Learning to Fairly Classify the Quality of Wireless Links

Gregor Cerar*†, Halil Yetgin*‡, Mihael Mohorčič*†, Carolina Fortuna*
*Department of Communication Systems, Jožef Stefan Institute, SI-1000 Ljubljana, Slovenia.
†Jožef Stefan International Postgraduate School, Jamova 39, SI-1000 Ljubljana, Slovenia.
‡Department of Electrical and Electronics Engineering, Bitlis Eren University, 13000 Bitlis, Turkey.
{gregor.cerar | halil.yetgin | miha.mohorcic | carolina.fortuna}@ijs.si

***Abstract*—Machine learning (ML) has been used to develop increasingly accurate link quality estimators for wireless networks. However, more in depth questions regarding the most suitable class of models, most suitable metrics and model performance on imbalanced datasets remain open. In this paper, we propose a new tree based link quality classifier that meets high performance and fairly classifies the minority class and, at the same time, incurs low training cost. We compare the tree based model, to a multilayer perceptron non-linear model and two linear models, namely logistic regression and support vector machine, on a selected imbalanced dataset and evaluate their results using five different performance metrics. Our study shows that 1) non-linear models perform slightly better than linear models in general, 2) the proposed non linear tree-based model yields the best performance trade-off considering F1, training time and fairness, 3) single metric aggregated evaluations based only on accuracy can hide poor, unfair performance especially on minority classes, and 4) it is possible to improve the performance on minority classes, by over 40% through feature selection and by over 20% through resampling, therefore leading to fairer classification results.**

***Index Terms*—link quality estimation, machine learning, unbalanced data, fair classification, data-driven optimization, data preprocessing, feature selection.**

I. Introduction

Machine learning (ML) is becoming an increasingly popular way of solving various problems in communications in general and wireless networks in particular. Data driven link quality estimation (LQE) techniques where the researchers manually developed models have been proposed over the last two decades [1]–[3]. More recently, the manual model development is being automated, by using machine learning algorithms that approximate the distribution of the underlying random variable and are thus able to learn the quality of a link [4], [5].

LQE models developed using ML algorithms can estimate the quality of a link in a continuous-valued space by means of performing regression [4], [6]–[9]. Alternatively, if they estimate the link quality in a discrete-valued space, ML performs classification [5], [10]–[12]. By analyzing the existing body of literature developing classification models for LQE, we notice two types of approaches; i) *binary- or two-class*, ii) *multi-class*.

The *binary- or two-class* approach, can be found in [10], [11], [13] while *multi-class* approach appears in [5], [12], [14]–[16], where [14], [16] use a three-class, [17] utilizes a four-class, [12], [15] rely on a five-class, and [5] leverages a seven-class output. These applications are leveraged for the categorization and estimation of the future link state, which is expressed through labels/classes and it is not always clear from the related work how the authors select the number of classes. The binary-class works seem to be motivated by the application requirements, particularly of a multi-hop routing protocol that needs to know whether a link is reliable or not. The three-class approach seems to be motivated by the non-linear S-shaped curve with three regions specified for wireless links [18]. The seven-class output is motivated by the geographical environment over which the wireless network operates considering the application of coverage estimation [5].

An important aspect that is not previously considered in LQE classification and possibly neither in general classification problems for wireless communications is the fairness of the ML models developed for classification. However, maintaining fairness in multi-class classification problems has been a challenging issue, especially when an imbalanced dataset is considered [19]. To exemplify the significance of classification unfairness in real-life scenarios, Chouldechova *et al.* [20] show evidence of racial bias in the recidivism prediction tool, in which white defendants are less likely to be classified as high-risk than black defendants and Obermeyer *et al.* [21] show biases in the health care decision-making system in which black patients who are captured by the algorithm at the same risk level are sicker than white patients. Resembling these real-life classification problems to the wireless communication links, when no good links are available and the classifier is unable to recognize intermediate links as these usually belong to the minority class that is unfairly discriminated, the communication might be hindered by selecting a bad link. Therefore, it is important to justify whether the decision made by a ML model is fair to all considered link quality classes. *Against this background, we propose a decision tree-based ML model for LQE with the goal of attaining fairness between link quality classes, albeit with the least possible accuracy compromise, and compare this accuracy/fairness performance trade-off to other existing ML models.*

From the analysis of the literature discussed above, we draw the following observations:

ISBN 978-3-903176-35-5 © IFIP

Observation-1: ML based classification studies that use linear ML methods, such as logistic regression (LR) alongside non-linear methods, such as neural networks [6], [10] reveal small performance differences in the range of few percentage points on the three zone S-like shaped link quality curve. According to [4], link quality tends to be a non-linear function, thus non-linear models are likely to perform better for LQE. However, this aspect is not systematically investigated in the literature.

Observation-2: Most of the related works on classification evaluate their performance using the $accuracy$ metric and perhaps some other application-specific metric, such as routing tree stability or depth. Notable exceptions are [5], [12], where the authors present a full confusion matrix to be able to assess which classes are well discriminated by the model and which are often confused. However, it is well-known in the ML communities that accuracy is a misleading metric, especially for imbalanced datasets [22], where it can hide bias or unfairness towards the minority class [19].

Observation-3: The authors of [12] provide a great level of details in their methodology and in their results. Their confusion matrices reveal very strong performance on certain classes and higher confusion on others. Relatively poorer performance on intermediate classes may be due to the class imbalance on the training data. However, we are unable to see if this is the case with their training data and by looking at their process, no countermeasures, e.g. resampling techniques seem to be adopted as a remedy.

Following the three listed observations, we identify opportunities to contribute and extend the existing body of work on LQE using ML based classification, as follows.

