additionally, the performance of the trained model was assessed on the validation dataset, which was used to fine-tune the model's hyperparameters.

To fine-tune the model's performance, the `GridSearchCV` class was employed. This class exhaustively searches a specified hyperparameter space, which is specified as a list of dictionaries. The `GridSearchCV` class receives inputs used to

evaluate the performance of the model for each hyperparameter combination. It then carries out a cross-validation procedure, utilizing the specified number of folds, to estimate the model's performance for each hyperparameter combination. The result of the GridSearchCV is a fitted GridSearchCV object, which contains information about the search process, the estimator object with the best hyperparameters found during the search. Using domain knowledge, the trained model was applied to a new testing dataset to generate predictions, which were then classified as indicating either a fault or no fault considering the rule-based algorithm that had been defined.

### 3.6 Performance evaluation metrics

Based on the literature [51-53], different approaches exist to assess the performance of the proposed models. In this study, the target variable is a continuous parameter;  $R^2$  (which indicates the goodness of fit between the observed and predicted values), the mean squared error (MSE), the mean absolute error and the root mean squared error have been used to evaluate the performance of the models as shown in Eq. (6) to Eq. (9).

Lower values for RMSE, MSE, MAE and higher values for  $R^2$  are indicative of a better predictive performance per machine learning approach.

Where,  $N$  represents the number of samples,  $f_i$  is the actual value predicted by the model,  $y_i$  denotes the predicted value and  $\bar{y}$  is mean value of  $y_i$ .

$$R^2 = 1 - \frac{\sum_{i=1}^N (y_i - f_i)^2}{\sum_{i=1}^N (y_i - \bar{y})^2} \quad (6)$$

$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - f_i)^2 \quad (7)$$

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - f_i)^2} \quad (8)$$

$$MAE = \frac{1}{N} \sum_{i=1}^N |y_i - f_i| \quad (9)$$

## 4. RESULTS AND DISCUSSION

### 4.1 Generated/acquired dataset

The RSL dataset utilized for this research consist of 5000 instances and 10 attributes. These attributes consist of customer node IDs, distance (measured in kilometers), transmitting frequency (in GHz), free space path loss (measured in dB), transmit power (measured in dBm), transmit antenna gain (measured in dBi), transmitter loss (measured in dB), miscellaneous losses, remote antenna gain (measured in dBi), and received signal level (measured in dBm). The free space pathloss column contains formula, where:

$$FSPL = 92.45 + 20 * LOG(C2) + 20 * LOG(B2).$$

C2 and B2 represents the frequencies and distances between the transmitter and receiver. Similarly, the frequency column is an aggregate of column (E3-G3+F3+I3-D3-H3).

The dataset is available for download at:

[https://docs.google.com/spreadsheets/d/1isu81PG1\\_cml8-WTPNtfwa\\_H5ZnPPQf6/edit?usp=share\\_link&ouid=111742986142993233907&rtpof=true&sd=true](https://docs.google.com/spreadsheets/d/1isu81PG1_cml8-WTPNtfwa_H5ZnPPQf6/edit?usp=share_link&ouid=111742986142993233907&rtpof=true&sd=true).

The setting of the threshold values for the RSSI for no fault is -70dBm above, while for fault detection is below -70dBm. According to international standard, RSSI greater than -70dBm has an excellent signal quality, while between -70dBm to -85dBm is good. Then between -86dBm to -100dBm is fair, below -100dBm is poor and -110 dBm has no signal.

### 4.2 Model comparison

Table 7 provides the performance metrics of three different models, namely Gradient Boosting Regression (GBR), Random Forest Regression (RFR), and K-Nearest Neighbors Regression (KNN), on a testing set of data. The values in the table are as follows:

Gradient Boosting Regression (GBR) has an R-squared value of 0.998686, an MAE of 0.221628, an MSE of 0.095309, and an RMSE of 0.308721.

Random Forest Regression (RFR) has an R-squared value of 0.999992, an MAE of 0.007101, an MSE of 0.00061, and an RMSE of 0.024697.

K-Nearest Neighbors Regression (KNN) has an R-squared value of 0.999989, a MAE of 0.007402, an MSE of 0.00077, and an RMSE of 0.027741.

**Table 7.** Model performance metrics on testing dataset

Model	Testing Metrics			
	R-Squared	MAE	MSE	RMSE
GBR	0.998686	0.221628	0.095309	0.308721
RFR	0.999992	0.007101	0.00061	0.024697
KNN	0.999989	0.007402	0.00077	0.027741

A comparison of the R-squared values, Mean Absolute Error (MAE), The MSE value and RMSE values shows that the RFR and KNN models have significantly lower RMSE values compared to the GBR model. This indicates that the RFR and KNN models have a better fit to the data and are making more accurate predictions compared to the GBR model.

