CLASSIFICATION OF WIRELESS LINKS USING MACHINE LEARNING TECHNIQUES

Gregor Cerar

Doctoral Dissertation
Jožef Stefan International Postgraduate School
Ljubljana, Slovenia

Supervisor: Prof. Dr. Mihael Mohorčič, Jožef Stefan Institute, Ljubljana, Slovenia
Co-Supervisor: Dr. Carolina Fortuna, Jožef Stefan Institute, Ljubljana, Slovenia

Evaluation Board:
Asst. Prof. Dr. Andrej Hrovat, Chair, Jožef Stefan Institute, Ljubljana, Slovenia
Asst. Prof. Dr. Marko Meža, Member, Faculty of Electrical Engineering, University of Ljubljana, Ljubljana, Slovenia
Asst. Prof. Dr. Halil Yetgin, Member, Bitlis Eren University, Turkey

MEDNARODNA PODIPLOMSKA ŠOLA JOŽEFA STEFANA
JOŽEF STEFAN INTERNATIONAL POSTGRADUATE SCHOOL

Gregor Cerar

CLASSIFICATION OF WIRELESS LINKS USING MACHINE LEARNING TECHNIQUES

Doctoral Dissertation

KLASIFIKACIJA BREZŽIČNIH POVEZAV S POMOČJO STROJNEGA UČENJA

Doktorska disertacija

Supervisor: Prof. Dr. Mihael Mohorčič

Co-Supervisor: Dr. Carolina Fortuna

Ljubljana, Slovenia, April 2021

Acknowledgments

First and foremost, I would like to thank my two supervisors, Prof. Dr. Mihael Mohorčič and Dr. Carolina Fortuna, for their patience, support and guidance during my doctoral studies over the past 5 years. With them, I have sharpened my writing skills and time management. Thank you, Dr. Fortuna, for helping me shift my studies to an exciting area of machine learning.

Then, I would like to thank my coworkers at the Department of Communication Systems at the Jožef Stefan Institute for their insightful discussions that inspired and motivated me not to give up on pursuing my PhD.

Also, I would like to thank Dr. Halil Yetgin for helping me with my biggest career achievement so far, a publication accepted in a top journal in our field. Thank you for the wonderful year and a half we spent together.

I would also like to thank the members of the evaluation board for providing very constructive reviews that helped further improving the thesis.

I would also like to thank the Slovenian Research Agency (ARRS) for the Young Researcher Grant that funded my research.

Last but not least, I would like to thank my wife Mira and my children Neva, Jurij, Štefan and Pavlina for putting up with my absence and long working hours. Thank you!

Abstract

Due to the nature of the wireless transmission medium, wireless communications are characterised by notably larger losses of data packets than wired communications. The quality of wireless links is highly dependent on channel variations, interference and even transceiver imperfections. Such link uncertainty instigated the development of numerous techniques that can withstand uncertain conditions by adjusting the parameters of the wireless link to achieve higher reliability or selecting a more reliable alternative wireless link for data transmission. These techniques rely on effective wireless link quality estimation.

Analytical approach, initially used for link quality estimation, was soon complemented and superseded by statistical and more recently machine learning approaches based on empirical data traces, resulting in data-driven models. Statistical approach fits the model to the underlying distribution of the specific property or behaviour under investigation, whereas machine learning models transfer link quality estimation into classification problem, potentially taking into account multiple phenomena of the link, thus being better suited for the real-world wireless links that exhibit dynamic behaviour and are often subject to various transient phenomena. Observation of wireless link quality is important also from the perspective of early anomaly detection, especially in large scale industrial or commercial deployments. Automatic detection of malfunctions, caused by software, hardware, or external factors in dynamic operating environment, can be an important asset to reduce unexpected maintenance downtime, and consequently financial losses.

In this dissertation we are concerned with classification of wireless links using machine learning techniques to support link quality estimation and anomaly detection. Our main attention is given to the challenges of designing wireless link classifiers based on machine learning techniques. In the first part, focused on link quality estimation, we perform in-depth quantitative research on how each step of feature engineering, data engineering and algorithm tuning influences the estimation performance. We pay special attention to improving the detection of minority classes, i.e. less frequent data samples in the wireless link dataset. We propose a new supervised classifier for link quality estimation that improves detection of minority class by over 40% through feature selection, and by over 20% through data re-sampling strategies, without any significant impact on detecting majority classes.

The second part of the dissertation is focused on wireless link anomaly detection, where we define four basic types of anomalies that occur on wireless links and describe their symptoms and probable causes. With a systematic quantitative approach, we investigate the performance of two threshold-based approaches, three supervised and three unsupervised reference machine learning algorithms. We show that the performance of supervised approaches may be dominant, however, certain unsupervised approaches combined with deep learning autoencoders for input features come close to the performance of supervised approaches while not requiring annotated data, which may prove as a significant advantage.

Povzetek

Zaradi narave brezžičnega prenosnega medija je za brezžične komunikacije značilna večja verjetnost izgube podatkovnih paketov kot pri žičnih komunikacijah. Kakovost brezžičnih povezav je močno odvisna od spreminjajočih razmer na kanalu, medsebojnih motenj in tudi pomanjkljivosti primopredajnika. Nezanesljivost brezžičnih povezav je spodbudila razvoj številnih tehnik, ki lahko kljubujejo takšnim pogojem s prilagajanjem parametrov brezžične povezave za doseganje večje zanesljivosti ali z izbiro nadomestne brezžične povezave za prenos podatkov. Skupno tem tehnikam je, da potrebujejo učinkovito oceno kakovosti brezžične povezave.

Analitični pristop, ki se je sprva uporabljal za ocenjevanje kakovosti povezav, sta kmalu dopolnila in nadomestila statistični, v zadnjem času pa vse bolj tudi pristop s strojnim učenjem. Slednja sta zasnovana na empiričnih meritvah, zato govorimo o podatkovno zasnovanih modelih. Statistični pristop temelji na ujemanju modela s statistično porazdelitvijo določene lastnosti ali vedenja. Modeli strojnega učenja prevedejo ocenjevanje kakovosti brezžične povezave na problem klasifikacije. Takšni modeli lahko upoštevajo vrsto pojavov na povezavah, zaradi česar so primerni za realne brezžične povezave, ki izkazujejo dinamično vedenje in so pogosto podvržene različnim prehodnim pojavom. Spremljanje kakovosti brezžične povezave je pomembno tudi z vidika zgodnjega odkrivanja anomalij, zlasti v obsežnih industrijskih ali komercialnih sistemih. Samodejno odkrivanje napak, ki jih povzročijo programska ali strojna oprema ter okoljski dejavniki, lahko pomembno prispeva k zmanjšanju neželenih izpadov delovanja in posledično finančnih izgub.

V tej disertaciji se ukvarjamo z razvrščanjem brezžičnih povezav z uporabo tehnik strojnega učenja v podporo učinkovitemu ocenjevanju kakovosti povezav in odkrivanju anomalij. Glavni poudarek je na izzivih načrtovanja klasifikatorjev brezžičnih povezav, ki temeljijo na tehnikah strojnega učenja. V prvem delu disertacije se osredotočamo na učinkovito ocenjevanje kakovosti brezžične povezave. Izvedli smo poglobljeno kvantitativno študijo o vplivih posameznih korakov inženiringa značilk in podatkov ter finega nastavljanja algoritmov na uspešnost ocenjevanja brezžičnih povezav. Posebno pozornost namenjamo izboljšanju zaznavanja manjšinskih razredov, tj. manj pogostim vzorcem podatkov iz nabora meritev na brezžičnih povezavah. Za oceno kakovosti povezav predlagamo tudi nov nadzorovani razvrščevalnik, ki s pomočjo optimalne izbire značilk izboljša zaznavanje manjšinskega razreda za več kot 40 %, z uporabo strategij ponovnega vzorčenja podatkov pa za več kot 20 %, ne da bi se s tem poslabšalo zaznavanje ostalih razredov.

Drugi del disertacije se osredotoča na odkrivanje anomalij v brezžičnih povezavah. Tu opredelimo štiri osnovne vrste anomalij, ki se pojavljajo na brezžičnih povezavah, opišemo njihove značilnosti in verjetne vzroke. S sistematičnim kvantitativnim pristopom raziskujemo delovanje dveh pragovnih pristopov ter po treh nadzorovanih in nenadzorovanih referenčnih algoritmov strojnega učenja. Pokažemo, da so nadzorovani algoritmi uspešnejši pri odkrivanju anomalij, da pa se jim nekateri nenadzorovani pristopi v povezavi s samokodirnikom s področja globokega učenja zelo približajo, pri čemer ne potrebujejo zahtevnega in časovno potratnega označevanja podatkov, kar lahko predstavlja pomembno prednost.

Contents

Abbreviations

4B	... Four-Bit
4C	... Foresee
AI	... Artificial Intelligence
BER	... Bit Error Rate
CDF	... Cumulative Distribution Function
DNN	... Deep Neural Network
ETX	... Expected Transmission count
F-LQE	... Fuzzy-logic based LQE
FFT	... Fast Fourier Transformation
FLI	... Fuzzy-logic Link Indicator
IForest	... Isolation Forest
IPS	... International Postgraduate School
InstaD	... Instantaneous Degradation
IoT	... Internet of Things
JSI	... Jožef Stefan Institute
KDD	... Knowledge Discovery and Data mining
KDP	... Knowledge Discovery Process
LOF	... Local Outlier Factor
LQE	... Link Quality Estimation
LQI	... Link Quality Indicator
LR	... Logistic Regression
ML	... Machine Learning
MLP	... Multi-Layer Perceptron
MSE	... Mean Squared Error
NN	... Neural Network
OC-SVM	... One-Class Support Vector Machine
PRR	... Packet Reception Ratio
PSR	... Packet Success Ratio
RBF	... Radial Basis Function
RForest	... Random Forest
RNP	... Required Number of Packets
ROS	... Random Over-Sample
RSS	... Received Signal Strength
RSSI	... Received Signal Strength Indicator
RUS	... Random Under-Sample
SGD	... Stochastic Gradient Descent
SNR	... Signal-to-Noise Ratio
SVM	... Support Vector Machine
SlowD	... Slow Degradation
SuddenD	... Sudden Degradation without recovery
SuddenR	... Sudden degradation with Recovery

TCP	...	Transmission Control Protocol
WMEWMA	...	Window Mean with an Exponentially Weighted Moving Average
WNN-LQE	...	Wavelet Neural Network based LQE
WSN	...	Wireless Sensor Network
kNN	...	k-Nearest Neighbour

Chapter 1

Introduction

With the advent and proliferation of wireless technologies in the early 1990s, it soon became clear that data transmission was inferior to that in wired networks. It was observed that the transmission of data packets over wireless links was more prone to excessive packet loss than over wired links. As illustrated in Figure 1.1, a wireless link is a connection between two communicating wireless nodes established with the aim to exchange information. The transmitter encodes information into the signal. Then, the signal is emitted through a medium (e.g., water, air) to the destined receiver node. In the receiver, the signal is decoded, and the original information is recovered. Along this transmission path we can experience information loss due to signal propagation conditions that can vary significantly due to fading, path loss, shadowing, Doppler effects and interferences, as well as transceiver hardware imperfections and software issues. These inherent dynamic conditions led to the development and continuous improvement of numerous advanced wireless communication techniques aimed at improving link performance between the transmitter and the receiver. These include a variety of modulation and coding schemes, medium access methods, error detection and correction techniques, antenna arrangements, radio spectrum management, high frequency communications, and others. These techniques, however, rely on wireless link state information so that the link parameters can be adapted, an alternative and more reliable link can be selected for wireless data transmission, and any malfunctions of a given link can be recognized.

Initial investigation of new wireless communication techniques often relies on more or less abstracted analytical or statistical link models, mimicking the particular behaviour under investigation. However, real world phenomena can only be captured by so-called data-driven models based on empirical data traces from representative networks. To date, many analytical and data-driven link quality models have been proposed to investigate and develop new techniques for minimizing losses, improving the performance, and reliability of wireless communications. Analytical models can elegantly describe and abstract important effects that occur on wireless links, but they cannot consider and include all influencing factors and phenomena. For this reason, statistical and more recently machine learning (ML) approaches resulting in data-driven models became the predominant tool in link quality research.

To date, many analytical and data-driven link quality models have been proposed to investigate and develop new techniques for minimizing losses, improving the performance, and reliability of wireless communications. Analytical models can elegantly describe and abstract important effects that occur on wireless links, but they cannot consider and include all influencing factors and phenomena. For this reason, statistical and more recently machine learning (ML) approaches resulting in data-driven models became the predominant tool in link quality research.