- We propose a new tree based link quality classifier that meets high classification performance and fairly classifies also the minority class while, at the same time, incurring low training cost.
- We compare the proposed tree based model, to a multi-layer perceptron (MLP) non-linear model and two linear models, namely LR and support vector machine (SVM), on a selected imbalanced dataset and show that the proposed model takes about 90 times less training time compared to MLP and the performance compromise is less than $\approx 1\%$.
- We adopt standard metrics from the ML community to evaluate the performance of our classifier. In addition to $accuracy$, we also use $precision$, $recall$, $F1$ and, where necessary, the detailed $confusion\ matrix$ based on which all the other metrics are computed. To date, no other LQE classification work considered all five different metrics for a thorough performance evaluation that also considers per class fairness.
- We explicitly study and evaluate ways to improve minority class discrimination on imbalanced datasets for the sake of a fair classification performance on all link quality classes. For this purpose, we select a publicly available wireless dataset that is suitable for developing an LQE classifier and is imbalanced.

The rest of this paper is structured as follows. Section II summarizes related work while Section III defines the learning problem, including a preliminary for linear and non-linear ML-based models, dataset selection and methodology. Section IV elaborates on selecting the best features for training a model with high performance and fair per-class discrimination capabilities. Section V studies how to compensate for the class imbalance in the dataset to further improve per class fairness while Section VI evaluates the performance of the proposed model. Finally, in Section VII summarizes the paper and identifies future directions.

II. Related work

To the extent of our knowledge, this is the first attempt to develop a ML-based LQE model that considers classification fairness among the accounted wireless link quality classes. Moreover, there is only a paucity of contributions considering decision tree-based ML algorithms for LQE.

One of the first ML models for LQE is proposed by Liu *et al.* [6], in which they use the 4C algorithm to train three ML models based on naïve Bayes, neural networks, and logistic regression algorithms, which ultimately produces a multi-class output. Subsequently, Liu *et al.* [10] extend their work to an online ML model, namely TALENT, where the model built on each device adapts to newly generated data points instead of being pre-computed on a server, and consequently yields a binary threshold-based output.

Similarly, Shu *et al.* [15] use the SVM algorithm to develop a five-class link quality model, while Okamoto *et al.* [8] use an online learning algorithm called adaptive regularisation of weight vectors for learning to estimate throughput from images, and then Bote-Lorenzo *et al.* [9] train online perceptrons, online regression trees, fast incremental model trees, and adaptive model rules. The latter two models consider continues-valued output, which means that they are simply constrained by numerical precision due to regression. Demetri *et al.* [5] propose a seven-class SVM classifier to estimate LoRa network coverage, using multiple input metrics to train the classifier, including multispectral aerial imagery. Surprisingly, the only reinforcement learning-based approach for LQE is found in [7], where the authors train a greedy algorithm with multiple input metrics to estimate packet reception ratio (PRR) as a continuous-valued output in terms of protocol improvement in mobility scenarios.

Furthermore, two LQE models using deep learning algorithms have been proposed, where the first model [4] introduces a new LQE metric for estimating link quality in smart grid environments that relies on signal-to-noise ratio (SNR) while producing a continuous-valued PRR output. In the other model, Luo *et al.* [12] incorporate multiple input metrics and train neural networks to discriminate an LQE model with five classes.

None of the aforementioned works dealing with multi-class classification problems consider fairness among accounted classes and decision tree-based ML algorithms. Only in our recent work [23], we evaluate the performance of logistic

ISBN 978-3-903176-35-5 © IFIP

regression, three-based, ensemble, and multilayer perceptron algorithms for LQE with a three-class output and show that feature engineering has a larger impact on the final LQE model performance than the choice of ML algorithms. However, the fairness among the considered classes was not analysed in this particular work.

III. Definition of the learning problem

We aim to learn to discriminate among the widely-used three-class distinction model [18], i.e., *good*, *intermediate* and *bad* classes for a link. To achieve this, we leverage the selected dataset and the identified linear and non-linear ML algorithms, and train the algorithms with a subset of the available data. This way, a model that is able to discriminate among the three target classes is developed and its performance is then evaluated on the remaining data. To conduct our study and evaluate the performance of the proposed DTree and the other three models, we use the standard approach for developing a classifier: we first perform data pre-processing, then continue with model training and selection.

A. Linear and non-linear ML-based models

Machine learning algorithms are suitable for automatically approximating the underlying distribution that generated a set of measurements. They are particularly useful when there is no analytical formula that models the phenomenon generating the distribution and a large number of empirical observations can be collected. If the measurements are closer to a non-linear function, then non-linear ML algorithms such as decision trees are more suitable for approximating them. Otherwise, linear models such as logistic regression (LR) are preferred due to their simplicity and relatively lower computational complexity [24].

For linear ML-based LQE model development, we consider *logistic regression* as a subset of the general linear regression and *support vector machine (SVM) with linear kernel*. A logistic regression function enforces the output of the linear function to lie between the value of 0 and 1, where the classification (labeling) of link quality is conducted based on a predetermined threshold. This can be achieved by maximizing the probability of a random data point to be correctly classified relying on maximum likelihood, gradient descent or other optimization algorithms. Similarly, SVM with linear kernel produces a hyperplane or a line (depending on the number of features) that precisely classifies data points. The main idea of the SVM is to maximize the margin between respective data points that are closer to the hyperplane [24].

On the other hand, the considered non-linear ML-based LQE models are developed using *decision trees* (DTree) and *multilayer perceptron (MLP)*. A decision tree represents a non-linear mapping of the independent and dependent variables, which can be utilized for classifying data that is difficult to separate with linear methods [24]. MLP represent a subset of feedforward artificial neural networks composed of at least three layers of nodes, each of which is a neuron that utilizes a non-linear activation function. MLP can classify data that is not linearly distinguishable [24].