In summary, based on the values in the Table 7, it appears that the Random Forest Regressor (RFR) model has the best overall performance, as it has the highest R-squared value (0.999992) and the lowest MAE, MSE, and RMSE values.

Similar comparison was performed on the Gradient Boosting Regression (GBR), Random Forest Regression (RFR), and K-Nearest Neighbors Regression (KNN), using the validation dataset shown in Table 7. The values in the table are as follows:

Gradient Boosting Regression (GBR) has an R-squared value of 0.998677, an MAE of 0.211548, an MSE of 0.092343, and an RMSE of 0.303879.

Random Forest Regression (RFR) has an R-squared value of 0.999998, an MAE of 0.002090, an MSE of 0.000108, and an RMSE of 0.010371.

K-Nearest Neighbors Regression (KNN) has an R-squared value of 0.999992, an MAE of 0.007231, an MSE of 0.000564, and an RMSE of 0.023739.

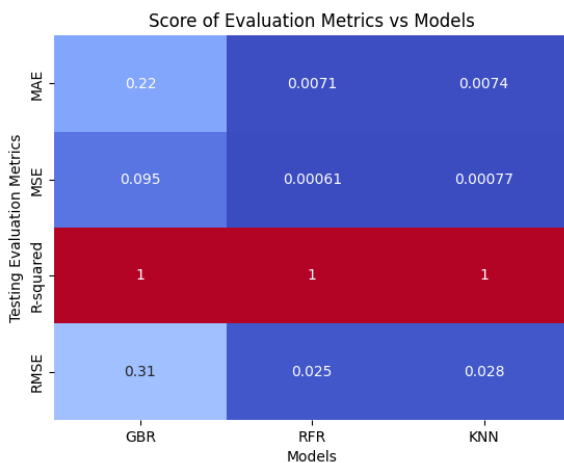
In summary, based on the values in the Table 8, it appears that the Random Forest Regressor (RFR) model has the best overall performance, as it has the highest R-squared value (0.999998) and the lowest MAE, MSE, and RMSE values on the validation dataset.

**Table 8.** Model performance metrics on validation dataset

Model	Validation Set Data Metrics			
	R-Squared	MAE	MSE	RMSE
GBR	0.998677	0.211548	0.092343	0.303879
RFR	0.999998	0.002090	0.000108	0.010371
KNN	0.999992	0.007231	0.000564	0.023739

**4.3 Heatmaps using the testing and validation dataset**

As shown from the heatmap in Figure 2, the R-squared values for all three models are 1, which indicates that all three models provide a perfect fit to the data. The MAE values are 0.22 for GBR, 0.0071 for RFR, and 0.0074 for KNN. This suggests that the RFR model has the smallest MAE and therefore the best performing classifier when comparing their individual mean absolute values. MSE values of 0.095 for GBR, 0.00061 for RFR, and 0.00077 for KNN means that the random forest regression model with smallest value has the best performance. RMSE values of 0.31 for GBR, 0.025 for RFR, and 0.028 for KNN also suggests that the RFR model has the smallest RMSE and therefore performs best. Based on the values of the evaluation metrics, it shows that the RFR model has the best performance among the three models using the testing dataset. Similar comparison of the performance metrics on the validation dataset shows that the random forest regression model performed best with 'MAE': 0.002090, 'MSE': 0.000108, 'R-squared': 0.999998, and 'RMSE': 0.010371 as shown in Figure 3.