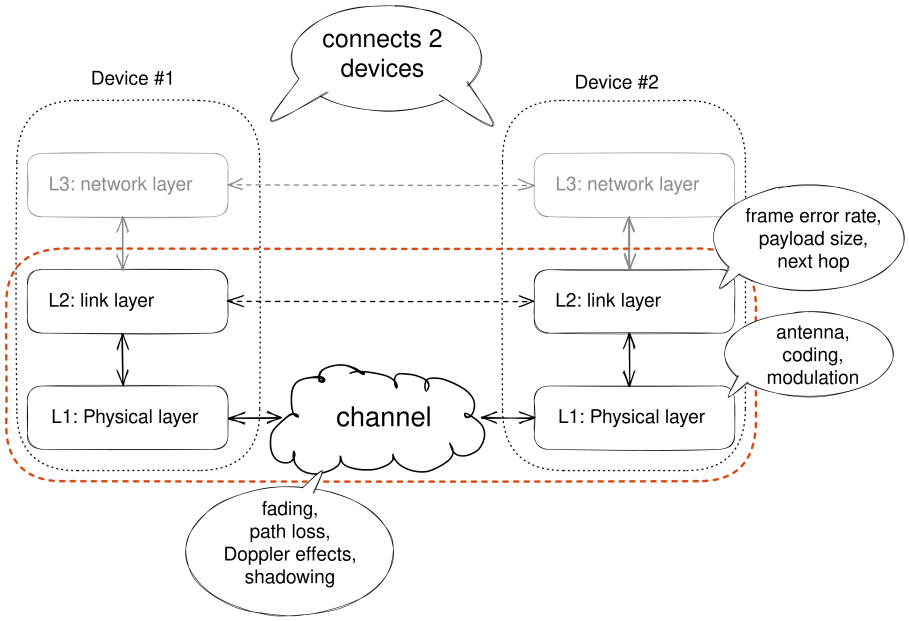


Figure 1.1: A unified model of a wireless link

Essentially, as depicted in Figure 1.2, there are two different approaches to obtain wireless link state information, a reactive approach (Fig. 1.2a) and a proactive approach (Fig. 1.2b). A reactive approach assesses wireless link's current state or recent past events, while a proactive gains experience from observations and attempts to estimate the future state of the wireless link.

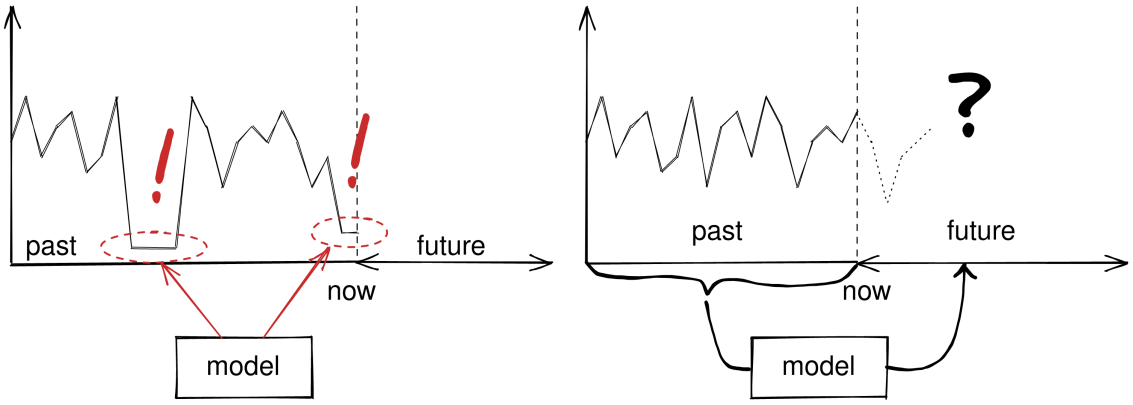


Figure 1.2: Reactive (a) and proactive (b) approach for obtaining wireless link state information

Wireless link state information can be expressed either as a continuous or a discrete value. Continuous values can be interpreted as a score or as a way of expressing a particular metric value, depending on the application. Discrete values can be viewed as a class, label, category, group, or binary response, and their meaning is tied to the application. Some examples of labels are "very bad", "bad", "intermediate", "good", "very good", "anomalous", and "degraded". In the thesis, we are only concerned with discrete values, i.e., binary and multi-class categorization.

From the literature, the application of link state information can be broadly categorized to (1) link quality assessment, (2) link quality estimation, (3) anomaly detection, and (4) anomaly estimation. As shown in the upper part of Figure 3, link quality assessment and link quality estimation target network operation applications. Their objective is to optimize data transmission in terms of throughput, latency, or reliability by evaluating (all) wireless links. In the lower part of Figure 1.3, anomaly detection and anomaly estimation target network maintenance applications. Their objective is to keep the network operational and reliable by evaluating the overall state of the network and detecting or even predicting malfunctions based on link states, making network monitoring and maintenance more efficient. Furthermore, the left-hand side of Figure 1.3 implicates that link quality assessment and anomaly detection are used for a reactive approach, while link quality estimation and anomaly estimation on the right-hand side are used for a proactive approach. In the thesis, two out of four link state information applications are studied in depth: (i) link quality estimation (indicated in top right quadrant), which predicts the future state of the wireless link, and (ii) anomaly detection (indicated in bottom left quadrant), which detects and classifies anomalies that occur on wireless links.

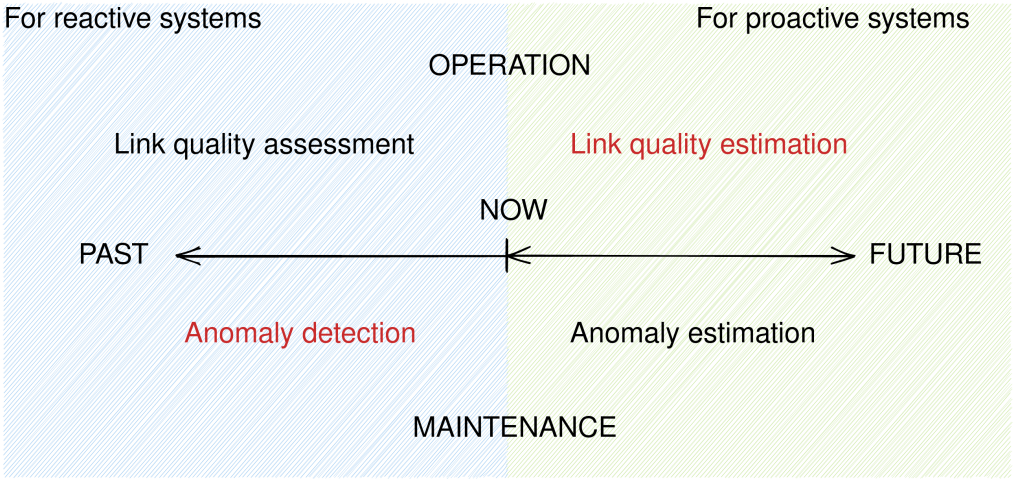


Figure 1.3: Link state information applications.

We started our research in link quality estimation by providing an in-depth survey and a comprehensive analysis of existing data-driven wireless link classifiers, and continued by their comprehensive and quantitative analysis by extracting the approximate performance per class from the reported results. These initial activities resulted in the design guidelines for the machine learning pipeline and data collection.

The survey showed that the earliest publication on data-driven link quality estimation dates back to mid 90's, where link quality was first modeled using a state machine, but data-driven approaches gained broad popularity only after 2000, while machine learning approaches began to trend after 2010. Deeper analysis of the publications reveals two design purposes of link quality estimators. The first design purpose is to improve the existing protocol. The second design purpose is to improve the performance of the existing link quality estimator. Each estimator in the literature emphasizes at least one of the following aspects of application quality: (i) reliability, (ii) adaptivity, (iii) stability, (iv) probing overhead, and (v) computational cost.

Quantitative analysis of the approximate performance per class from the reported results indicates that data-driven approaches are able to perform well in estimating link quality. Machine learning approaches are able to achieve comparable performance or out-

perform traditional techniques. Newer approaches based on machine learning also have great potential and are very innovative, e.g., automatically incorporating satellite imagery and depth perception sensors into link estimators for long- and short-range links, respectively.

However, analysis of link quality estimation research also reveals that descriptions of collected data sets, experiment configurations, data preprocessing steps, and model design decisions are often incomplete, making replication and comparison between different approaches difficult if not impossible. Thus, perhaps the most important outcome of the comprehensive analysis are the design guidelines for data preprocessing and data collection for machine learning based approaches. Their intention is to promote a more systematic description of steps and design decisions and enable an in-depth investigation of the impact of each step of the data preprocessing pipeline and design decisions on the final performance, following the Knowledge Discovery Process methodology.

As shown in Figure 1.4, the ML preprocessing pipeline consists of a series of sequential tasks that must be performed in exact order on the raw data. The raw data that passes through the preprocessing pipeline is transformed and (optionally) reorganized into a form suitable for the ML algorithm. The pipeline makes the process repeatable, reproducible, and easy to update as the raw data changes. The raw data traces in the form of time series first pass through the data preparation stage. The raw data traces are cleaned of invalid values and placeholders for missing samples are prepared. The interpolation step takes care of filling in the blanks. At this point, the data traces have the correct format and all entries are valid. In the feature engineering stage, new candidate features can be created by considering the context of the data, such as time dependence. The feature selection step selects most relevant features for further consideration in the process, while others are discarded. Since dealing with data in the form of time series, the size of the observation window and the size of the prediction window need to be determined in the window selection step. In the last stage of the preprocessing pipeline, the number of samples of each class is balanced to improve the fairness of the ML algorithm.

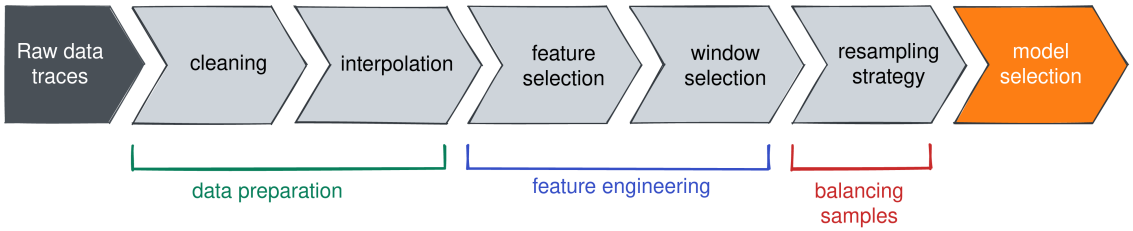


Figure 1.4: Machine learning preprocessing pipeline used for LQE investigation

A thorough investigation of the influence of the ML pipeline steps indicated that the largest improvement to the overall performance comes from the cleaning and interpolation steps, with significant improvements across all target classes and no known side effects. These steps, however, can be seen as extended data collection steps and should be in common to all algorithms under consideration. Thus, from the perspective of designing new classifiers for link quality estimation or anomaly detection, the most interesting stages are feature engineering and balancing of samples, which are receiving main attention in this dissertation. In particular, following the proposed guidelines we have developed a new tree-based link quality estimator that features low training time and excellent classification performance, and places special emphasis on classification fairness for minority classes. By tuning the pipeline parameters, we achieved a higher degree of classification fairness and demonstrated that the dataset resampling contributes most to classification fairness for

minority classes, but comes with the side effect of a small performance penalty for better represented classes.

As a complement to link quality estimation and its importance for improving network operation, discussed above, we also investigated anomaly detection and its relevance for reactive network maintenance. As we show in Figure 5, anomalies in wireless communication networks can be caused by different underlaying phenomena and may exhibit differently with respect to the time scale and symptoms (e.g., received signal strength, delay, jitter) from an application perspective. Different anomalies that can occur on a wireless link can be abstracted into four basic types of anomalies, i.e. sudden degradation (SuddenD), sudden recovery (SuddenR), instantaneous degradation (InstaD), and slow degradation (SlowD). The first three types have a common sudden drop and a distinctly different recovery time, while SlowD has a steady rather than a sudden drop.

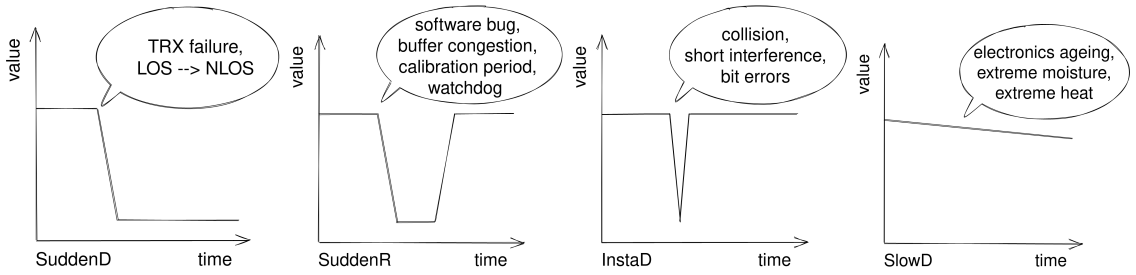


Figure 1.5: Representation of types of anomalies on wireless links

In the case of SuddenD, services become unavailable, offline, and unreachable, with possible causes being transceiver failures, a sudden change from line-of-sight (LoS) to non line-of-sight (NLoS), or obstacles with electromagnetic shielding. Symptoms of SuddenR are that services become sluggish and unreachable for a period of time. Possible causes are a buffer overflow, an aperiodic calibration period of the transceiver and a reset due to a software error where the watchdog had to intervene. In the case of an InstaD anomaly, a real-time service may experience immediate delays while other non-real-time services may operate unaffected. This type of degradation can be caused by an instantaneous fault, collision, quantization errors, errors in reading values, or sudden saturations in the transceiver's electronic components. Finally, a SlowD anomaly may go unnoticed for a very long time, so users may not even notice a difference in quality of service immediately. When relevant thresholds are triggered, users start to experience a degradation in quality of service. After the compensation methods employed are exhausted (e.g., buffers, queues, bandwidth preservation strategies), communications may be interrupted and intended services may become unavailable. Possible causes are ageing of electronics and extreme conditions, such as high humidity and extreme temperatures.

Since anomalies cause different effects on communication systems, it is important to reliably detect and classify them from the perspective of potentially applicable mitigation or remediation strategies. For our investigations of anomaly classifiers using machine learning, we adapted the data preprocessing pipeline as shown in Figure 1.6 and focused on the feature engineering stage. In addition to raw time series some other interesting feature representations include aggregated representation, histogram representation and frequency-domain representation.