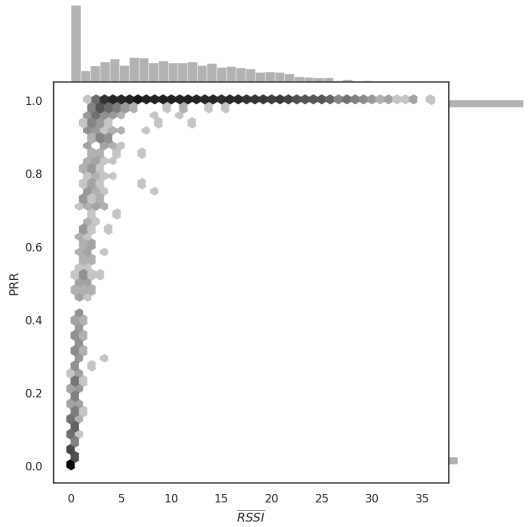


Fig. 1: PRR and average $\overline{\text{RSSI}}$ relationship for Rutgers trace-set (log-scale).

B. Trace-set selection

As discussed in Section I, the third aspect of our investigation requires an imbalanced dataset that is suitable for training a ML based LQE. We also prefer a publicly available dataset so that the research can be easily replicated. We have identified a number of such publicly available datasets, namely Roofnet [25], Rutgers [26], "packet-metadata" [27], University of Michigan [28], EVARILOS [29] and Colorado [30].

Roofnet [25] is a well known WiFi-based trace-set and contains the largest number of data points among the identified trace-sets, however PRR, as a target training metric for the classifier, can only be computed as an aggregate value per link without the knowledge of how the link quality varied over time. Rutgers is smaller than Roofnet, however is large enough to train a ML model and is appropriately formed for our purpose. The trace-set for each node contains raw received signal strength indicator (RSSI) value along with the sequence number.

Upon closer investigation for the remaining trace-sets, we concluded that they are not suitable for our intended purpose. The "packet-metadata" [27] comes with a plethora of features convenient for LQE research. In addition to the typical LQI and RSSI, it provides information about the noise floor, transmission power, dissipated energy as well as several network stacks and buffer related parameters. However, packet loss can only be observed in rare cases with very small packet queue length.

The trace-set from the University of Michigan [28] is somewhat incomplete and suffers from an inconsistent data format containing lack of units, missing sequence numbers and inadequate documentation. The two EVARILOS trace-sets [29] are mainly well-formatted, whereas each contains fewer than 2,000 entries. In the Colorado trace-setColorado [30], the

ISBN 978-3-903176-35-5 © IFIP

TABLE I: Global parameters for ML-based LQE models.

Step/Parameter	Default value
Missing data	Domain knowledge (zero-fill)
History window size ($W_{history}$)	10
Prediction window size (W_{PRR})	10
Features set	RSSI, $\overline{RSSI}_{10}$, $RSSI_{SD,10}$
Resampling strategy	Random oversampling (ROS)
Link quality labels	*Good*, *intermediate*, *bad*
Globally used ML algorithms	**Linear:** Linear (Logistic) **Non-linear:** Decision trees (DTree) with tree depth limited to 4, the min. samples per node set to 50
Cross-validation strategy	Randomize & 10-times Stratified K-Fold

diversity of the link performance is missing as all links seem to exhibit less than 1% packet loss.

After careful consideration we selected the "Rutgers trace-set" [26] as the candidate dataset for this work. The dataset was created using the ORBIT testbed and includes 4,060 distinct link traces, which are gleaned from 812 unique links with 5 different noise levels, i.e., 0, -5, -10, -15 and -20 dBm. Readily available trace-set features include raw RSSI, sequence numbers, source node ID, destination node ID and artificial noise levels. The packets are sent every 100 milliseconds for a period of 30 seconds, therefore, each trace is composed of 300 packets. Besides, based on the specifications of the radio used, each RSSI value is defined between 0 and 128, where the value of 128 indicates an error and is therefore invalid. A statistical analysis of the Rutgers trace-set reveals that 960 link traces out of 4,060 (23.65%) are entirely empty indicating no packets were received, and that a total of 1,218,000 packets were sent and only 773,568 (63.51%) were correctly received.

We plot in Fig. 1 the relationship between RSSI and the PRR computed based on the available sequence numbers. The darker hexagonal areas of Fig. 1 indicate that the majority of links are of either "poor quality" (bottom-left) or "good quality" (top), while gray areas are of "intermediate quality". The bars on the right hand side of the figure show the imbalanced nature of the dataset, more precisely, 61% of the links are *good*, 34% are *bad* and only 5% *intermediate*.

C. Experimental details

As a baseline reference model, we select the *majority classifier*, which in our case, classifies all the links in good quality class. In order to evaluate the most suitable ML-based LQE model, we utilize accuracy, precision, recall and F1 metrics, where precision indicates how precisely the model classifies links (high precision) and recall reveals how many relevant links were actually classified (high recall), while F1 is the harmonic mean of the former two. For our analysis, we include per class score values in parentheses for precision, recall and F1 values as in the following order: *good*, *intermediate* and *bad*. Then, these values in parenthesis are averaged using a *weighted average value per class* method to obtain precision, recall and F1 values, respectively. For the sake of providing a fair comparison, before any ML-based LQE model is developed, the dataset is shuffled and 10-times stratified K-Fold is employed to produce estimated classes [31]. For the development of ML-based LQE models, we utilize the global parameters of Table I throughout the paper, unless stated otherwise. $W_{history}$ in Table I represents the historical window that is utilized for calculating the features and W_{PRR} depicts the prediction window that is used for identifying the link quality labels. $\overline{RSSI}_{10}$ represents the averaged RSSI over 10 packets and $RSSI_{SD,10}$ represents the standard deviation of the RSSI over 10 packets. Missing values in the Rutgers are filled using the zero-filling technique, as outlined in Table I.