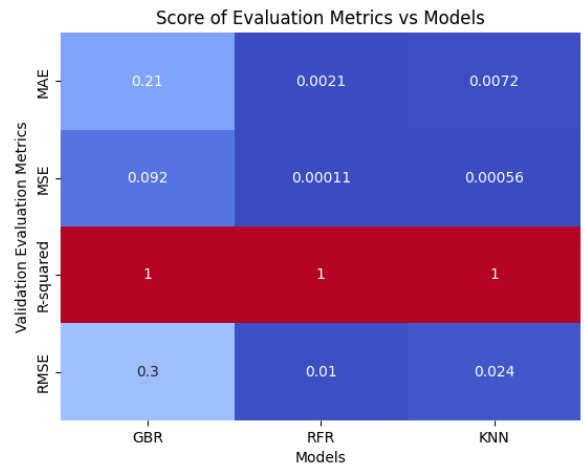


**Figure 2.** Heatmap for testing dataset

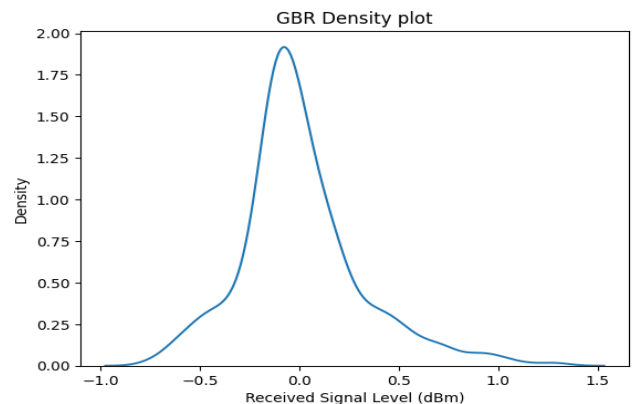
**4.3.1 Density plots**

Figure 4 shows the distribution of residuals from the gradient boosting regression model. The density plot of the received signal levels of the wireless radios is skewed, with the center of the plot not located at zero. This suggests that the residuals from the machine learning model are not normally distributed, and there is a systematic error in the model's predictions. This suggests that the model is not making accurate predictions for the received signal levels, and there may be issues with the model's fit to the data. The skewed shape of the plot and the lack of consistency in the residuals suggest that there are features that are not captured by the model. The fact that the plot in Figure 5 is symmetrical with one peak centered at 0 suggests that the residuals are normally distributed around 0, which is a desirable characteristic for a well-performing machine learning model. The narrow spread indicates that the residuals have low variability, which could

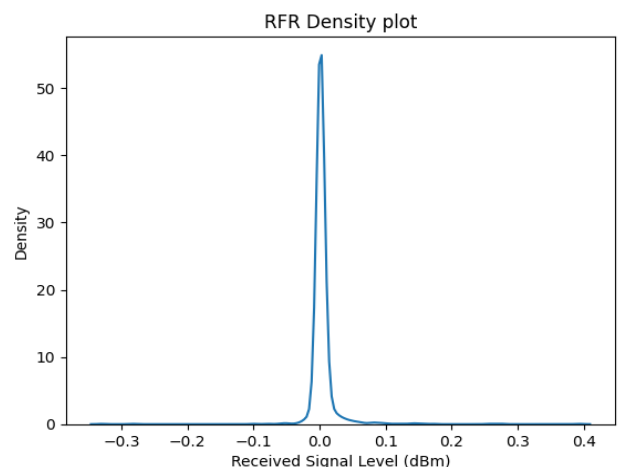
mean that the model is making accurate predictions. This is consistent with the idea that the model is making accurate predictions with low variability in the residuals. Figure 6 depicts the k-NN model that was trained on wireless radio received signal levels data. It also appears that the density plot of the received signal levels of the wireless radios is symmetrical with a single peak, indicating that the residuals from the model are normally distributed. The center of the plot being at 0 suggests that the received signal levels were accurately predicted. The narrow spread between suggests that the residuals have low variance and are consistent, which is a desirable property for a machine learning model.