An important aspect in investigating data-driven approaches is the availability of annotation for the used dataset which drives the selection of supervised or unsupervised machine learning algorithms. The latter are more desirable option for practical applica-

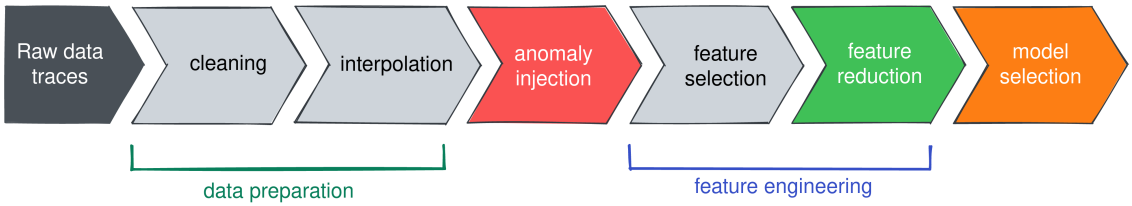


Figure 1.6: Preprocessing pipeline for machine learning

tions, because they do not require an annotated dataset, which can be time-consuming and expensive to obtain, yet the trade off is lower achieved accuracy. This can potentially be compensated by some further feature engineering. An interesting candidate in this respect is for instance a dimensionality reduction using some deep learning methods such as deep learning autoencoder, which is a generalized principal component analysis method, to perform automatic feature space reduction.

1.1 Motivation and Hypothesis

This dissertation investigates the use of data-driven approaches for wireless link classification, focusing on two particular aspects of link quality observation that are important for better decision making in wireless networks: link quality estimation and link anomaly detection. For link quality estimation, we investigated different ML approaches for predicting the future state of wireless links, with particular attention to the design process used and the impact of design decisions on the overall performance of link quality estimation. For link anomaly detection in wireless networks, we investigated different ML approaches using manually engineered features with particular attention on the possibility of performing automatic feature engineering with autoencoders based on deep neural networks.

The dissertation contributes to new insights and a deeper understanding of the suitability of different ML-based approaches for wireless link observation and classification and their use for wireless network management.

In the dissertation, we investigated the following hypotheses:

H1 Wireless link quality can be efficiently and accurately estimated using machine learning approaches.

H2 Wireless link anomalies can be effectively and reliably detected using machine learning approaches which can outperform rule-based approaches.

H3 Data pre-processing and algorithm parametrization have significant impact on the performance of wireless link quality estimation and wireless link anomaly detection.

H4 Wireless link anomaly detection based on traditional machine learning approaches can be further improved by using deep learning neural networks in the pre-processing step.

1.2 Methodology

The methodology used in this dissertation combined quantitative study and in-depth analysis of the literature and existing approaches with experimental measurements and collection of data traces in a wireless testbed, practical implementation of the state of the art and new ML-based models for LQE, and their computationally intensive performance evaluation.

We began by studying existing data-driven modeling of link quality metrics and link quality estimators, examining used methods, their input features, used datasets, and evaluation approaches. After implementing a classifier with a set of reference algorithms, we investigated the influence of pre-processing steps on the LQE output. The evaluation was performed in a quantitative manner observing the performance of the model output separately for each individual pre-processing step. Within quantitative comparison, we used multiple evaluation metrics, such as precision, recall and F1 score for the purpose of a benchmark. The experimental setup used a suitable freely available dataset and open-source machine learning library scikit-learn, while custom developed pre-processing functions and steps were made openly accessible on GitHub.

For the anomaly detection in wireless links we started with the investigation of existing link anomaly studies and formalization of different types of anomalies. Then, we prepared a clean dataset containing time-series observation of different wireless links and we injected formalized types of anomalies, thus creating a controlled environment. For the implementation of the classifiers, we were using the scikit-learn library and TensorFlow framework for neural networks. In the evaluation process, we performed quantitative examination of performance differences between supervised and unsupervised algorithms. In addition, we examined the impact of different time-series representation, along with newly proposed encoded representation.

1.3 Contributions

The scientific contributions of the dissertation are summarized as follows:

C1 In-depth survey and comprehensive analysis of existing data-driven approaches for wireless link quality estimation and wireless link anomaly detection resulting in design guidelines for approaches based on machine learning and for data collection (Chapter 2)

C2 Design of a generic machine learning pipeline with systematic investigation of the impact of data representation, feature space and pre-processing on data-driven approaches for wireless link quality estimation and wireless link anomaly detection. (Chapters 3, 4, 5, 6)

C3 A novel supervised link quality estimation classifier based on cross-layer data obtained from a representative real-world wireless network and its performance evaluation using standard methodology and metrics. (Chapter 4)

C4 Novel supervised and unsupervised anomaly detection classifiers based on cross-layer data obtained from real-world wireless network and their performance evaluation using standard methodology and metrics. (Chapters 5, 6)

C5 Performance enhancement of anomaly detection classifiers using autoencoder based on unsupervised deep learning neural networks for encoding input features in the pre-processing step. (Chapters 5, 6)

These main contributions have been published or submitted for publication in journals and at conferences along with a number of further minor contributions. The details related to the publications are listed in the Bibliography section at the end of the thesis.

1.4 Organization of the Dissertation

The dissertation is logically structured into 7 main chapters, starting and ending with Introduction and Conclusions and Future Work, respectively, while the main body consists of two parts. As mentioned in the introduction, two out of four applications for link state information are studied in detail, each being placed in a separate part of the thesis. The first part (Chapters 2, 3, 4) focuses on wireless link quality estimation, and the second part (Chapters 5, 6) focuses on anomaly detection in wireless links.

Chapter 1 provides a brief general introduction to wireless links and motivation for the research in wireless link quality estimation and anomaly detection. Following the high-level taxonomy it narrows the scope of dissertation to data-driven ML-based LQE approaches with particular attention given to data collection and preparation. The general introduction is followed by brief description of the motivation for this particular research topic and a list of hypotheses, an outline of the research methodology used, declaration of original contributions to science, and finally by the organization of the dissertation.

Part I of the main body is dedicated to the first link state information application, namely wireless link quality estimation. Chapter 2 provides a detailed and comprehensive overview of existing data-driven approaches, their applications, design process, freely available datasets suitable for link quality estimation, and design guidelines for LQE development. Chapter 3 presents a novel tree-based LQE model with high accuracy and low training time. Also, it examines various pre-processing steps and their impact on the model performance.

Data-driven approaches depend on suitability and representativeness of recorded datasets, but in reality these are rarely balanced and may exhibit different biases, further emphasizing the importance of data pre-processing steps. To this end, Chapter 4 explores how classification fairness of minority data classes in imbalanced dataset can be improved with minimal impact on majority data classes.

Part II of the main body is dedicated to the second link state information application, namely anomaly detection in wireless links. Chapter 5 defines four distinctive types of anomalies and their possible causes. It compares the performance of six reference ML algorithms, along with threshold approaches, on four different data representations. It also introduces encoded features using deep learning autoencoders, which show promising results. Chapter 6 further extends this work by fine-tuning ML algorithms and performing computationally intensive model selection with the aim to find the best ML pipeline configuration for anomaly detection.

Finally, Chapter 7 concludes the dissertation, summarizes the main contributions to science and outlines some possible directions for future work.

Part I

Wireless Link Quality Estimation

Chapter 2

Machine Learning for Wireless Link Quality Estimation

Data-driven LQE approaches rely on actual measured data that capture real-world phenomena. In traditional statistical approaches, this data is used to fit a model that best approximates the underlying distribution of typically one selected phenomenon, thus abstracting overly complex behaviour. With the recently increased use of ML-based approaches, leveraging large amounts of empirical data traces collected across various wireless networks, technologies and operating scenarios, data-driven LQE models are becoming increasingly sophisticated and accurate, enabling a more generic and high-level understanding of links' behaviour. However, ML techniques do not support as intuitive understanding of the relationship between the empirical data and resulting model as statistical distributions, and are relying on a complex development process in which each step has multiple design choices that can notably affect the overall performance of the model and need to be carefully considered to meet the requirements of served applications.

In this respect, this chapter provides a comprehensive study of wireless link quality estimators (LQEs) developed from empirical data, emphasising ML-based approaches, and the lessons learned serve as a foundation for the remaining chapters in the thesis. We start the investigation along seven different aspects, namely the underlying technologies and standards, the purpose of the estimator, the input metrics, the models used, the output value, the evaluation process, and finally their reproducibility.

From the perspective of served applications it is primarily important how ML-based LQE models serve their respective quality requirements, in particular reliability, adaptivity, responsiveness, stability, computational cost, and probing overhead. We show that approaches can be divided into two broad groups. The purpose of the first group is to improve the performance of a protocol or process, whereas the purpose of the second group is to propose a new or improved LQE.

Another important aspect for better understanding of the existing ML-based LQE models is how they follow the standard design process used in the machine learning community, known as Knowledge Discovery Process (KDP), consisting of data pre-processing, model building and model evaluation stages. We investigate and quantify the design decisions regarding cleaning and interpolation of data, feature selection, re-sampling strategy and ML model selection. We also provide insights into their impact on the overall LQE performance in terms of standard metrics such as accuracy, F1 score, precision and recall. Proving significant impact of data pre-processing on the overall performance of ML-based LQE model, we explore the steps of this stage more in depth in Chapters 3 and 4.

A particularly important aspect for building as well as for evaluating and comparing ML-based LQE models is the availability of representative datasets. To this end, we

evaluate selected open datasets suitable for data-driven LQE research, where we examine their origin, underlying technology, type of communication, available recorded metrics, provided metadata regarding the data collection process, and the total size of datasets.

This chapter concludes with elaborated lessons learned and generic design guidelines for developing ML-based LQE models considering application quality aspects as well as for designing a collection of generic trace-sets.

From the hypotheses outlined in Chapter 1.1, this chapter addresses and partially confirms hypothesis **H1**:

H1 Wireless link quality can be efficiently and accurately estimated using machine learning approaches.

This chapter analyzes the rich body of existing literature on LQEs using models developed from data traces. It shows that the ML-based techniques used for modelling link quality are becoming increasingly sophisticated and accurate, which confirms hypothesis **H1**. The analysis of the literature also shows that authors with the proposed ML-based models already outperform rule-based or algorithm-based estimators.

As to the contributions outlined in Chapter 1.3, this chapter represents contribution **C1**. This chapter provides an in-depth and comprehensive survey of existing data-driven approaches for wireless link quality estimation, which serves as a foundation for the subsequent research of ML-based LQE.

The publication included in this chapter is:

- G. Cerar, H. Yetgin, M. Mohorčič and C. Fortuna, *Machine Learning for Wireless Link Quality Estimation: A Survey* in IEEE Communications Surveys & Tutorials, doi: 10.1109/COMST.2021.3053615.

696 IEEE COMMUNICATIONS SURVEYS & TUTORIALS, VOL. 23, NO. 2, SECOND QUARTER 2021

Machine Learning for Wireless Link Quality Estimation: A Survey

Gregor Cerar, *Graduate Student Member, IEEE*, Halil Yetgin, *Member, IEEE*, Mihael Mohorčič, *Senior Member, IEEE*, and Carolina Fortuna

***Abstract*—Since the emergence of wireless communication networks, a plethora of research papers focus their attention on the quality aspects of wireless links. The analysis of the rich body of existing literature on link quality estimation using models developed from data traces indicates that the techniques used for modeling link quality estimation are becoming increasingly sophisticated. A number of recent estimators leverage Machine Learning (ML) techniques that require a sophisticated design and development process, each of which has a great potential to significantly affect the overall model performance. In this article, we provide a comprehensive survey on link quality estimators developed from empirical data and then focus on the subset that use ML algorithms. We analyze ML-based Link Quality Estimation (LQE) models from two perspectives using performance data. Firstly, we focus on how they address quality requirements that are important from the perspective of the applications they serve. Secondly, we analyze how they approach the standard design steps commonly used in the ML community. Having analyzed the scientific body of the survey, we review existing open source datasets suitable for LQE research. Finally, we round up our survey with the lessons learned and design guidelines for ML-based LQE development and dataset collection.**

***Index Terms*—Link quality estimation, machine learning, data-driven model, reliability, reactivity, stability, computational cost, probing overhead, dataset preprocessing, feature selection, model development, wireless networks.**

I. Introduction

IN WIRELESS networks, the propagation channel conditions for radio signals may vary significantly with time and space, affecting the quality of radio links [1]. In order to ensure a reliable and sustainable performance in such networks, an effective link quality estimation (LQE) is required by some protocols and their mechanisms, so that the radio link parameters can be adapted and an alternative or more reliable channel can be selected for wireless data transmission. To put it simply, the better the link quality, the higher the ratio of successful reception and therefore a more reliable communication. However, challenging factors that directly affect the quality of a link, such as channel variations, complex interference patterns and transceiver hardware impairments just to name a few, can unavoidably lead to unreliable links [2]. On one hand, incorporating all these factors in an analytical model is infeasible and thus such models cannot be readily adopted in realistic networks due to highly arbitrary and dynamic nature of the propagation environment [3]. On the other hand, effective prediction of link quality can provide great performance returns, such as improved network throughput due to reduced packet drops, prolonged network lifetime due to limited retransmissions [4], constrained route rediscovery, limited topology breakdowns and improved reliability, which reveal that the quality of a link influences other design decisions for higher layer protocols. Eventually, variations in link quality can significantly influence the overall network connectivity. Therefore, effectively estimating or predicting the quality of a link can provide the best performing link from a set of candidates to be utilized for data transmission.

More broadly, the quality of a wireless link is influenced by the design decisions taken for: i) wireless channel, ii) physical layer technology, and iii) link layer, as depicted in Fig. 1.