IV. The influence of feature selection on performance and fairness

Feature selection is the step in data preprocessing concerned with determining unprocessed features or creating synthetic features for the training of ML algorithms. Features can be conducted manually or produced by the aid of algorithms. The training feature available in our dataset is the raw RSSI value and the other is the sequence number that can be exploited for the limited time series analysis, and computation of PRR, on which the link quality classes depend. The arbitrary values associated to distinct classes, which were also set in [18], are defined in the form of the following rule:

$$y = f(\text{PRR}) = \begin{cases} \text{bad}, & \text{if PRR} \leq 0.1 \\ \text{intermediate}, & \text{otherwise} \\ \text{good}, & \text{if PRR} \geq 0.9, \end{cases} \quad (1)$$

$$\mathbf{y} = [y_1, y_2, \ldots, y_n], \quad \forall y \in \{\text{bad, intermediate, good}\}. \quad (2)$$

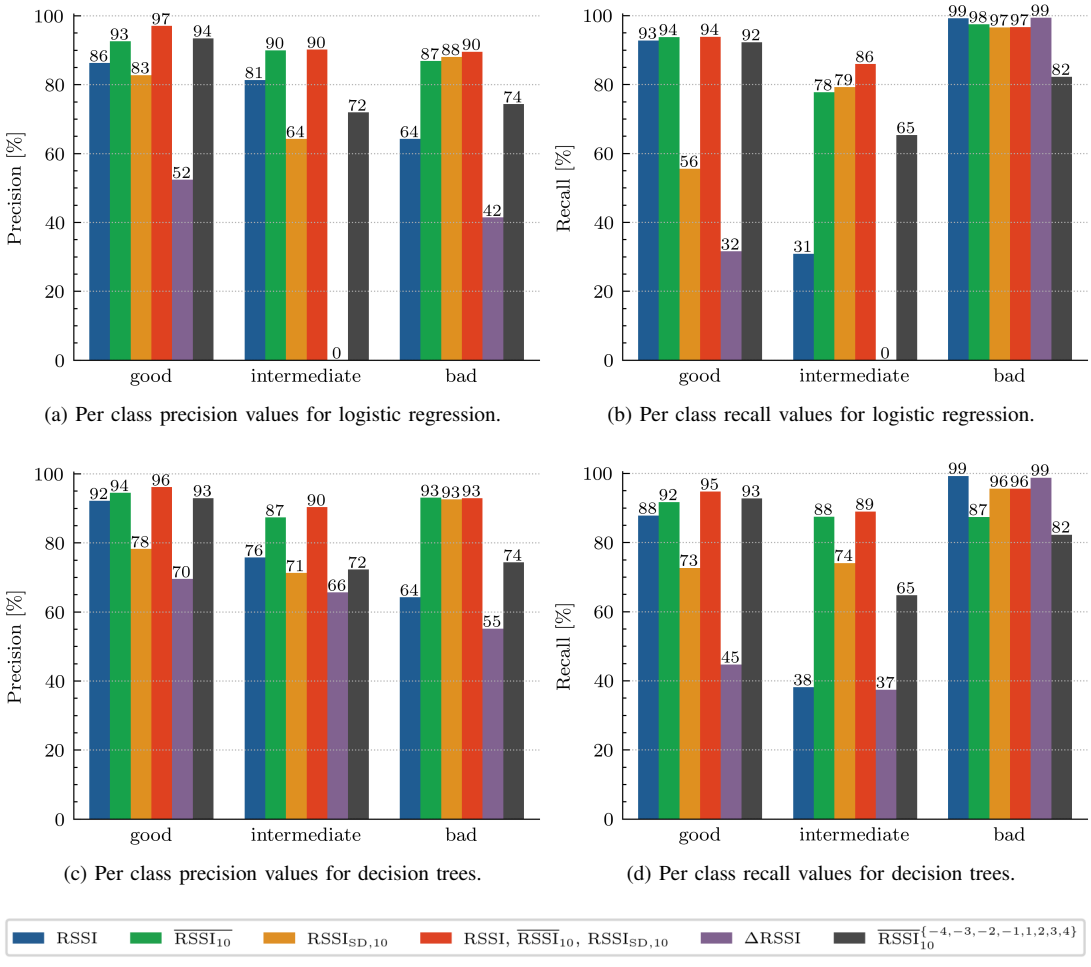
One of the widely-used approaches in ML for such trace-sets with small number of features is to examine whether synthetic features, such as average RSSI over a time window period or polynomial interactions [32], can assist the training to obtain more accurate models compared to that of the raw RSSI values. We study an extensive combination of features including 1) readily available RSSI, 2) averaged RSSI over 10 packets $\overline{RSSI}_{10}$, 3) standard deviation of RSSI over 10 packets $RSSI_{SD,10}$, 4) a combination of the three RSSI, $\overline{RSSI}_{10}$, $RSSI_{SD,10}$, derivate RSSI ΔRSSI ("left" derivative), and negative power of the averaged RSSI $\overline{RSSI}_{10}^{\{-4,-3,-2,-1,1,2,3,4\}}$ that are listed in Table II and present the influence of the best-performing set of feature combinations on the classification performance. The table evaluates how well the learned model predicts link quality as per Eq. (1) for the next prediction window W_{PRR}, while relying on the parameters of Table I.

The results show that using only $RSSI$ yields 74% *accuracy* for the linear model and 75% for the non-linear one as per the first line corresponding to each algorithm in Table II, while the F1 scores are about 70% and 72%, respectively, confirming the fact that *accuracy* overestimates the performance of the model on imbalanced datasets [22]. Breaking down into per class performance, it can be seen that F1 on the majority *good* class is 78% with a precision of 86% and recall of only 93% as also visually represented in Figures 2a and 2b. High precision

ISBN 978-3-903176-35-5 © IFIP

TABLE II: Comparison of various sets of features using linear and non-linear ML algorithms.