**Figure 3.** Heatmap for validation dataset

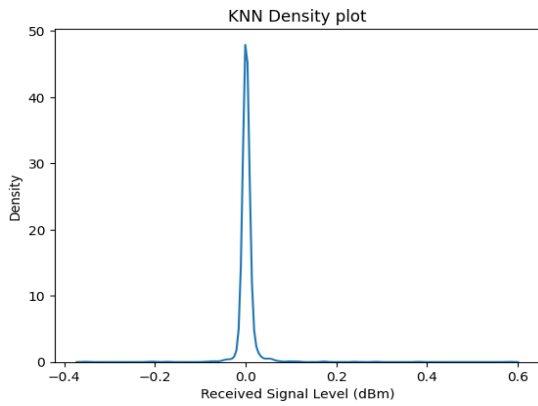


**Figure 4.** Density plot of GBR

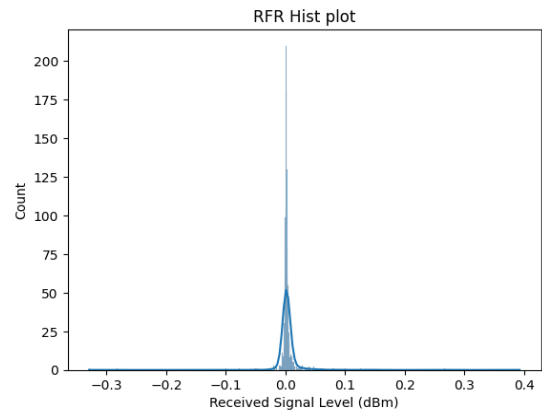


**Figure 5.** Density plot of RFR

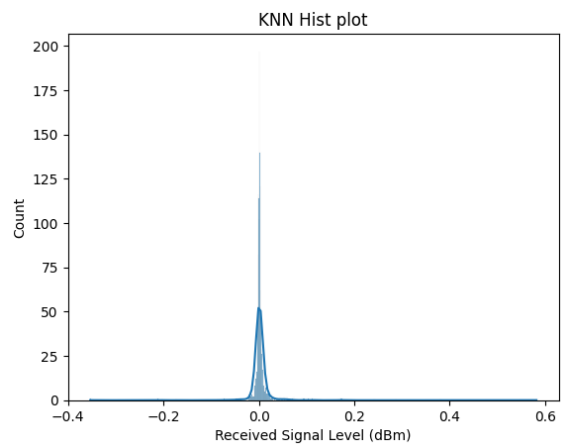




**Figure 6.** Density plot of KNN



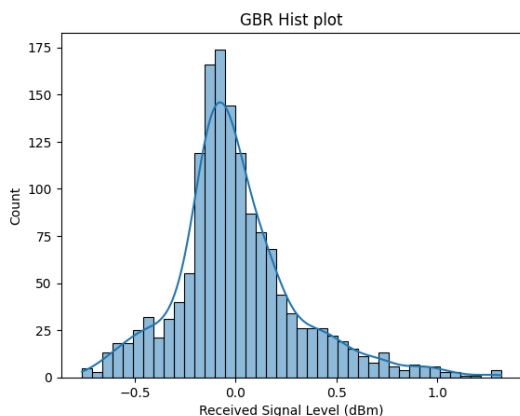
**Figure 8.** Histogram plot for RFR



**Figure 9.** Histogram plot for KNN

#### 4.3.2 Histogram and KDE plots

A histogram plot shows the distribution of residuals which represents the errors made by the model in its predictions while the KDE plots show an estimate of the probability density functions of the residuals. A normally distributed residuals would suggest that the model is making accurate predictions, on average. The center of the histogram plot should be close to zero, and the distribution should be symmetrical. If the residuals are not normally distributed, it suggests that the model is making systematic errors in its predictions and may need to be re-evaluated. If the residuals are evenly spread out and the KDE plot is symmetrical, it suggests that the model is making accurate predictions. If the residuals are not evenly spread out and the KDE plot is skewed, it suggests that the model is not making accurate predictions for certain ranges of the target values, and this may be a sign of overfitting or underfitting. Figure 7 depicts a non-symmetrical histogram for the GBR model with multiple peaks. This may indicate that the model is not accurately reporting the data. The ideal residual distribution should have a mean of 0 and a constant variance. The histogram in question has a non-zero mean and is skewed, which means it is not symmetrical. Figure 8 displays the random forest regression model's histogram and KDE plot, with a symmetrical histogram centered at 0. This implies that the residuals have a mean of 0, which shows that the model is, on average, producing unbiased predictions. A tall peak signifies that the model is making accurate prediction for those values and capturing the underlying patterns in the data. Figure 9 shows the histogram and the KDE plot of the KNN regression model with a symmetrical histogram centered around 0 suggest that the model is performing reasonably well.



**Figure 7.** Histogram plot for GBR

#### 4.4 The industry relevance of the research, result and work

Received signal level deviation model is an approach used to detect faults in telecommunication networks. It involves analyzing the deviation of received signal levels from expected values (RSSI of -70dBm and above) to identify any anomalies that may indicate a fault (RSSI below -70dBm). The relevance of this model in the telecommunication industry lies in its ability to quickly detect faults in a network, which is critical for maintaining high-quality communication services. By identifying and addressing faults early, service providers can minimize downtime, improve network performance, and enhance customer satisfaction.

The model works by analyzing the difference between the actual received signal level and the expected value for a given network element. If the deviation exceeds a predetermined threshold, it is flagged as a potential fault. The model can be applied to various network elements, such as base stations, antennas, and transmission lines. The results of using the model for fault detection in a telecommunication network can be significant. By detecting faults early, service providers can reduce the time and resources required to fix the issue, minimizing the impact on customers. Additionally, the model can help service providers identify patterns and trends in network faults, allowing them to implement preventive measures to reduce the likelihood of future issues.