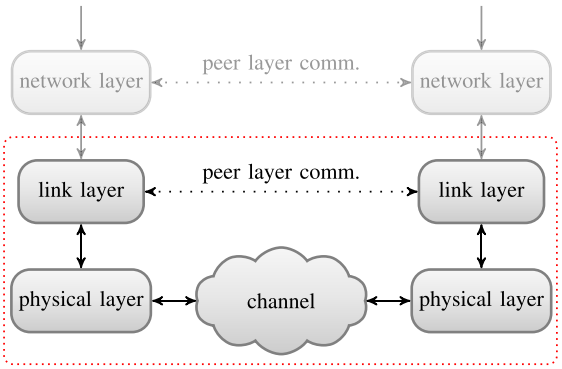


Fig. 1. The unified model of data-driven LQE comprising of physical layer (layer 1) and link layer (layer 2).

Manuscript received June 19, 2020; revised November 30, 2020; accepted January 16, 2021. Date of publication January 22, 2021; date of current version May 21, 2021. This work was supported in part by the Slovenian Research Agency under Grant P2-0016 and Grant J2-9232, and in part by the European Community H2020 NRG-5 Project under Grant 762013. *(Corresponding author: Mihael Mohorčič.)*

Gregor Cerar and Mihael Mohorčič are with the Department of Communication System, Jožef Stefan Institute, SI-1000 Ljubljana, Slovenia, and also with the Jožef Stefan International Postgraduate School, Jožef Stefan Institute, SI-1000 Ljubljana, Slovenia (e-mail: gregor.cerar@ijs.si; miha.mohorcic@ijs.si).

Halil Yetgin is with the Department of Communication Systems, Jožef Stefan Institute, SI-1000 Ljubljana, Slovenia, and also with the Department of Electrical and Electronics Engineering, Bitlis Eren University, 13000 Bitlis, Turkey (e-mail: halil.yetgin@ijs.si; hyetgin@beu.edu.tr).

Carolina Fortuna is with the Department of Communication Systems, Jožef Stefan Institute, SI-1000 Ljubljana, Slovenia (e-mail: carolina.fortuna@ijs.si).

Digital Object Identifier 10.1109/COMST.2021.3053615

TABLE I
EXISTING SURVEYS AND TUTORIALS RELATING TO THE TERMS THAT CAN DEFINE THE QUALITY OF A LINK IN THE STATE-OF-THE-ART LITERATURE

Publication	A summary with particular focus	Related context in the relevant publication	Its related section
[2], 2012	A survey on empirical studies of low power links in wireless sensor networks as well as on LQE without paying any special attention to procedures using ML techniques	Characteristics of low-power links and link quality estimation	Section V
[48], 2012	A tutorial on improving the reliability of wireless communication links using cognitive radios	Failures in wireless networks	Section II-B
[49], 2013	A survey of the techniques and protocols to handle mobility in wireless sensor networks	Prediction of link quality for mobility estimation	Section IV
[50], 2014	A survey on fair resource sharing/allocation in wireless networks	The impact of link quality on packet delay	Section III-B
[51], 2016	A survey of communication related issues in unmanned aerial vehicle communication networks	Dynamic topology changes and time-varying links	Sections I-B/I-C
[52], 2018	A survey on link- and path-level reliable data transfer schemes in underwater acoustic networks	Channel quality control on physical layer as shown in Table II	Section III
[53], 2018	A tutorial on key technologies of cloud access radio network optical fronthaul	Link performances of radio over fiber transport schemes illustrated in Table X	Section VII-E
[54], 2018	A survey on deep learning applications for different layers of wireless networks	A brief discussion on deep learning for link evaluation	Section IV-C
[55], 2019	A survey on deep learning techniques applied to mobile and wireless networking research	Deep learning driven network control and network-level mobile data analysis	Sections I/VI
[56], 2019	A survey of effective capacity models used in various wireless networks	A brief discussion on selection of better quality links	Section VII-B
[57], 2019	A survey of current issues and machine learning solutions for massive machine type communications in ultra-dense cellular Internet of things networks	Learning link quality and reliability to adapt communication parameters	Section VI-A
This survey	A comprehensive survey of data-driven LQE models, application quality aspects regarding the development of ML-based LQE models, ML design process for LQE models and publicly available trace-sets suitable for LQE research. Additionally, we provide a comprehensive performance data for wireless link quality classification and for design decisions taken throughout the LQE model development. Finally, we also put forward a comprehensive lessons learned section for the development of ML-based LQE model as well as the design guidelines for ML-based LQE development and dataset collection.	Data-driven link quality estimation models	All sections

The channel used for communication can be described by several parameters, such as operating frequency, transmission medium (e.g., air, water), environment (e.g., indoor, outdoor, dense urban, suburban) as well as relative position of the communicating parties (e.g., line-of-sight, non-line-of-sight) [1]. The physical layer technology implemented at the transmitter and receiver comprises several complex and well-engineered blocks, such as the antenna (e.g., single, multiple or array), frequency converter, analog to digital converter, synchronization and other baseband operations. The link layer is responsible for successfully delivering the data frame via a single wireless hop from transmitter to receiver, therefore it comprises of frame assembly and disassembly techniques, such as attaching/detaching headers, encoding/decoding payload, as well as mechanisms for error correction and controlling retransmissions [3]. While the quality of a link is ultimately influenced by a sequence of complex, well studied, designed and engineered processing blocks, the performance of the realistic and operational systems is quantified by a relatively limited number of observations [2], the so-called *link quality metrics*, which are detailed later in Section II-C using Table IV.

In this article, we refer to the wireless link abstraction as comprising of link layer and physical layer. More explicitly, *link quality* is referred to the quality of a wireless link that is concerned with the link layer and the physical layer. The LQE models reviewed in this survey paper are based on physical and link layer metrics, namely all potential metrics for the evaluation of link quality that lie within the dotted rectangle of Fig. 1.

To briefly overview, the research on data-driven LQE using real measurement data started in the late 90s [5] and is still carried on with a plethora of publications in the last decade [5]–[16]. Early studies on this particular topic mainly utilized recorded traces and the models were developed manually [5], [7]–[16]. Over the past few years, researchers have paid a lot of attention to the development of LQE using ML algorithms [6], [17]–[19].

A. Applications of ML in Wireless Networks

The use of ML techniques in LQE is promising to significantly improve the performance of wireless networks due to the ability of the technology to process and learn from large amount of data traces that can be collected across various technologies, topologies and mobility scenarios. These characteristics of ML techniques empower LQE to become much more agile, robust and adaptive. Additionally, a more generic and high level understanding of wireless links could be acquired with the aid of ML techniques. More explicitly, an intelligent and autonomous mechanism for analyzing wireless links of any transceiver and technology can assist in better handling of current operational aspects of increasingly heterogeneous networks. This opens up a new avenue for wireless network design and optimization [58], [59] and calls for the ML techniques and algorithms to build robust, agile, resilient and flexible networks with minimum or no human intervention. A number of contributions for such mechanisms can be found in the literature, for instance radio spectrum observatory network is designed in [60] and [61].

The diagram provided in Fig. 2 exhibits a broad picture of what problems are being solved by ML in wireless networks and what broad classes of ML methods are being used for solving these particular problems. It can be observed that improvements on all layers of the communication network stack, from

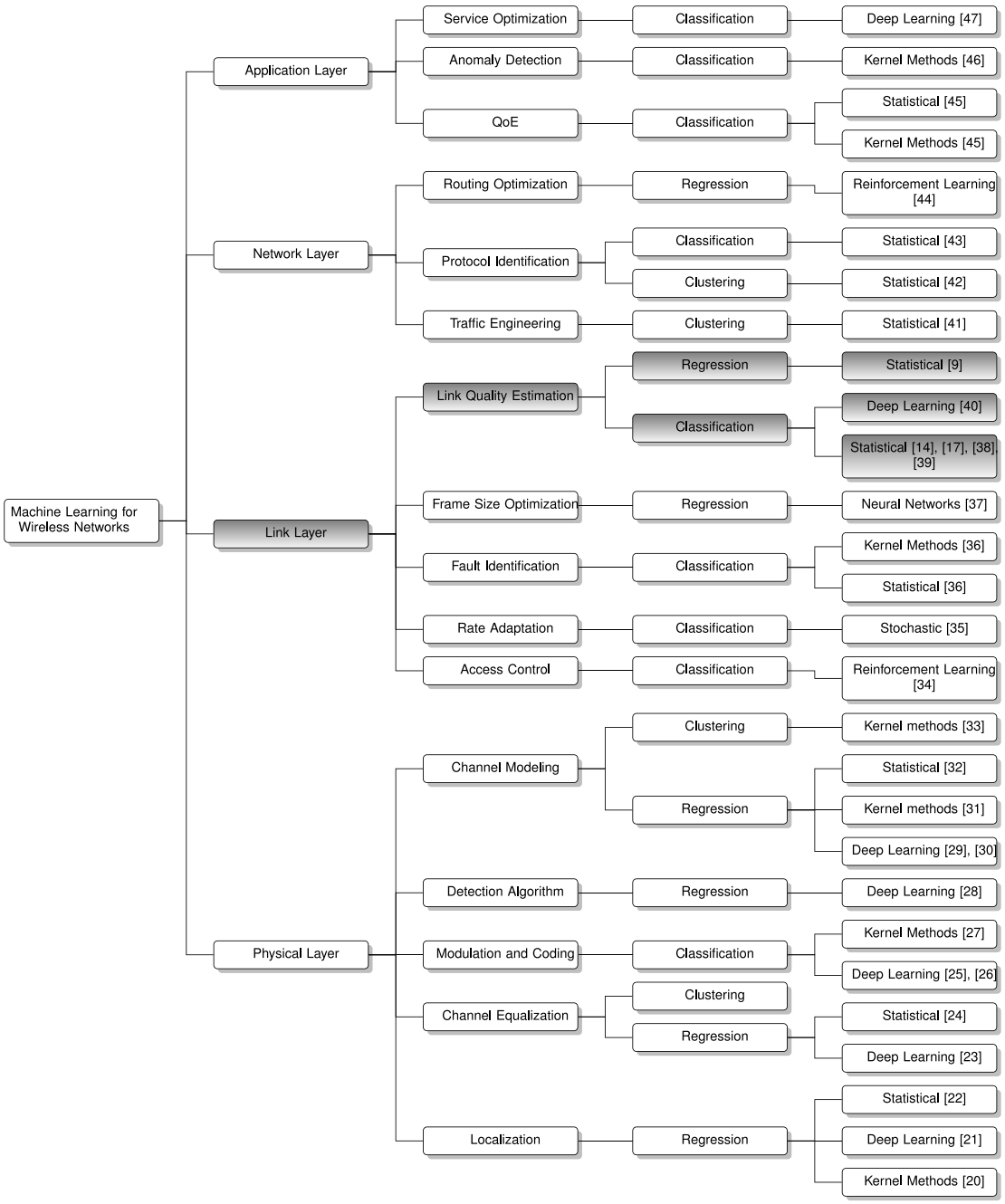


Fig. 2. Layered taxonomy of machine learning solutions for wireless communication networks.

physical to application, are being proposed using classification, regression and clustering techniques. For each technique, algorithms having statistical, kernel, reinforcement, deep learning, and stochastic flavors are being used. The scope of the ML works analyzed in this article is shaded with gray in Fig. 2 and further detailed later in Fig. 5. For a more comprehensive and intricate analysis, [54] and [55] survey deep learning in wireless networks, and [62] surveys Artificial Intelligence (AI)

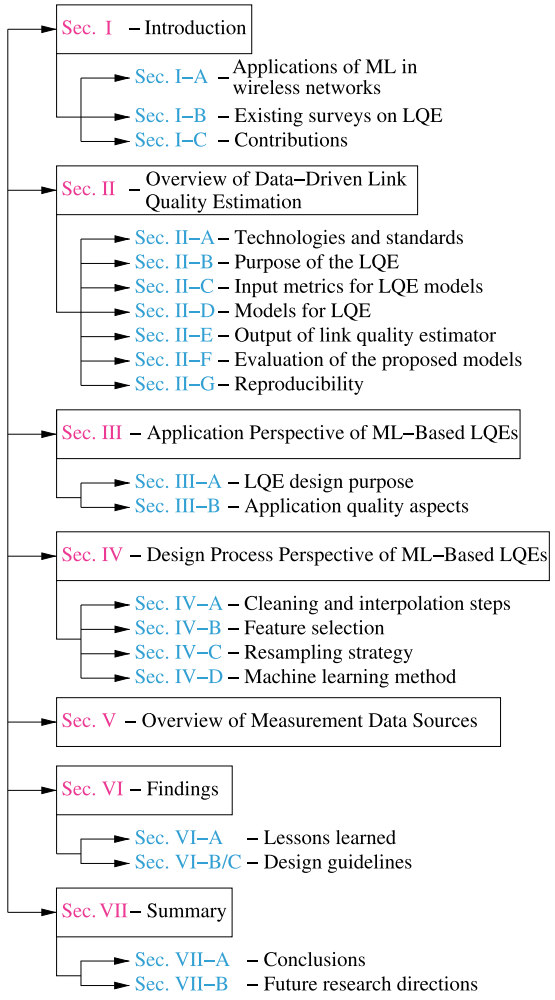


Fig. 3. Structure overview of this survey paper.

techniques, including ML and symbolic reasoning in communication networks, but without investing any particular effort on LQE.