Algorithm	Feature set	Acc. [%]	Precision [%]	Recall [%]	F1 [%]
Linear (Logistic)	RSSI	74.4	77.3 (86.3, 81.4, 64.3)	74.4 (92.8, 30.9, 99.3)	70.8 (89.5, 44.8, 78.1)
	$\overline{\text{RSSI}}_{10}$	89.7	89.8 (92.6, 90.0, 86.9)	89.7 (93.8, 77.8, 97.5)	89.5 (93.2, 83.5, 91.9)
	$\text{RSSI}_{\text{SD},10}$	77.1	78.4 (82.8, 64.3, 88.1)	77.1 (55.6, 79.3, 96.6)	76.6 (66.5, 71.0, 92.1)
	RSSI, $\overline{\text{RSSI}}_{10}$, $\text{RSSI}_{\text{SD},10}$	**92.2**	**92.3** (97.1, 90.2, 89.6)	**92.2** (93.9, 86.0, 96.7)	**92.2** (95.5, 88.0, 93.0)
	ΔRSSI ("left" derivative)	43.7	31.3 (52.4, 0.0, 41.5)	43.7 (31.6, 0.0, 99.4)	32.7 (39.4, 0.0, 58.5)
	$\overline{\text{RSSI}}_{10}^{\{-4,-3,-2,-1,1,2,3,4\}}$	80.0	80.0 (93.5, 72.0, 74.4)	80.0 (92.3, 65.4, 82.3)	79.9 (92.9, 68.6, 78.1)
Non-linear (DTree)	RSSI	75.1	77.5 (92.2, 75.8, 64.3)	75.1 (87.8, 38.2, 99.3)	72.9 (90.0, 50.8, 78.1)
	$\overline{\text{RSSI}}_{10}$	91.6	91.6 (94.5, 87.4, 93.1)	91.6 (91.7, 87.5, 87.4)	91.6 (93.1, 57.4, 94.3)
	$\text{RSSI}_{\text{SD},10}$	80.8	80.7 (78.3, 71.3, 92.6)	80.8 (72.7, 74.1, 95.6)	80.7 (75.4, 72.7, 94.1)
	RSSI, $\overline{\text{RSSI}}_{10}$, $\text{RSSI}_{\text{SD},10}$	**93.2**	**93.2** (96.2, 90.4, 93.0)	**93.2** (94.8, 89.0, 95.6)	**93.2** (95.5, 89.7, 94.3)
	ΔRSSI ("left" derivative)	60.3	63.5 (69.6, 65.7, 55.2)	60.3 (44.7, 37.4, 98.8)	57.6 (54.4, 47.7, 70.8)
	$\overline{\text{RSSI}}_{10}^{\{-4,-3,-2,-1,1,2,3,4\}}$	80.0	79.9 (93.0, 72.3, 74.4)	80.0 (92.8, 64.8, 82.3)	79.8 (92.9, 68.4, 78.1)



(a) Per class precision values for logistic regression.

(b) Per class recall values for logistic regression.

(c) Per class precision values for decision trees.

(d) Per class recall values for decision trees.

Fig. 2: Per class influence of the feature selection on fairness.

and recall on this class show that the model is able to find the largest part of good links with minimal confusion. On the other hand, on the minority *intermediate* class, the F1 is as low as 44% with a precision of 81% and recall of only 31%. Low recall means that only a fraction of the links classified as *intermediate* are indeed *intermediate*. Such model needs improvement to detect more *intermediate* links accurately for better and fairer recognition of this minority class.

Smoothing the $RSSI$ over 10 packets increases the performance to 89% and 91% respectively (line 2 in the table)

ISBN 978-3-903176-35-5 © IFIP

TABLE III: Comparison of various data resampling strategies using linear and non-linear ML algorithms.

Algorithm	Resampling	Acc. [%]	Precision [%]	Recall [%]	F1 [%]
Linear (Logistic)	None	96.8	89.2 (98.8, 69.9, 98.9)	84.3 (99.0, 55.2, 98.6)	86.0 (98.9, 61.7, 97.4)
	RUS	92.2	92.3 (97.1, 90.2, 89.6)	92.2 (93.9, 86.0, 96.7)	92.2 (95.5, 88.0, 93.0)
	ROS	92.2	92.3 (97.1, 90.2, 89.6)	92.2 (93.9, 86.0, 96.7)	92.2 (95.5, 88.0, 93.0)
Non-linear (DTree)	None	97.0	87.8 (98.9, 66.9, 97.5)	87.9 (98.6, 67.0, 98.1)	87.8 (98.7, 67.0, 97.8)
	RUS	93.1	93.1 (96.2, 90.2, 93.0)	93.1 (94.6, 89.0, 89.6)	93.1 (95.4, 89.6, 94.3)
	ROS	93.2	93.2 (96.2, 90.4, 93.0)	93.2 (94.8, 89.0, 95.6)	93.2 (95.5, 89.7, 94.3)

while generating certain synthetic features further improves the results by 2-3 percentage points. Concretely, the fourth line corresponding to each algorithm in the table shows that learning from the feature set of $RSSI$, $\overline{\text{RSSI}}_{10}$, $\text{RSSI}_{\text{SD},10}$ yields 92% and 93% accuracy, respectively. The high values of precision and recall for these feature combinations can also be visualized as in Figures 2a and 2b.

These results show that only using instant $RSSI$ as a feature with our imbalanced dataset is not sufficient to learn to discriminate the minority intermediate class sufficiently well. The F1 score for the *intermediate* class is only 44% for the linear model and 50% for the non-linear model trained with $RSSI$ only. Similarly, also the precision and recall results for the intermediate class are modest for $RSSI$ only. As visualized in Figures 2c and 2d, precision is 76% and recall is 38% for the *intermediate* class.