#### 5. CONCLUSIONS

In this study, the efficacy of machine learning in monitoring

degraded radio links using received signal level (RSL) parameters has been performed. This research demonstrates that employing machine learning models enables precise identification of RSL deterioration through the use of RSL datasets. To identify the optimal classifier that fits the model, a dataset is trained using a machine learning classifier. Various base classifiers are evaluated based on the obtained results. The findings indicate that Random Forest Regression performs better than Gradient Boosting Regression and K-Nearest Neighbors in R-squared, MAE, MSE, and RMSE metrics. The proposed method for telecommunication maintenance using the deviation of received signal level can be used to reduce time to provide maintenance activities to customers.

The research contribution of this study is the development of a novel fault detection model for received signal level in telecommunication enterprise infrastructure using machine learning techniques. The modeling of an enterprise point to multipoint wireless communication network using pathloss 5.0 software and the extraction of data from the vector images of the simulated wireless network is a significant contribution to the field. The application of the gradient boosting regression (GBR), random forest regression (RFR) and K-Nearest Neighbor (KNN) regression models to the extracted dataset is another contribution, as it provides a comprehensive evaluation of the performance of the developed fault detection model. The proposed algorithm's ability to trigger a "Fault" or "No-fault" condition by comparing a threshold value with the received signal levels (RSL) of new and unseen dataset is an essential contribution, as it facilitates the timely maintenance of the wireless link. The use of the developed model can contribute to the improvement of telecommunication enterprise infrastructure by providing an efficient and reliable fault detection system for received signal levels. This research also contributes to the field of wireless communication by presenting a novel approach for fault detection using machine learning techniques.

Overall, the proposed fault detection model has the potential to be used in real-world telecommunication enterprise infrastructure to improve the reliability and efficiency of wireless links. It can help reduce maintenance costs, minimize downtime, and improve the overall performance of telecommunication networks.

## REFERENCES

- [1] Mahmood, T., Munir, K. (2020). Enabling predictive and preventive maintenance using IoT and big data in the telecom sector. In 5th International Conference on Internet of Things, Big Data and Security, pp. 169-176. <https://doi.org/10.5220/0009325201690176>
- [2] Kibria, M.G., Nguyen, K., Villardi, G.P., Zhao, O., Ishizu, K., Kojima, F. (2018). Big data analytics, machine learning, and artificial intelligence in next-generation wireless networks. *IEEE Access*, 6: 32328-32338. <https://doi.org/10.1109/ACCESS.2018.2837692>
- [3] Cheng, J.C.P., Chen, W., Chen, K., Wang, Q. (2020). Data-driven predictive maintenance planning framework for MEP components based on BIM and IoT using machine learning algorithms. *Automation in Construction*, 112: 103087. <https://doi.org/10.1016/j.autcon.2020.103087>
- [4] Ciocoiu, L., Siemieniuch, C.E., Hubbard, E.M. (2017). From preventative to predictive maintenance: The organisational challenge. *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit*, 231(10): 1174-1185. <https://doi.org/10.1177/0954409717701785>
- [5] Sakib, N., Wuest, T. (2018). Challenges and opportunities of condition-based predictive maintenance: A review. *Procedia CIRP*, 78: 267-272. <https://doi.org/10.1016/j.procir.2018.08.318>
- [6] Wan, J.F., Tang, S.L., Li, D., Wang, S.Y., Liu, C.L., Abbas, H., Vasilakos, A.V. (2017). A manufacturing big data solution for active preventive maintenance. *IEEE Transactions on Industrial Informatics*, 13(4): 2039-2047. <https://doi.org/10.1109/TII.2017.2670505>
- [7] Wei, G.Z., Zhao, X.J., He, S.G., He, Z. (2019). Reliability modeling with condition-based maintenance for binary-state deteriorating systems considering zoned shock effects. *Computers & Industrial Engineering*, 130: 282-97. <https://doi.org/10.1016/j.cie.2019.02.034>
- [8] Jardine, A.K.S., Lin, D., Banjevic, D. (2006). A review on machinery diagnostics and prognostics implementing condition-based maintenance. *Mechanical Systems and Signal Processing*, 20(7): 1483-1510. <https://doi.org/10.1016/j.ymssp.2005.09.012>
- [9] Zenisek, J., Holzinger, F., Affenzeller, M. (2019). Machine learning based concept drift detection for predictive maintenance. *Computers & Industrial Engineering*, 137: 106031. <https://doi.org/10.1016/j.cie.2019.106031>
- [10] Boldt, M., Ickin, S., Borg, A., Kulyk, V., Gustafsson, J. (2021). Alarm prediction in cellular base stations using data-driven methods. *IEEE Transactions on Network and Service Management*, 18(2): 1925-1933. <https://doi.org/10.1109/TNSM.2021.3052093>
- [11] García, A.J., Toril, M., Oliver, P., Luna-Ramírez, S., Ortiz, M. (2020). Automatic alarm prioritization by data mining for fault management in cellular networks. *Expert Systems with Applications*, 158: 113526. <https://doi.org/10.1016/j.eswa.2020.113526>
- [12] Kastouni M.Z., Lahcen, A.A. (2020). Big data analytics in telecommunications: Governance, architecture and use cases. *Journal of King Saud University - Computer and Information Sciences*, 34(6). <https://doi.org/10.1016/j.jksuci.2020.11.024>
- [13] Rezaei, S., Radmanesh, H., Alavizadeh, P., Nikoofar, H., Lahouti, F. (2016). Automatic fault detection and diagnosis in cellular networks using operations support systems data. In *NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium*, Istanbul, Turkey, pp. 468-473. <https://doi.org/10.1109/NOMS.2016.7502845>
- [14] Cherrared, S., Imadali, S., Fabre, E., Gossler, G., Yahia, I.G.B. (2019). A survey of fault management in network virtualization environments: challenges and solutions. *IEEE Transactions on Network and Service Management*, 16(4): 1537-1551. <https://doi.org/10.1109/TNSM.2019.2948420>
- [15] De-La-Bandera, I., Palacios, D., Mendoza, J., Barco, R. (2020). Feature extraction for dimensionality reduction in cellular networks performance analysis. *Sensors*, 20(23): 6944. <https://doi.org/10.3390/s20236944>
- [16] Xie, J., Yu, F.R., Huang, T., Xie, R.C., Liu, J., Wang, C.M., Liu, Y.J. (2019). A survey of machine learning techniques applied to software defined networking