B. Existing Surveys on LQE

To contrast our study against existing survey papers on the aspects of link quality estimation, we have identified a comprehensive list of survey and tutorial papers summarized in Table I. We have observed that there are existing discussions on the "link quality" considering various wireless networks, as outlined in Table I. However, only Baccour *et al.* attempted to address LQE in [2]. They highlighted distinct and sometimes contradictory observations coming from a large amount of research work on LQE based on different platforms, approaches and measurement sets. Baccour *et al.* provide a survey on empirical studies of low power links in wireless sensor networks[1] without paying any special attention to procedures using ML techniques. In this survey paper, we complement the aforementioned survey by analyzing the rich body of existing and recent literature on link quality estimation with the focus on model development from data traces using ML techniques. We analyze the ML-based LQE from two complementary perspectives: application requirements and employed design process. First, we focus on how they address quality requirements that are important from the perspective of the applications they serve in Section III. Second, we analyze how they approach the standard design steps commonly used in the ML community in Section IV. Moreover, we also review publicly available data traces that are most suitable for LQE research.

C. Contributions

Considering recent contributions on LQE using ML techniques, it can be challenging to reveal the relationship between design choices and reported results. This is mainly because each model relying on ML assumes a complex development process [63], [64]. Each step of this process has a great potential to significantly affect the overall performance of the model, and hence these steps and their associated design choices must be well understood and carefully considered. Additionally, to provide the means for fair comparison between existing and future approaches, it is of critical importance to be able to reproduce the LQE model development process and results [65]–[67], which indeed also requires open sharing of data traces.

The major contributions of this article can be summarized as follows.

- We provide a comprehensive survey of the existing literature on LQE models developed from data traces. We analyze the state of the art from several perspectives including target technology and standards, purpose of LQE, input metrics, models utilized for LQE, output of LQE, evaluation and reproducibility. The survey reveals that the complexity of LQE models is increasing and that comparing LQE models against each other is not always feasible.
- We provide a comprehensive and quantitative analysis of wireless link quality classification by extracting the approximated per class performance from the reported results of the literature in order to enable readers to readily distinguish the performance gaps at a glimpse.
- We analyze the performance of candidate classification-based LQEs and reveal that autoencoders, tree based methods and SVMs tend to consistently perform better than logistic regression, naive Bayes and artificial neural networks whereas the non-ML TRIANGLE estimator performs considerably well on the two, i.e., *very good* and *good* quality links, of the five classes included in the analysis.
- We identify five quality aspects regarding the development of an ML-based LQE that are important from the

[1]This survey paper is also a more recent contribution on link quality estimation models than [2] from 2012. Besides, we focus our attention on the data-driven LQE models with ML techniques.

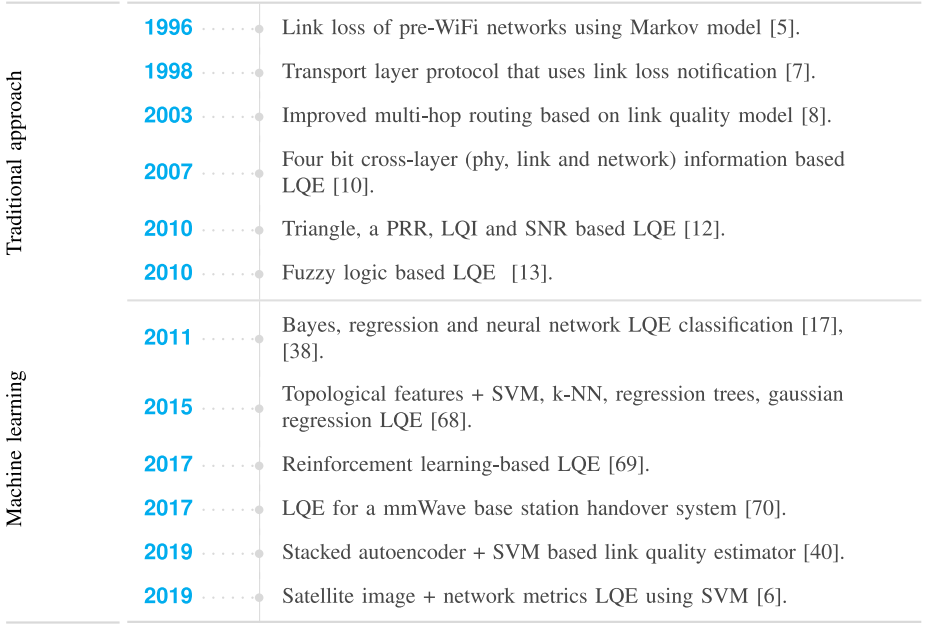


Fig. 4. Timeline of the most prominent models in the evolution of wireless LQE.

application perspective: reliability, adaptivity/reactivity, stability, computational cost and probing overhead. We provide insightful analyses on how ML-based LQE models address these five quality aspects considering the use of ML methods for a diverse set of specific problems.
- Starting from the standard ML design process, we investigate and quantify the design decisions that the existing ML-based LQE models considered and provide insights for their potential impact on the final performance of the LQE using the accuracy as well as the F1 score and precision vs. recall metrics.
- We survey publicly available datasets that are most suitable for LQE research and review their available features with a comparative analysis.
- We provide an elaborated lessons learned section for the development of ML-based LQE model. Based on the lessons learned from this survey paper, we derive generic design guidelines recommended for the industry and research community to follow in order to effectively design the development process and collect trace-sets for the sake of LQE research.

The rest of this article is structured as portrayed in Fig. 3. Section II provides a comprehensive survey of the state-of-the-art literature on LQE models built from data traces. Section III and Section IV analyze ML-based LQE models from the perspective of application requirements, and of the design process, respectively. Section V then provides a comprehensive analysis of the open datasets suitable for LQE research. As a result of our extensive survey, Section VI provides lessons learned and design guidelines, while Section VII finally concludes the paper and elaborates on the future research directions.

II. Overview of Data-Driven Link Quality Estimation

With the emergence and spread of wireless technologies in the early 90s [71], it became clear that packet delivery in wireless networks was inferior to that of wired networks [5]. At the time of the experiment conducted in [5], wireless transmission medium was observed to be prone to unduly larger packet losses than the wired transmission mediums. Up until today, roughly speaking, numerous sophisticated communication techniques, including modulation and coding schemes, channel access methods, error detection and correction methods, antenna arrays, spectrum management, high frequency communications and so on, have emerged. As part of this combination of revolutionary techniques, a diverse number of estimation models for the assessment of link quality, based on actual data traces in addition to or instead of simulated models, have been proposed in the literature.

The research of data-driven LQE based on measurement data reaches back into late 90s [5] and has gained momentum particularly in the last decade [6]. As summarized in the timeline depicted in Fig. 4, early attempts on LQE research mainly hinge on the recorded traces with statistical approaches and the manually developed models [5], [7]–[16]. On the other hand, only after 2010, researchers have started paying a great attention to the development of LQE model using ML algorithms [17]–[19].

To date, many analytical and statistical models have been proposed to mitigate losses and improve the performance of wireless communication. These models include channel models, radio propagation models, modulation/demodulation and encoding/decoding schemes, error correction codes, and

multi-antenna systems just to name a few. Such models are essentially based on model-driven link quality estimators, where they calculate predetermined variables based on the communication parameters of the associated environment. However, their one significant shortcoming is that they abstract the real environment, and thus consider only a subset of the real phenomena. Data-driven models, on the other hand, rely on actual measured data that capture the real phenomena. The data are then used to fit a model that best approximates the underlying distribution. As it can be readily seen in Fig. 4, up until 2010, statistical approaches were the favored tools for LQE research. From then on, as in other research areas of wireless communication, portrayed in Fig. 2, ML-based models replaced the conventional approaches and became the preferred tool for LQE research.

Empirical observation of wireless link traffic is a crucial part of the data-driven LQE. An observation of link quality metrics within a certain estimation window, e.g., time interval or a discrete number of events, allows for constructing different varieties of data-driven link quality estimators. However, there are a few drawbacks of the data-driven approaches that need to be taken into account. Since the ultimate model strictly depends on the recorded data traces, it has to be carefully designed in a way that records adequate information about the underlying distribution of the phenomena. If sufficient measurements of the distribution can be captured, then it is possible to automatically build a model that can approximate that particular distribution. Data-driven LQE models are in no way meant to fully replace or supersede model-driven estimators but to complement them. It is certainly possible to incorporate a model-driven estimator into a data-driven one as the input data.

To some extent, different varieties of data-driven metrics and estimators were studied in [2], where the authors made three independent distinctions among hardware- and software-based link quality estimators. The software-based estimators are further split into Packet Reception Ratio (PRR)-based, Required Number of Packets (RNP)-based, and score-based subgroups. The first distinction is based on the estimator's origin presenting the way how they were obtained. The second distinction is based on the mode their data collection was done, which can be in passive, active and/or hybrid manner, depending on whether dummy packet exchange was triggered by an estimator. The third distinction is based on which side of the communication link was actively involved. LQE metrics can be gathered either on the receiver, transmitter or both sides.

Going beyond [2], Tables II and III provide a comprehensive summary of the most related publications that leverage a data-driven approach for LQE research. All the studies summarized in Tables II and III rely on real network data traces recorded from actual devices. The first column in Tables II and III contains the title, reference and the year of publication. The second column provides the testbed, the hardware and the technology used in each publication, whereas the third column lists the objectives of these publications with respect to LQE approach. Columns four, five and six focus on the characteristics of the estimators, particularly on their corresponding statistical aspects of the data traces and their public availability of the trace-sets for reproducibility, respectively.

A. Technologies and Standards

As outlined in the second column of Tables II and III, earlier studies on LQE were performed on WaveLAN [5], [7], a precursor on the modern Wi-Fi. The study in [5] aimed to characterize the loss behavior of proprietary AT&T WaveLAN. It used packet traces with various configurations for the transmission rate, packet size, distance and the corresponding packet error rate. Then, they built a two-state Markov model of the link behavior. The same model was then utilized in [7] to estimate the quality of wireless links in the interest of improving Transmission Control Protocol (TCP) congestion performance. More recently, [70], [72] used IEEE 802.11 standard in their studies for throughput and online link quality estimators.

Later on, the majority of publications related to LQE focused on wireless sensor networks relying on IEEE 802.15.4 standard and only a few targeted other type of wireless networks, such as Wi-Fi (IEEE 802.11) or Bluetooth (IEEE 802.15.1). This can be explained by the fact that IEEE 802.15.4-based wireless sensor networks are relatively cheaper to deploy and maintain. Perhaps, the first such larger testbed was available at the University of Berkley [8] using MicaZ nodes and TinyOS [73], which is an open source operating system for constrained devices. Other hardware platforms, such as TelosB and TMote, and operating systems, e.g., Contiki, have emerged and enabled researchers to further experiment with improving the performance of single and multi-hop communications for wireless networks composed of battery-powered devices.

Finally, one recent contribution focuses on LoRA technology, a type of Low Power Wide Area Network (LPWAN) for estimating the quality of links, and therefore aiming for the improvement of the coverage for the technology [6].

Whereas earlier research on LQE leveraged proprietary technologies [5], wireless sensor networks utilized relatively low cost hardware and open source software, therefore enabled a broader effort from the research community. This resulted in a large wave of research focusing on ad-hoc, mesh and multihop communications [8], [10], [13]–[17], [19], [38], [40], [74], all of which rely on the estimation of link quality. The nodes implementing the aforementioned technologies are still being maintained in various university testbeds.

B. Purpose of the LQE

With respect to the research goal summarized in the third column of Tables II and III, the surveyed papers can be categorized into two broad groups. The goal of the first group was to improve the performance of a protocol or process. The goal of the second group of papers was to propose a new or improve an existing link quality estimator. For this class of papers, any protocol improvement in the evaluation process was secondary.

1) LQE for Protocol Performance Improvement: The authors of [5], [7] investigated TCP performance improvement,

TABLE II
EXISTING WORK ON LINK QUALITY ESTIMATION USING REAL NETWORK DATA TRACES (PART 1 OF 2)