When the two models are trained with a combination of features, namely $RSSI$, $\overline{\text{RSSI}}_{10}$, $\text{RSSI}_{\text{SD},10}$, the performance of the *intermediate* class increases by more than 44%, resulting in a F1 score of 88% for the linear model and 89% for the non-linear model. This large increase in performance, leading to a fairer classification, also comes with slight increases of $1-2\%$ in the F1 scores of the majority classes. According to Figure 2a this feature combination results in a very good precision on all three classes for the linear model, namely 97% on *good* and 90% on *intermediate* and *bad* respectively. For the non-linear number, the values depicted in Figure 2c are all very high as well, namely 96% on *good* and 90% on *intermediate* and 93% on *bad* classes. It can be seen that the non-linear model is slightly more precise at determining *bad* links with a slight penalty for *good* links compared to the linear model. The recall values are also very high for both models. According to Figure 2b, the recall is 94% on *good* and 86% on *intermediate* and 97% on *bad* classes when the model is trained with the linear logistic regression, while Figure 2d presents that the recall is 95% on *good* and 89% on *intermediate* and 96% on *bad* classes when the model is trained with the non-linear decision tree. It can be seen from these results that the advantage of the DTree model comes from its ability to yield higher recall values showing that not too many true positive have been missed in classification. While some of the *intermediate* class links are still missed as there is about 10 percentage points difference compared to the other two classes (*bad* and *good*), $RSSI$, $\overline{\text{RSSI}}_{10}$, $\text{RSSI}_{\text{SD},10}$ feature set provides the highest fairness.

The feature analysis also shows that by smoothing the training data, therefore removing noise and transitory fluctuations and capturing the boundaries of the variations, the learner can improve its performance and become fairer on the intermediate class. It is observed that the transient fluctuations are more prominent on the intermediate class, which is conforming to the findings of the literature [18].

V. Compensating for the minority class in the training data to improve per class fairness

To compensate for the imbalanced class in the training data, and mitigate bias, the ML literature suggests employing resampling methods developed using statistical tools. These methods modify the distributions of the classes and re-balance the dataset. For our work, we consider two simple standard candidates; i) random oversampling (ROS), ii) random undersampling (RUS). The ROS [33] approach considers duplicating the trace-set entries of the minority classes for all class sizes to reach the size of the majority class. The resultant resampled dataset is larger than the original. Contrarily, the RUS [33] approach reduces all majority class sizes to the size of the minority class by randomly eliminating instances from other larger classes. Therefore, the obtained resampled dataset becomes smaller.

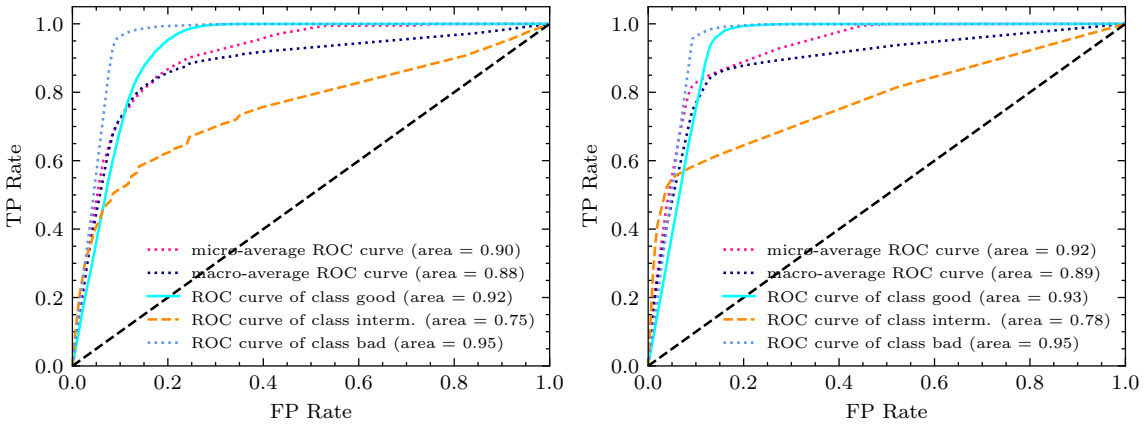
Table III presents the results of evaluation for the selected resampling strategies. For both classes of algorithms, Table III reveals that employing RUS and ROS resampling strategies degrades the accuracy by nearly 4%, albeit improves the precision, recall and F1 score up to about 8%. However, looking at the per-class break-downs in Table III, a more detailed insight can be acquired, where the performance discrimination on the majority classes decreases, expressively, the precision for the *good* class drops from 98% and 97% for the linear model and from 98% and 96% for the non-linear model, while the precision for the *bad* class drops from 98% to 89% for the linear model and from 97% to 93% for the non-linear model. However, the precision for the *intermediate* class increases by over 30 percentage points from 69% to 90% for the linear model and from 66% to 90% for the non-linear model. Similar conclusions can be drawn for the other metrics.

The analyses in this section demonstrate that when optimizing the overall performance of the classifier without considering per-class fairness, the best results are obtained on the actual dataset resulted in 97% accuracy. However, in this case the performance of recognizing the minority classes, namely

ISBN 978-3-903176-35-5 © IFIP

TABLE IV: The impact of linear and non-linear ML algorithms on the effectiveness of the ultimate LQE model.

Type	Algorithm	Acc. [%]	Precision [%]	Recall [%]	F1 [%]	Training Time [s]
Baseline	Majority classifier	33.3	11.1 (33.3, 0.0, 0.0)	33.3 (0.0, 0.0, 0.0)	16.7 (50.0, 0.0, 0.0)	0.6
Linear	Logistic regression	92.2	92.3 (97.1, 90.2, 89.6)	92.2 (93.9, 86.0, 96.7)	92.2 (95.5, 88.0, 93.0)	2.5
	SVM (linear kernel)	92.1	92.2 (97.4, 90.0, 89.2)	92.1 (93.7, 85.8, 96.8)	92.1 (95.5, 87.8, 92.8)	93.6
Non-linear	DTree	93.1	93.1 (96.2, 90.2, 93.0)	93.1 (94.6, 89.0, 95.6)	93.1 (95.4, 89.6, 94.3)	1
	MLP	93.4	93.4 (96.7, 90.5, 93.0)	93.4 (94.9, 89.5, 90.0)	93.4 (95.8, 90.0, 94.3)	93.4



(a) Multi-class ROC curve for logistic regression.