- (SDN): Research issues and challenges. In *IEEE Communications Surveys & Tutorials*, 21(1): 393-430. <https://doi.org/10.1109/COMST.2018.2866942>
- [17] Mathew, V., Toby, T., Singh, V., Rao, B.M., Kumar, M.G. (2020). Prediction of remaining useful lifetime (RUL) of turbofan engine using machine learning. In *2017 IEEE International Conference on Circuits and Systems (ICCS)*, Thiruvananthapuram, India, pp. 306–311. <https://doi.org/10.1109/ICCS1.2017.8326010>
- [18] Dash, S.S., Nayak, S.K., Mishra, D. (2021). A review on machine learning algorithms. *Smart Innovation, Systems and Technologies*, 153: 495-507. [https://doi.org/10.1007/978-981-15-6202-0\\_51](https://doi.org/10.1007/978-981-15-6202-0_51)
- [19] Sezer, E., Romero, D., Guedea, F., MacChi, M., Emmanouilidis, C. (2018). An industry 4.0-enabled low cost predictive maintenance approach for SMEs. In *2018 IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC)*, Stuttgart, Germany, pp. 1-8. <https://doi.org/10.1109/ICE.2018.8436307>
- [20] Jimenez, J.J.M., Schwartz, S., Vingerhoeds, R., Grabot, B., Salaün, M. (2020). Towards multi-model approaches to predictive maintenance: A systematic literature survey on diagnostics and prognostics. *Journal of Manufacturing Systems*, 56: 539-557. <https://doi.org/10.1016/j.jmsy.2020.07.008>
- [21] Wuest, T., Weimer, D., Irgens, C., Thoben, K.D. (2016). Machine learning in manufacturing: Advantages, challenges, and applications. *Production & Manufacturing Research*, 4(1): 23-45. <https://doi.org/10.1080/21693277.2016.1192517>
- [22] Carvalho, T.P., Soares, F.A.A.D.M.N., Vita, R., Francisco, R.P., Basto, P., Alcalá, S.G.S. (2019). A systematic literature review of machine learning methods applied to predictive maintenance. *Computers & Industrial Engineering*, 137: 106024. <https://doi.org/10.1016/j.cie.2019.106024>
- [23] Çınar, Z.M., Nuhu, A.A., Zeeshan, Q., Korhan, O., Asmael, M., Safaei, B. (2020). Machine learning in predictive maintenance towards sustainable smart manufacturing in industry 4.0. *Sustainability*, 12(19): 8211. <https://doi.org/10.3390/su12198211>
- [24] Lei, Y.G., Li, N.P., Guo, L., Li, N.B., Yan, T., Lin, J. (2018). Machinery health prognostics: A systematic review from data acquisition to RUL prediction. *Mechanical Systems and Signal Processing*, 104: 799-834. <https://doi.org/10.1016/j.ymssp.2017.11.016>
- [25] Uddin, S., Khan, A., Hossain, M.E., Moni, M.A. (2019). Comparing different supervised machine learning algorithms for disease prediction. *BMC Medical Informatics and Decision Making*, 19: 281. <https://doi.org/10.1186/s12911-019-1004-8>
- [26] Cardoso, D., Ferreira, L. (2021). Application of predictive maintenance concepts using artificial intelligence tools. *Applied Sciences*, 11(1): 18. <https://doi.org/10.3390/app11010018>
- [27] Sun, B., Zeng, S.K., Kang, R., Pecht, M.G. (2012). Benefits and challenges of system prognostics. *IEEE Transactions on Reliability*, 61(2): 323-335. <https://doi.org/10.1109/TR.2012.2194173>
- [28] Schwendemann, S., Amjad, Z., Sikora, A. (2021). A survey of machine-learning techniques for condition monitoring and predictive maintenance of bearings in grinding machines. *Computers in Industry*, 125: 103380. <https://doi.org/10.1016/j.compind.2020.103380>