Title	Tech.	Goal	Input	Model	Output	Data	Reproduce
A trace-based approach for modeling wireless channel behavior [5], 1996	WaveLAN, BARWAN testbed, BSD 2.1	Maximize throughput, channel error model	SNR, signal quality, throughput, PRR	Improved two-state Markov model	Probability of error to occur and persist	Not specified (<1500 bytes/-packet, 1000 s/trace)	No*
Explicit loss notification and wireless web performance [7], 1998	WaveLAN, University of California testbed	Improve TCP Reno on wireless links, maximize throughput	Bitrate, packet size, no. bits, throughput, BER	CDF of error and error-free durations	Probability of error to occur and persist	800 000 packets (100 000 packets/-experiment, 8 experiments)	No*
Taming the underlying challenges of reliable multihop routing in sensor networks [8], 2003	Proprietary, MicaZ mote, TinyOS	Improve routing table management	PRR	Shortest path, minimum transmission, broadcast, destination sequenced distance vector	Decision on keep/remove routing table entry	≈600 000 packets (8 packets/s, 200 packets/P_{Tx})	No*
(4B) Four-bit wireless link estimation [10], 2007	Intel Mirage: 85x MicaZ; USC TutorNet: 94x TelosB; IEEE 802.15.4, TinyOS	Improve routing table management	LQI, PRR, broadcast, ACK count	Construct 4-bit score of link state	Estimated link quality	Mirage: N.A., 40-69 min/experiment; TutorNet: N.A., 3-12h/experiment;	No*
A Kalman filter-based link quality estimation scheme for wireless sensor networks [9], 2007	TelosB, IEEE 802.15.4	PRR estimation	RSSI, noise floor	Kalman filter + SNR to PRR mapping	PRR estimation	25 200 000 (500 samples/s, 14 h)	No
PRR is not enough [11], 2008	IEEE 802.11, IEEE 802.15.4	Link state estimation	PRR	Gilbert-Elliott Model (2-state Markov process); *good* and *bad* state	Link quality transition probability	Rutgers and Mirage trace-sets	Yes
The triangle metric: fast link quality estimation for mobile wireless sensor networks [12], 2010	Tmote Sky, Sentilla JCreate, IEEE 802.15.4, Contiki OS	New LQE	RSSI, noise floor, LQI	Pythagorean equation maps to distance from the origin (hypotenuse)	Estimated link quality as *very good*, *good*, *average* or *bad*	30 000 + N.A., (64 packets/s, all channels, unicast)	No
F-LQE: A fuzzy link quality estimator for wireless sensor networks, [13] 2010, [75] 2011	RadiaLE testbed, 49x TelosB, IEEE 802.15.4, TinyOS	Link quality estimation, improve routing	PRR	Fuzzy logic maps current to estimated link quality	Binary high/low-quality (HQ/LQ) link estimation	N.A. (bursts, packet sizes, 20-26 channel)	No*
Foresee (4C): Wireless link prediction using link features [17], 2011	54x Tmote (local), 180x Tmote Sky (Motelab), IEEE 802.15.4,	Improve routing	PRR, RSSI, SNR, LQI	Logistic regression model	Probability of receiving next packet	80 000 + 80 000 noise floor (≈10 packets/s)	No*
Fuzzy logic-based multidimensional link quality estimation for multihop wireless sensor networks [14], 2013	(local) 15x TelosB, TinyOS, IEEE 802.15.4	Improve routing, minimize topology changes	PRR	Fuzzy logic link quality estimator	Binary high/low-quality link estimation	N.A., (20 min/experiment, 12h)	No
Temporal adaptive link quality prediction with online learning, [38] 2012, [18] 2014	Motelab, Indriya and (local) 54x Tmote testbed, IEEE 802.15.4	Link quality estimation, improve Routing	PRR, RSSI, SNR, LQI	Logistic regression with SGD and s-ALAP adaptive learning rate	Binary, estimates if link quality above desired threshold	480 000, (30 bytes size, 6 000 per exp., 10/sec.), Rutgers and Colorado trace-sets	No [38] Yes [18]
Low-Power link quality estimation in smart grid environments [15], 2015	IEEE 802.15.4	Improve routing, LQE reactivity	RNP, SNR, PRR	Optimized F-LQE [13] with better reactivity	Binary high/low-quality link estimation	N.A., 500kV substation env. data, TOSSIM 2 simulator	No
Time series analysis to predict link quality of wireless community networks [68], 2015	Conventional routers, IEEE 802.15.4, IEEE 802.11, AX.25, (FunkFeuer mesh network)	Link quality estimation, regression, clustering, time-series analysis	LQ, NLQ, ETX	SVM, k-nearest neighbor, regression trees, Gaussian process for regression	Predicted LQ value for different windows sizes	N.A., (404 nodes, 2 095 links, 7 days of data)	No*
Machine-learning based channel quality and stability estimation for stream-based multichannel wireless sensor networks [76], 2016	CC2420, IEEE 802.15.4, Matlab simulation	Evaluation of new algorithm with two possible extensions	RSSI, LQI, channel rank, channel	Normal equation-based channel quality prediction, weighted input extension, stability extension	Channel quality estimation based on 3-class estimator	Simulation	Yes
WNN-LQE: Wavelet-neural-network-based link quality estimation for smart grid WSNs [19], 2017	10x CC2530 WSNs, IEEE 802.15.4	Improve routing, estimate PRR range	SNR	Wavelet-neural-network-based link quality estimator	Upper and lower bound of confidence interval for PRR	2 500 (20 bytes size, 3.33 per second)	No

Note: Asterisk (*) indicates that the experiment was performed on a public testbed, but no data is available.

group of papers proposed a novel link quality estimators as an intermediate step towards achieving their goal, e.g., performance improvement of TCP, routing optimization and so on.

One of the earliest publications from this group is [8] that aimed for improving the reactivity of routing tables in constrained devices, such as sensor nodes. They collected traces of transmissions for nodes located at various distances

TABLE III
EXISTING WORK ON LINK QUALITY ESTIMATION USING REAL NETWORK DATA TRACES (PART 2 OF 2)

Title	Tech.	Goal	Input	Model	Output	Data	Reproduce
A reinforcement learning-based link quality estimation strategy for RPL and its impact on topology management [69], 2017	**Sim.:** Cooja simulator (Contiki 3.x); **Exp.:** 23x TelosB, CC2420, IEEE 802.15.4	Improve RPL protocol	PER, RSSI, energy consumption	Unsupervised ML	PRR estimation	**Sim.:** ∞; **Exp.:** N.A., 178 links, mobile nodes (0.5 m/s), University of Pisa	**Sim.:** Yes; **Exp.:** No
Research on Link Quality Estimation Mechanism for Wireless Sensor Networks Based on Support Vector Machine [74], 2017	2x TelosB, CC2420, IEEE 802.15.4, TinyOS 2.x	link quality estimation, comparison	RSSI, LQI, PRR	SVM classifier	Classification, 5 classes	121 datapoints	No
Machine-learning-based throughput estimation using images for mmWave communications [70], 2017	2x IEEE 802.11ad @ 60 GHz (mmWave), RGB-D camera (Kinect)	Throughput estimation, obstacle detection, comm. handover w/o control frames	Throughput, depth value (Kinect)	Online adaptive regularization of weight vectors (AROW)	regression, throughput estimation	N.A.	No
Quick and efficient link quality estimation in wireless sensors networks [16], 2018	Grenoble testbed FIT-IoT, 28x AT86RF231, IEEE 802.15.4	Analysis of LQI, fast decisions, improve routing	LQI	Classification based on arbitrary values	Classify link as *good*, *uncertain* or *weak*	N.A. (2 000 per link, 16 channels)	No*
Online ML algorithms to predict link quality in community wireless mesh networks [72], 2018	Conventional routers, IEEE 802.15.4, IEEE 802.11, AX.25, (FunkFeuer mesh network)	Link quality estimation, online regression, compares online ML algorithms	LQ, NLQ, ETX	online perceptrons, online regression trees, fast incremental model trees, adaptive model rules	Metric estimation, regression	N.A. ($\approx$500 nodes, $\approx$2 000 links, FunkFeuer distributed community network)	No*
Link Quality Estimation Method for Wireless Sensor Networks Based on Stacked Auto-encoder [40], 2019	8x TelosB, TinyOS, IEEE 802.15.4	Link quality estimation, classification	SNR, RSSI, LQI, and PRR from transmitter and receiver	Neural network-based classification	Estimated link quality as *very bad*, *bad*, *common*, *good*, *very good*	N.A., interior corridors, grove, parking lots, road	No
Automated Estimation of Link Quality for LoRa: A Remote Sensing Approach [6], 2019	Dragino LoRa 1.3 (RF96 chip), LoRa	Link quality estimation, environment classification	Node/Gateway position, time-stamp, RSSI, SNR, multispectral aerial images	SVM classification of LoRa coverage	Mapping LoRa coverage onto geographical map	8 642 samples, 23 sites, 1 packet per 40s, Delft (NL)	No
On Designing a Machine Learning Based Wireless Link Quality Classifier [39], 2020	29x IEEE 802.11	Link quality prediction, importance of preprocessing	RSSI	logistic, regression, SVM, decision trees, random forest, multi-layer perceptron	Classification of future link state as *good*, *intermediate* or *bad*	Rutgers dataset	Yes

Note: Asterisk (*) indicates that the experiment was performed on a public testbed, but no data is available.

with respect to each other. Then, they computed reception probabilities as a function of distances and evaluated a number of existing link estimation metrics. They also proposed a new link estimation metric called Window Mean with an Exponentially Weighted Moving Average (WMEWMA) and showed an improvement in network performance as a result of more appropriate routing table updates. The improvements were shown both in simulations and in experimentation. This study was also among the earliest studies introducing the three different grade regions of wireless links, i.e., *good*, *intermediate* and *bad*.

Later, [10] noticed that by considering additional metrics alongside WMEWMA, also from higher levels of the protocol stack, the link estimation could be better coupled with data traffic. Therefore, they introduced a new estimator referred to as Four-Bit (4B), where they combined information from the physical (PRR, Link Quality Indicator (LQI)), link (ACK count) and network layers (routing) and demonstrated that it performs better than the baseline they chose for the evaluation.

In [13], the authors developed a new link quality estimator named Fuzzy-logic based LQE (F-LQE) that is based on fuzzy logic, which exploits average values, stability and asymmetry properties of PRR and Signal-to-Noise Ratio (SNR). As for the output, the model classifies links as high-quality (HQ) or low-quality (LQ). The same authors compared F-LQE against PRR, Expected Transmission count (ETX) [77], RNP [78] and 4B [10] on the RadiaLE testbed [75]. The comparison of the metrics was performed using different scenarios including various data burst lengths, transmission powers, sudden link degradation and short bursts. Among their findings, they showed that PRR, WMEWMA and ETX, which are PRR-based link quality estimators, overestimate the link quality, while RNP and 4B underestimate the link quality. The authors of [75] demonstrated that F-LQE performed better estimation than the other estimators compared.

The authors of [14] used fuzzy logic and proposed a Fuzzy-logic Link Indicator (FLI) for link quality estimation. The FLI model uses PRR, the coefficient of variance of PRR and the quantitative description of packet loss burst, which

TABLE IV
METRICS THAT CAN BE USED TO MEASURE THE QUALITY OF A LINK

Link quality metrics	Hardware based	Software-based			Image based	Topological	Sides involved		Gathering method		Related base-metric(s)
		PRR-based	RNP-based	Score-based			Rx	Tx	Passive	Active	
RSSI	✓						✓		✓		RSS, SNR
LQI	✓						✓		✓		Vendor-specific
SNR	✓						✓		✓		RSS, noise floor
BER	✓						✓		✓		–
PRR		✓					✓		✓		PER
WMEWMA		✓					✓		✓		PER, PRR
4B			✓				✓	✓	✓	✓	LQI, PRR, ACK, broadcast
LQ, NLQ			✓				✓	✓		✓	–
ETX			✓				✓	✓		✓	LQ, NLQ
4C				✓			✓		✓		LQI, PRR, SNR, RSSI
TRIANGLE				✓			✓		✓		SNR, LQI
Image-based					✓						
Topological						✓					

are gathered independently, while the previous F-LQE [13] requires information sharing of PRR. FLI was evaluated in a testbed for 12 hours worth of simulation time against 4B [10], and it was reported to perform better.

Foresee (4C) [17] is the first metric from this group focused on protocol improvement that introduced statistical ML techniques. The authors used Received Signal Strength Indicator (RSSI), SNR, LQI, WMEWMA and smoothed PRR as input features into the models. They trained three ML models based on naïve Bayes, neural networks and logistic regression. TALENT [38] was then improved on 4C by introducing adaptive learning rate.

More recently, [69] proposed enhancement to the RPL protocol, which is used in lossy wireless networks. This backward compatible improvement (mRPL) for mobile scenarios introduces asynchronous transmission of probes, which observe link quality and trigger the appropriate action.

New or Improved Link Quality Estimator: Srinivasan *et al.* [11] proposed a two-state model with *good* and *bad* states, and 4 transition probabilities between the states to improve on the existing WMEWMA [10] and 4B [10]. Then, Senel *et al.* [9] took a different approach and developed a new estimator by predicting the likelihood of a successful packet reception. Besides, Boano *et al.* [12] introduced the TRIANGLE metric that uses the Pythagorean equation and computes the distance between the instant SNR and LQI. This study identifies four different link quality grades including *very good*, *good*, *average* and *bad* links. Some of the classifiers propose a five-class model [40], [74] and a three-class model [16], [39] for LQE research. Other LQE models leverage regression rather than classification in order to generate a continuous-valued estimate of the link [6], [70], [72].

C. Input Metrics for LQE Models

With respect to the input metrics used for estimating the quality of a link summarized in the fourth column of Tables II and III, we distinguish between the single and the multiple metric approaches. Single metric approaches use a one dimension vector while multiple metric approaches use a multidimensional vector as input for developing a model.

Single metric input approaches have a number of advantages. The trace-set is smaller and thus often easier to collect, the model typically requires less computational power to compute, and as shown in [17] they can be more straightforward to implement, especially on constrained devices. However, by only analyzing and relying on a single measured variable, such as RSSI, important information might be left out. For this reason, it is better to collect traces with *several, possibly uncorrelated metrics*, each of them being able to contribute meaningful information to the final model. A good example of the latter is using RSSI and spectral images for instance.

The estimators surveyed based on single input metric appear in [8], [11], [16], [19], [39] whereas the estimators based on multiple metrics are considered in [5]–[7], [9], [10], [12]–[15], [17], [38], [40], [69], [70], [72], [74].

One can readily observe from the fourth column of Tables II and III that the most widely used metric, either directly or indirectly, is the PRR, which is used as model input in [5], [8]–[11], [13]–[15], [17], [38]. Other input metrics derived from PRR values are also used as input metrics in [9], [12]. Looking at the frequency of use, PRR is followed by hardware metrics, i.e., RSSI, LQI and SNR in [9], [10], [12], [16], [17], [19], [38]. Other features are less common and tend to appear scarcely in single papers.