(b) Multi-class ROC curve for decision trees regression.

Fig. 3: Multi-class receiver operating characteristic (ROC) representations portraying the performance of the two classification models.

intermediate links is up to 67% achieved by the non-linear model. In cases where correctly discriminating all classes is a requirement, then resampling is the recommended approach as it increases the correct discrimination, i.e., fairness, of the *intermediate* links by over 20%.

VI. Performance Evaluation of the Model

We now compare the proposed DTree model with the other three ML models and a majority baseline, as summarized in Table IV. In general, non-linear ML-based LQE models performed slightly better than the linear counterparts within a tiny margin of about 1%. This confirms the relatively non-linear nature of the problem and also verifies previous findings where linear regression (linear algorithm) and neural networks (non-linear algorithm) performed similarly [10].

The tiny margin observed in Table IV is also confirmed in Figs. 3a and 3b, where the figures present receiver operating characteristic (ROC) curve and the area under the curve (AUC) values for each of the class, and their micro and macro average performances. Indeed, non-linear model is slightly better due to a higher AUC value compared to that of the linear counterparts, for all link classes. This tiny margin is mainly due to the fact that in Rutgers trace-set, nodes are relatively close and in line-of-sight, and thus measurements data highly likely follow normal distribution. Contrarily, in case of non-line-of-sight and mobility scenarios, the input data would no longer follow any known statistical distribution. This is where non-linear counterparts, especially non-parametric algorithms, would be advantageous. For intermediate links, non-linear models outperformed the linear counterparts with about 2 percentage points margin.

Considering computational complexity reflected in training time, as per the last column of Table IV, we clearly demonstrate that the proposed LQE model based on DTree outperformed other LQE models in terms of computational complexity and at the same time, the DTree model accomplished one of the best performances for both the general model and the intermediate link class. DTree takes only 1 minute to train as opposed to 2.5 minutes for the logistic regression and it achieves slightly better performance (1%). It takes 90 times less training time compared to MLP and the performance compromise is less than 1%.

VII. Summary and Future Work

In this paper, we proposed a new decision tree based LQE model so as to improve fairness on minority classes. We compare the proposed classifier against three other ML approaches on a selected imbalanced dataset using five different performance metrics. Our study reveals that using additional metrics, such as F1 score to complement the widely used accuracy can help identify suboptimal performance on imbalanced datasets. For LQE, this means that the models are unfair and tend to confuse the *intermediate* quality links with *bad* quality links. To this end, we demonstrated the impact of feature selection and resampling techniques on improving per-class classification. On the selected dataset, we showed

ISBN 978-3-903176-35-5 © IFIP

that the performance on the minority class can be increased by over 40% through feature selection and by over 20% through resampling, leading to increased fairness. We also showed that non-linear models seem to be more appropriate for the problem, however, their advantage over linear models is marginal. Finally, we demonstrated that once training time is also taken into account, the proposed decision tree based model outperforms all the other considered models.

As a future work, we plan to extend the considered ML models to multi-technology LQE estimation as well as to use the recently developed LIME [34] library for explainable deep learning to further investigate fairness aspects on such models.

ACKNOWLEDGMENT

This work was funded in part by the Slovenian Research Agency (Grant no. P2-0016 and J2-9232) and in part by the EC H2020 NRG-5 Project (Grant no. 762013).

REFERENCES

[1] G. T. Nguyen, R. H. Katz, B. Noble, and M. Satyanarayanan, "A trace-based approach for modeling wireless channel behavior," in Winter Simulation Conf., California, USA, 8-11 December 1996, pp. 597–604.

[2] H. Balakrishnan and R. H. Katz, "Explicit loss notification and wireless web performance," in IEEE Globecom Internet Mini-Conference, Sydney, Australia, November 1998, http://nms.lcs.mit.edu/~hari/papers/globecom98/.

[3] A. Woo, T. Tong, and D. Culler, "Taming the underlying challenges of reliable multihop routing in sensor networks," in Proceedings of the 1st international conference on Embedded networked sensor systems, California, USA, 5–7 November 2003, pp. 14–27.

[4] W. Sun, W. Lu, Q. Li, L. Chen, D. Mu, and X. Yuan, "WNN-LQE: Wavelet-Neural-Network-Based Link Quality Estimation for Smart Grid WSNs," IEEE Access, vol. 5, pp. 12 788–12 797, July 2017.

[5] S. Demetri, M. Zúñiga, G. P. Picco, F. Kuipers, L. Bruzzone, and T. Telkamp, "Automated estimation of link quality for LoRa: A remote sensing approach," in ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN'19), Montreal, CA, 15-18 April 2019.

[6] T. Liu and A. E. Cerpa, "Foresee (4C): Wireless link prediction using link features," in 10th Int. Conference on Information Processing in Sensor Networks (IPSN'10), Chicago, USA, 12-14 April 2011.

[7] E. Ancillotti, C. Vallati, R. Bruno, and E. Mingozzi, "A reinforcement learning-based link quality estimation strategy for RPL and its impact on topology management," Comp. Comms., vol. 112, pp. 1–13, Nov. 2017.

[8] H. Okamoto, T. Nishio, M. Morikura, K. Yamamoto, D. Murayama, and K. Nakahira, "Machine-learning-based throughput estimation using images for mmwave communications," in 2017 IEEE 85th Vehicular Technology Conference (VTC Spring). IEEE, 2017, pp. 1–6.

[9] M. L. Bote-Lorenzo, E. Gómez-Sánchez, C. Mediavilla-Pastor, and J. I. Asensio-Pérez, "Online machine learning algorithms to predict link quality in community wireless mesh networks," Computer Networks, vol. 132, pp. 68–80, February 2018.

[10] T. Liu and A. E. Cerpa, "Temporal adaptive link quality prediction with online learning," ACM Transactions on Sensor Networks (TOSN), vol. 10, no. 3, p. 46, 2014.