- [29] Sajid, S., Haleem, A., Bahl, S., Javaid, M., Goyal, T., Mittal, M. (2021). Data science applications for predictive maintenance and materials science in context to Industry 4.0. *Materials Today Proceedings*, 45: 4898-4905. <https://doi.org/10.1016/j.matpr.2021.01.357>
- [30] Calabrese, M., Cimmino, M., Fiume, F., Manfrin, M., Romeo, L., Ceccacci, S., Paolanti, M., Toscano, G., Ciandrini, G., Carrotta, A., Mengoni, M., Frontoni, E., Kapetis, D. (2020). SOPHIA: An event-based IoT and machine learning architecture for predictive maintenance in industry 4.0. *Information*, 11(4): 202. <https://doi.org/10.3390/INFO11040202>
- [31] Susto, G.A., Beghi, A., Luca, C.D. (2012). A predictive maintenance system for epitaxy processes based on filtering and prediction techniques. *IEEE Transactions on Semiconductor Manufacturing*, 25(4): 638-649. <https://doi.org/10.1109/TSM.2012.2209131>
- [32] Verhagen, W.J.C., Boer, L.W.M.D. (2018). Predictive maintenance for aircraft components using proportional hazard models. *Journal of Industrial Information Integration*, 12: 23-30. <https://doi.org/10.1016/j.jii.2018.04.004>
- [33] Tounsi, Y., Anoun, H., Hassouni, L. (2020). CSMAS: Improving multi-agent credit scoring system by integrating big data and the new generation of gradient boosting algorithms. In *Proceedings of the 3rd International Conference on Networking, Information Systems & Security*, pp. 1-7. <https://doi.org/10.1145/3386723.3387851>
- [34] Munirathinam, S., Ramadoss, B. (2016). Predictive models for equipment fault detection in the semiconductor manufacturing process. *International Journal of Engineering and Technology*, 8(4): 273-285. <https://doi.org/10.7763/ijet.2016.v8.898>
- [35] Zemanian, A.H., Chan, Y.R., Chang, V.A. (1999). Transmission networks. *International Journal of Circuit Theory and Applications*, 27(3): 293-301. [https://doi.org/10.1002/\(SICI\)1097-007X\(199905/06\)27:3<293::AID-CTA51>3.0.CO;2-N](https://doi.org/10.1002/(SICI)1097-007X(199905/06)27:3<293::AID-CTA51>3.0.CO;2-N)
- [36] Luengo, J., García-Gil, D., Ramírez-Gallego, S., García, S., Herrera, F. (2020). *Big Data Preprocessing: Enabling Smart Data*. Springer Cham. <https://doi.org/10.1007/978-3-030-39105-8>
- [37] Ramírez-Gallego, S., Krawczyk, B., García, S., Woźniak, M., Herrera, F. (2017). A survey on data preprocessing for data stream mining: Current status and future directions. *Neurocomputing*, 239: 39-57. <https://doi.org/10.1016/j.neucom.2017.01.078>
- [38] Cai, J., Luo, J.W., Wang, S.L., Yang, S. (2018). Feature selection in machine learning: A new perspective. *Neurocomputing*, 300: 70-79. <https://doi.org/10.1016/j.neucom.2017.11.077>
- [39] Zhang, Y., Wen, J.X., Yang, G.S., He, Z.W., Wang, J. (2019). Path loss prediction based on machine learning: Principle, method, and data expansion. *Applied Sciences*, 9(9): 1908. <https://doi.org/10.3390/app9091908>
- [40] Nguyen, Q.H., Ly, H.B., Ho, L.S., Al-Ansari, N., Le, H.V., Tran, V.Q., Prakash, I., Pham, B.T. (2021). Influence of data splitting on performance of machine learning models in prediction of shear strength of soil. *Mathematical Problems in Engineering*, 2021: 4832864. <https://doi.org/10.1155/2021/4832864>
- [41] Ying, X. (2019). An overview of overfitting and its