Table IV summarizes metrics that can be used for measuring the quality of the link. Every metric from the first column of the table can also be used as input for another new metric. The so-called hardware-based metrics [2], such as RSSI, LQI, SNR and Bit Error Rate (BER) are directly produced by the transceivers, and they also depend on underlying metrics, such as Received Signal Strength (RSS), SNR, noise floor, implementation artifacts and vendor. The so-called software-based metrics are usually computed based on a blend of hardware and software metrics. It is clear from the first and the last columns of Table IV that the number of independent input variables is limited. However, recently, additional input has been taken into account in [68]. *Topological features* assuming cross-layer information exchange, where LQE is informed

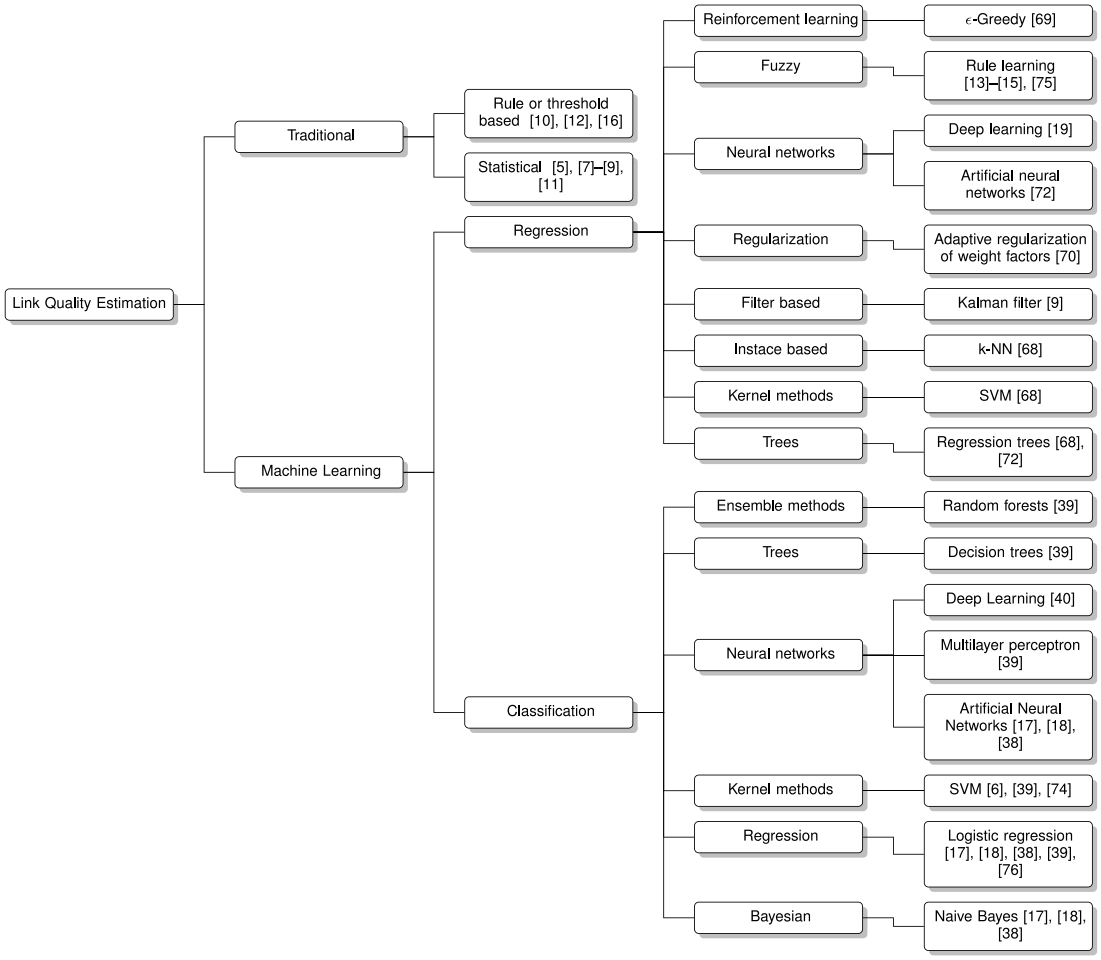


Fig. 5. Taxonomy of the LQE approaches using ML algorithms and traditional methods.

of node degree, hop count, strength and distance is considered in [68], while [70] and [6] have exclusively shown that *imaging data* can be used as input for LQE models as an alternative source of data, as outlined at the bottom of Table IV.

In addition to finding other new sources of data, a challenging task would be to analyze a large set of measurements in various environments and settings, from a large number of manufacturers to understand how measurements vary across different technologies and differ across various implementations within the same technology, and derive the truly effective metrics for an efficient development of LQE model.

D. Models for LQE

Considering the models used for developing LQE summarized in the fifth column of Tables II and III, the publications surveyed can be distinguished as those using statistical models [5], [7]–[9], [11], rule and/or threshold based models [10], [12], [16], fuzzy ML models [13]–[15], [75], statistical ML models [6], [17], [18], [38], [39], [68], [70], [72], [74], [76], reinforcement learning models [69] and deep learning models [19], [40]. For readers' convenience, the corresponding taxonomy is portrayed in Fig. 5.

With regard to the *statistical models*, the authors of [5], [7] manually derived error probability models from traces of data using statistical methods. Additionally, Woo *et al.* [8] derived an exponentially weighted PRR by fitting a curve to an empirical distribution, whereas Senel *et al.* [9] first used a Kalman filter to model the correct value of the RSS, then they extracted the noise floor from it to obtain SNR, and finally, they leveraged a pre-calibrated table to map the SNR to a value for the Packet Success Ratio (PSR). Srinivasan *et al.* [11] used the Gilbert-Elliot model, which is a two-state Markov process with *good* and *bad* states with four transition probabilities. The output of the model is the channel memory parameter that describes the "burstiness" of a link.

Considering the *rule based models*, 4B [10] constructs a largely rule based model of the channel that depends on the values of the four input metrics, whereas Boano *et al.* [12] formulate the metric using geometric rules. First, Boano *et al.* [12] computed the distance between the instant SNR and LQI vectors in a 2D space. Then, they used three empirically set thresholds to identify four different link quality grades: *very good*, *good*, *average* or *bad*. Finally, [16] manually rules out good and bad links based on LQI values and then, for the remaining links, computes additional statistics that are used to determine their quality with respect to some thresholds.

The first *fuzzy model*, F-LQE [13] uses four input metrics incorporating WMEWMA, averaged PRR value, stability factor of PRR, asymmetry level of PRR and average SNR, and fuzzy logic to estimate the two-class link quality. Rekik *et al.* [15] adapts F-LQE to smart grid environments with higher than normal values for electromagnetic radiation, in particular 50 Hz noise and acoustic noise. Finally, Guo *et al.* [14] proposed a different two-class fuzzy model based on the two input metrics, namely coefficient of variance of PRR and quantitative description of packet loss burst, which are gathered independently, and are different from the ones used for F-LQE.

One of the earliest *statistical ML model*, the so-called 4C, was proposed by Liu and Cerpa, [17], where 4C amalgamated RSSI, SNR, LQI and WMEWMA, and smoothed PRR to train three ML models based on naïve Bayes, neural networks and logistic regression algorithms. Then, Liu and Cerpa [18], [38] introduced TALENT, an online ML approach, where the model built on each device adapts to each new data point as opposed to being precomputed on a server. TALENT yields a binary output (i.e., whether PRR is above the predefined threshold), while 4C produces a multi-class output. TALENT also uses state-of-the-art models for LQE, such as Stochastic Gradient Descent (SGD) [79] and smoothed Almeida–Langlois–Amaral–Plakhov algorithm [80] for the adaptive learning rate and logistic regression.

Other statistical models, such as Shu *et al.* [74] used Support Vector Machine (SVM) algorithm along with RSSI, LQI and PRR as input to develop a five-class model of the link. Besides, Okamoto *et al.* [70] used an on-line learning algorithm called adaptive regularization of weight vectors to learn to estimate throughput from throughput and images. Then, Bote-Lorenzo *et al.* [72] trained online perceptrons, online regression trees, fast incremental model trees, and adaptive model rules with Link Quality (LQ), Neighbor Link Quality (NLQ) and ETX metrics to estimate the quality of a link, whereas Demetri *et al.* [6] benefit from a seven-class SVM classifier to estimate LoRa network coverage area by means of using 5 input metrics to train the classifier including multi-spectral aerial images. More recently, [39] evaluated four different ML models, namely logistic regression, tree based, ensemble, multilayer percepron, against each other.

The only proposed *reinforcement learning model* for link quality estimation appears in [69]. The authors train a greedy algorithm with Packet Error Rate (PER), RSSI and energy consumption input metrics to estimate PRR in view of protocol improvement in mobility scenarios.

The two LQE models using *deep learning algorithms* have also been proposed. For the first model, Sun *et al.* [19] introduce Wavelet Neural Network based LQE (WNN-LQE), a new LQE metric for estimating link quality in smart grid environments, where they only rely on SNR to train a wavelet neural network estimator in view of accurately estimating confidence intervals for PRR. In the latter model, Luo *et al.* [40] incorporate four input metrics, namely SNR, LQI, RSSI, and PRR, and trains neural networks to distinguish a five-class LQE model.

E. Output of Link Quality Estimator

Regarding the output of link quality estimators summarized in the sixth column of Tables II and III, we can observe three distinct types of the output values.

The first type is a *binary or a two-class output*, which is produced by the classification model. This type of output can be found in [8], [14], [15], [18], [75]. The applications noticed are mainly (binary) decision making [8] and above/below threshold estimation [14], [15], [18], [75].

The second type is *multi-class output* value. Similar to the first type, it is also produced by the classification model. The multi-class output values are utilized in [6], [12], [16], [40], [74], [76], where [16], [39], [76] use a three-class, [12] utilizes a four-class, [40], [74] rely on a five-class, and [6] leverages a seven-class output. The applications observed are the categorization and estimation of the future LQE state, which is expressed through labels/classes.

It is not clear from the analyzed work how the authors selected the number of classes in the case of multi-class output LQE models. The three class output models seem to be justified by the three regions of a wireless links [2]. The seven class output model [6] justifies the 7 types of classes based on seven types of geographical tiles. For the rest or the work, it is not clear what is the justification and advantage of a four, or five class LQE model. Generally, by adding more classes, the granularity of the estimation can be increased while the computing time, memory size and processing power increase.

The third type is the *continuous-valued output*. In contrast to the first two types, it is produced by a regression model, which is considered by [5], [7], [9]–[11], [17], [19], [68]–[70], [72]. The value is typically limited only by numerical precision. The applications observed are the direct estimation of a metric [5], [7], [9], [19], [68]–[70], [72], probability value [11], [17] and their proposed scoring metric [10], which are later used for comparative analysis.

Some of the proposed or identified applications require continuous-valued LQE estimation, for instance, network congestion controller (TCP Reno) [7], communication handover [70], and routing table managers [10], [17], [19], [68], [69], [72]. For other routing table managers and applications, a discrete valued LQE suffices according to the surveyed work. Note that any continuous estimator can be subsequently converted to discrete valued one.

TABLE V
A SURVEY OF THE COMPARISON FOR LQE MODELS AND THEIR RESPECTIVE EVALUATION METRICS CONSIDERING THE RESEARCH PAPERS COMPREHENSIVELY SURVEYED IN TABLES II AND III

ID	Evaluation metrics	The proposed LQE models	Link quality estimators that the proposed LQE models are compared to
1	Confusion matrix	[12], [16]	
2	Confusion matrix, accuracy, precision, recall, F1	[39]	
3	Classification accuracy, confusion matrix	[18], [38]	ETX [77], STLE [81], 4B [10], 4C [17]
4	Confusion matrix, recall, classification accuracy	[40]	SVC, extreme learning (EML), WNN [19]
5	Classification accuracy	[74]	FLI [14], LQI-PRR [82]
6	Classification accuracy, power estimation error	[6]	
7	Avg. delivery cost, classification accuracy	[17]	STLE [81], 4B [10]
8	RMSE, number of topology changes	[14]	4B [10]
9	(RMSE) Throughput estimation error	[70]	
10	(RMSE) PRR estimation error	[19]	SNR, back-propagation Neural Network, ARIMA, XCoPred
11	MAE	[68]	SVM, regression trees, k-nearest neighbor, Gaussian process for regression
12	MAE, computational load	[72]	Baseline routing performance, online perceptrons, online regression trees, fast incremental model trees vs. adaptive model rules
13	CDF, LQE stability	[15]	ETX [77], F-LQE [14]
14	Mean and stdev. of estimation error, CDF, R^2	[5]	
15	LQE sensitivity, LQE stability, CDF	[13], [75]	ETX [77], WMEWMA [8], RNP [78], 4B [10]
16	Number of downloads	[7]	
17	PRR, number of parent changes	[8]	
18	Total number of transmissions, average tree depth, delivery rate (PSR)	[10]	ETX [77], Collection Tree Protocol (CTP) [83], MultiHopLQI
19	PSR	[9]	ETX [77], RNP [78]
20	Throughput	[11]	
21	Channel rank estimation, energy consumption, channel switching delay, stability	[76]	
22	Average packet loss, num. of control packets, energy consumption	[69]	

F. Evaluation of the Proposed Models

We analyze the way Tables II and III evaluate the proposed LQE models along several dimensions. The evaluation metric analysis of the surveyed literature is presented in Table V. The second column of the table lists the metrics used to evaluate the LQE model by the research papers listed in the third column of the table. The fourth column of the table identifies what other existing link quality estimators were utilized and compared against the ones proposed in the papers outlined in the third column.

1) Evaluation From the Purpose of the LQE Perspective: Firstly, we analyze the evaluation of the models through the lens of the purpose of the LQE as discussed in Section II-B. We identify direct evaluation, where the paper directly quantifies the performance of the proposed LQE models vs. indirect evaluation, where the improvement of the protocol or the application as a result of the LQE metric is quantified.

Direct evaluations of LQE models typically evaluate the predicted or estimated value against a measured or simulated ground truth. The metrics used for evaluation depend on the output of the proposed model for LQE discussed in Section II-E.