[11] S. Rekik, N. Baccour, M. Jmaiel, and K. Drira, "Low-power link quality estimation in smart grid environments," in International Wireless Communications and Mobile Computing Conference (IWCMC'15), Dubrovnik, Croatia, 24-28 August 2015.

[12] X. Luo, L. Liu, J. Shu, and M. Al-Kali, "Link quality estimation method for wireless sensor networks based on stacked autoencoder," IEEE Access, vol. 7, pp. 21 572–21 583, 2019.

[13] Z.-Q. Guo, Q. Wang, M.-H. Li, and J. He, "Fuzzy logic based multidimensional link quality estimation for multi-hop wireless sensor networks," IEEE Sensors Journal, vol. 13, no. 10, pp. 3605–3615, October 2013.

[14] W. Rehan, S. Fischer, and M. Rehan, "Machine-learning based channel quality and stability estimation for stream-based multichannel wireless sensor networks," MDPI Sensors, vol. 16, no. 9, p. 1476, Sept. 2016.

[15] J. Shu, S. Liu, L. Liu, L. Zhan, and G. Hu, "Research on link quality estimation mechanism for wireless sensor networks based on support vector machine," Chinese Journal of Electronics, vol. 26, no. 2, pp. 377–384, April 2017.

[16] H.-J. Audéoud and M. Heusse, "Quick and efficient link quality estimation in wireless sensors networks," in Wireless On-demand Network Systems and Services (WONS), 2018 14th Annual Conference on. IEEE, 2018, pp. 87–90.

[17] C. A. Boano, M. Zuniga, T. Voigt, A. Willig, and K. Römer, "The triangle metric: Fast link quality estimation for mobile wireless sensor networks," in International Conference on Computer Communication Networks, Zurich, Switzerland, 2-5 August 2010.

[18] N. Baccour, A. Koubâa, L. Mottola, M. A. Zúñiga, H. Youssef, C. A. Boano, and M. Alves, "Radio link quality estimation in wireless sensor networks: A survey," ACM Transactions on Sensor Networks (TOSN), vol. 8, no. 4, p. 34, September 2012.

[19] T. Zhang, t. zhu, J. Li, M. Han, W. Zhou, and P. Yu, "Fairness in semi-supervised learning: Unlabeled data help to reduce discrimination," IEEE Transactions on Knowledge and Data Engineering, pp. 1–1, 2020.

[20] A. Chouldechova, "Fair prediction with disparate impact: A study of bias in recidivism prediction instruments," Big Data, vol. 5, no. 2, pp. 153–163, June 2017.

[21] Z. Obermeyer, B. Powers, C. Vogeli, and S. Mullainathan, "Dissecting racial bias in an algorithm used to manage the health of populations," Science, vol. 366, no. 6464, pp. 447–453, 2019. [Online]. Available: https://science.sciencemag.org/content/366/6464/447

[22] L. A. Jeni, J. F. Cohn, and F. De La Torre, "Facing imbalanced data–recommendations for the use of performance metrics," in 2013 Humaine association conference on affective computing and intelligent interaction. IEEE, 2013, pp. 245–251.

[23] G. Cerar, H. Yetgin, M. Mohorˇ cič, and C. Fortuna, "On Designing a Machine Learning Based Wireless Link Quality Classifier," in 31st International Symposium on Personal, Indoor and Mobile Radio Communications, Virtual Conference, 31 August-3 September 2020.

[24] C. M. Bishop, Pattern Recognition and Machine Learning (Information Science and Statistics). Berlin, Heidelberg: Springer-Verlag, 2006.

[25] D. Aguayo, J. Bicket, S. Biswas, G. Judd, and R. Morris, "Link-level measurements from an 802.11 b mesh network," in ACM SIGCOMM Computer Communication Review, vol. 34, no. 4. ACM, 2004, pp. 121–132.

[26] S. K. Kaul, M. Gruteser, and I. Seskar, "Creating wireless multi-hop topologies on space-constrained indoor testbeds through noise injection," in TRIDENTCOM, Barcelona, Spain, 1-3 March 2006.

[27] S. Fu, Y. Zhang, Y. Jiang, C. Hu, C.-Y. Shih, and P. J. Marrón, "Experimental study for multi-layer parameter configuration of WSN links," in 2015 IEEE 35th International Conference on Distributed Computing Systems (ICDCS). IEEE, 2015, pp. 369–378.

[28] Y. Chen, A. Wiesel, and A. O. Hero, "Robust shrinkage estimation of high-dimensional covariance matrices," IEEE Transactions on Signal Processing, vol. 59, no. 9, pp. 4097–4107, 2011.

[29] T. Van Haute, E. De Poorter, F. Lemic, V. Handziski, N. Wirström, T. Voigt, A. Wolisz, and I. Moerman, "Platform for benchmarking of RF-based indoor localization solutions," IEEE Communications Magazine, vol. 53, no. 9, pp. 126–133, 2015.

[30] E. Anderson, G. Yee, C. Phillips, D. Sicker, and D. Grunwald, "The impact of directional antenna models on simulation accuracy," in Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks, 2009. WiOPT 2009. 7th International Symposium on. IEEE, 2009, pp. 1–7.

[31] S. Arlot, A. Celisse et al., "A survey of cross-validation procedures for model selection," Statistics surveys, vol. 4, pp. 40–79, March 2010.

[32] A. A. Freitas, "Understanding the crucial role of attribute interaction in data mining," AI Review, vol. 16, no. 3, pp. 177–199, Nov. 2001.

[33] N. V. Chawla, N. Japkowicz, and A. Kotcz, "Special issue on learning from imbalanced data sets," ACM SigKDD Explorations Newsletter, vol. 6, no. 1, pp. 1–6, June 2004.

[34] M. T. Ribeiro, S. Singh, and C. Guestrin, ""why should I trust you?": Explaining the predictions of any classifier," in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016, 2016, pp. 1135–1144.