- solutions. *Journal of Physics: Conference Series*, 1168(2): 022022. <https://doi.org/10.1088/1742-6596/1168/2/022022>
- [42] Navabi, S., Wang, C., Bursalioglu, O.Y., Papadopoulos, H. (2018). Predicting wireless channel features using neural networks. In 2018 IEEE International Conference on Communications (ICC), Kansas City, MO, USA, pp. 1-6. <https://doi.org/10.1109/ICC.2018.8422221>
- [43] Wickramasuriya, D.S., Perumalla, C.A., Davaslioglu, K., Gitlin, R.D. (2017). Base station prediction and proactive mobility management in virtual cells using recurrent neural networks. In 2017 IEEE 18th Wireless and Microwave Technology Conference (WAMICON), Cocoa Beach, FL, USA. <https://doi.org/10.1109/WAMICON.2017.7930254>
- [44] Gideon, K., Nyirenda, C., Temaneh-Nyah, C. (2017). Echo state network-based radio signal strength prediction for wireless communication in Northern Namibia. *IET Communications*, 11(12): 1920-1926. <https://doi.org/10.1049/iet-com.2016.1290>
- [45] Li, S.Z., Wang, N., Du, X.H., Liu, A.D. (2019). Internet web trust system based on smart contract. *Communications in Computer and Information Science*, 1058: 295-311. [https://doi.org/10.1007/978-981-15-0118-0\\_23](https://doi.org/10.1007/978-981-15-0118-0_23)
- [46] Oroza, C.A., Zhang, Z., Watteyne, T., Glaser, S.D. (2017). A machine-learning-based connectivity model for complex terrain large-scale low-power wireless deployments. *IEEE Transactions on Cognitive Communications and Networking*, 3(4): 576-584. <https://doi.org/10.1109/TCCN.2017.2741468>
- [47] Okokpujie, K., Kennedy, G.C., Oluwaleye, S., John, S.N., Okokpujie, I.P. (2023). An overview of self-organizing network (SON) as network management system in mobile telecommunication system. *Information Systems for Intelligent Systems*, 324: 309-318. [https://doi.org/10.1007/978-981-19-7447-2\\_28](https://doi.org/10.1007/978-981-19-7447-2_28)
- [48] Agboje, O., Nkordeh, N., Oladoyin, O., Uzairue, S., Okokpujie, K., Bob-Manuel, I. (2020). MIMO CHANNELS: Optimizing throughput and reducing outage by increasing multiplexing gain. *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, 18(1): 441-448. <https://doi.org/10.12928/telkomnika.v18i1.8720>
- [49] Okokpujie, I.P., Odigilia, I.M., Okokpujie, K., Subair, R.E., Ogundipe, A.T., Tartibu, L.K., Ikumapayi, O.M. (2022). Influence of corporate social responsibility on business evaluation of mobile communication network MTN in Nigeria. *International Journal of Sustainable Development and Planning*, 17(7): 2199-2207. <https://doi.org/10.18280/ijstdp.170720>
- [50] Okokpujie, K., Okokpujie, I.P., Ogundipe, A.T., Anike, C.D., Asaboro, O.B., Vincent, A.A. (2023). Development of a sustainable Internet of Things-based system for monitoring cattle health and location with web and mobile application feedback. *Mathematical Modelling of Engineering Problems*, 10(3): 740-748. <https://doi.org/10.18280/mmep.1003023>
- [51] Okokpujie, K., Okokpujie, I.P., Ayomikun, O.I., Orimogunje, A.M., Ogundipe, A.T. (2023). Development of a web and mobile applications-based cassava disease classification interface using Convolutional Neural Network. *Mathematical Modelling of Engineering Problems*, 10(1): 119-128. <https://doi.org/10.18280/mmep.100113>
- [52] Ruiz-Sarmiento, J.R., Monroy, J., Moreno, F.A., Galindo, C., Bonelo, J.M., Gonzalez-Jimenez, J. (2020). A predictive model for the maintenance of industrial machinery in the context of industry 4.0. *Engineering Applications of Artificial Intelligence*, 87: 103289. <https://doi.org/10.1016/j.engappai.2019.103289>
- [53] Falamarzi, A., Moridpour, S., Nazem, M., Cheraghi, S. (2019). Prediction of tram track gauge deviation using artificial neural network and support vector regression. *Australian Journal of Civil Engineering*, 17(1): 63-71. <https://doi.org/10.1080/14488353.2019.1616357>