When the output are categorical values, then it is possible to use metrics based on predicted label count versus the label count of the ground truth. Confusion matrices are used by [12], [16], [18], [38]–[40] as seen in rows 1, 2, 3 and 4 of Table V, classification accuracy is used by [6], [17], [18], [38], [40], [74] as observed in rows 3, 5, 6 and 7, and recall is used in combination with accuracy and confusion matrix by [40] as illustrated in the fourth row of the table. Only more recently, [39] uses the combined confusion matrix, precision, recall and F1 to provide more detailed insights into the performance of their classifier. Well known evaluation metrics, such as classification precision, classification sensitivity, F1 and Receiver Operating Characteristic (ROC) curve are used seldom or not at all among the evaluation metrics in the surveyed classification work. However, they can be computed for some of the metrics based on the provided confusion matrices.

The LQE metrics listed in rows 1-3 of Table V can be compared to each other in terms of performance by mapping the 5 and 7 class estimators to the 2 or 3 class estimator. This results in a number of comparable 2 or 3 dimension confusion matrices that can be analyzed. However, as the metrics are developed and evaluated under different datasets, the comparison would not be exactly fair and it would not be clear which design decision led one to be superior to another. The same discussion holds also for other rows of the table that share common evaluation metrics. High level comparisons that abstract such details are provided later in Sections III and IV for selected ML works that reported their results in sufficient detail.

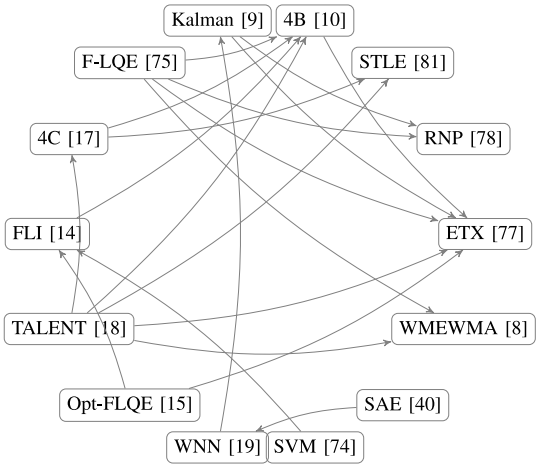


Fig. 6. Visualization of relationships for cross-comparison of the research papers with their corresponding evaluation metrics outlined in Table V.

When the output is continuous, then each predicted value is compared against each measured or simulated value using a distance metric. For instance, the authors of [14], [19], [70] use Root-Mean-Square Error (RMSE) as a distance metric as shown in rows 8-10 of Table V, whereas the authors of [68], [72] use mean absolute error (Mean Absolute Error (MAE)) as in rows 11 and 12 of the table. Some other research papers as in [5], [13], [15], [75] use Cumulative Distribution Function (CDF) as illustrated in rows 13-15, while the authors of [5] leverage R^2 in row 14 of Table V.

Indirect evaluations of LQE models evaluate against application specific metrics. The papers evaluate the performance of their objective functions based on the presence of link quality estimators. For example, the studies conducted in [5]–[8], [11], [12], [16], [69], [70], [72], [76] consider their respective objective functions for the particular applications and demonstrate to obtain better results by means of using estimators compared to the cases with the absence of estimators. While these research papers are likely to be leading on the respective use cases of LQE models owing to their first attempts in their specific application domains, their results and design decisions are still difficult to compare against each other. Various application specific evaluation metrics, such as number of downloads [7], number of parent changes [8], throughput [11] can also be found as listed in the rows 16-22 of Table V.

2) Evaluation From Cross-Comparison Perspective: Secondly, we categorize papers that *evaluate their outcomes against other estimators existing at the time of writing* versus papers that are somewhat *stand alone*. For instance, in row 3 of Table V, TALENT [38] is evaluated against ETX, STLE, 4B, WMEWMA and 4C. For more clarity, this is represented visually in Fig. 6 with directed arrows exiting from TALENT and entering the boxes of the respective metrics, which explicitly depicts the relationship between the last two columns of Table V. Such comparisons are informative as demonstrated by [75]. Among their findings, they showed that PRR, WMEWMA, and ETX, which are PRR-based link quality estimators, overestimate the link quality, while RNP and 4B underestimate the link quality. F-LQE performed better estimation than the other compared estimators.

However, metrics of the surveyed papers [6], [16], [69], [70], [76] are not evaluated against other existing estimators, due to their unique approach (application) and/or being among the first to tackle certain aspect of estimation. For instance, the authors of [76] evaluate the estimated ranking/classification of subset of wireless channels and the authors of [69] evaluate the impact of networking performance with estimator assisted routing algorithm against vanilla (m)RPL protocol, while the authors of [70] evaluate estimated and real throughput degradation when line of sight is blocked by an object. Besides, the authors of [16] evaluate data-driven bidirectional link properties, and [6] evaluates estimated vs. ground truth signal fading, which is influenced by ML algorithm's ability to classify geographical tiles.

3) Evaluation From Infrastructure Perspective: Thirdly, we categorize papers to those that perform evaluation and validation on real testbeds [5]–[10], [12]–[14], [17], [18], [38], [69], [70], [75] shown as in rows 1, 3, 6, 8, 9, 14-19, 22, those that perform evaluation in simulation such as [15], [69], [76] in rows 13, 21, 22, and the rest that perform only numerical evaluation. The papers in the first category, that perform evaluation and validation on testbeds, are better at presenting how the estimator will actually influence the network. The papers from second category performing simulation can provide good foundation for further examination and potential implementation. Finally, the papers in third category, that only do numerical evaluation, can unveil possible improvements through statistical relationships.

4) Evaluation From Convergence Perspective: Fourthly, during our analysis it has emerged that a number of papers reflect on and quantify the convergence of their model. For instance, in [11], they concluded that their model starts to converge at approximately 40,000 packets. In [9], the authors demonstrated that the link degradation could be detected even with a single received packet. The metric proposed in [12] required approximately 10 packets to provide the estimation in either a static or mobile scenario. In [17], they suggested that data gathered from 4-7 nodes for approximately 10 minutes should be sufficient to train their models offline. Although these papers indicate convergence rate/size, a community wide systematic investigation of LQE model convergence is missing.

At this point, we can conclude that research community in general have shown remarkable improvements, use cases, and skills toward better estimators. However, despite the aforementioned evaluation of proposed estimators, providing a completely fair comparison of LQE models is not feasible considering the diverse evaluation metrics outlined in Table V.

G. Reproducibility

Reproducibility of the results is recognized as being an important step in the scientific process [65]–[67] and is

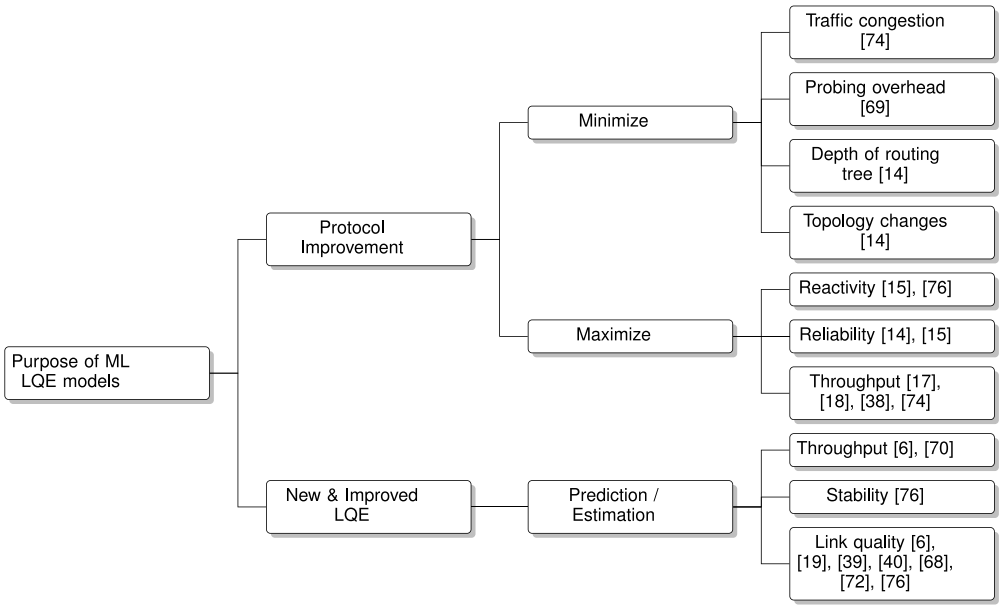


Fig. 7. Classification of the works by considering the purpose for which the ML LQE model was developed.

important for replication as well as for reporting explicit improvements over the baseline models. When researchers publicly share the data, simulation setups and their relevant codes it becomes easy for others to pick up, replicate and improve upon, thus speeding up the adoption and improvement. For instance, when a new LQE model is proposed, it can be ran on the same data or testbed as a set of existing models provided the data and models are publicly accessible to the community. The existing models can also be re-evaluated in the same setup, thus replicating the existing results or they can be used as baselines in new scenarios. With this approach, the performance of the new LQE model can be directly compared to the existing models with relatively low effort.

With respect to the reproducibility of the results in the surveyed publications, we notice that only [11], [18], [39] are easily reproducible because they rely on publicly available tracesets. Studies reported in [5], [7], [8], [10], [16], [17], [75] use open testbeds that, in principle, could be used to collect data and the results can be reproduced. However, it is not clear whether some of these testbeds are still operational given that 10-20 years have passed after the date of publication of the corresponding research. We were not able to find any evidence that the results in [9], [12], [14], [15], [19] could be reproduced as they strictly rely on an internal one-time deployment and data collection.

III. Application Perspective of ML-Based LQEs

In this section, we provide an analysis of the ML-based LQEs from application perspectives. We identify what is important from an application perspective and how that affects ML methods utilized for the LQE modeling. We first focus on the purpose of the LQE model development followed by the analyses of the application quality aspects.

A. LQE Design Purpose

In Section II-B, we have reflected on the purpose for which an LQE model was developed, and as depicted in Fig. 7, we found that about half of the ML-based LQE studies developed an estimator with the goal of improving an existing protocol, while the other half aimed for a new and superior LQE model. Fig. 7 presents that "protocol improvement" group attempts to minimize or maximize a particular objective, such as traffic congestion, probing overhead, topology changes, just to name a few. Most of the studies that fall into "new & improved LQE" group only aim to improve the prediction or estimation of the quality of a link.

The body of the work considering "protocol improvements" is intricate to quantitatively compare against each other since numerical details of the LQE models are not explicitly provided in the respective works, as previously discussed in Section II-F. Similar difficulties also arise for a large part of the body of work related to "new & improved LQE" models since they do not utilize consistent evaluation metrics. For instance, for LQE models formulated as a classification problem, only a subset of the works leverages accuracy as a metric, while other subsets use confusion matrix or specifically defined metrics, which indeed renders them impractical to quantitatively compare against each other, as outlined in Table V and discussed in Section II-F. Attaining a fair comparison is even more difficult for the works that formulate the LQE problem as a regression. Later in Section VI-C, we provide guidelines with regards to this aspect.

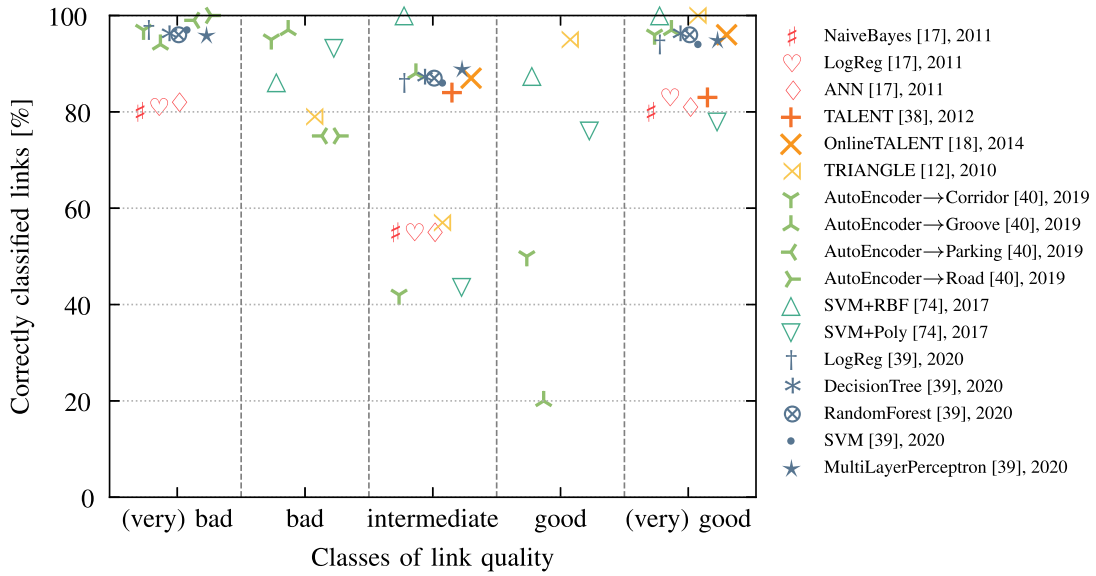


Fig. 8. Comparison of the wireless link quality classification performances throughout the surveyed papers.

Fig. 8 presents a high level comparison of the selected works that use ML for LQE model development [17], [18], [38]–[40] and one that does not [12]. All the considered works formulated the LQE model as a classification problem and it is possible to extract the approximated per class performance from the reported performance results of those respective works. Notice that they are different in terms of; i) the input features used to train and evaluate the models (more details in Section II-C), ii) the number of classes used for the model (more details in Section II-E), and iii) the considered ML algorithm (more details in Section II-D).

On the x-axis, Fig. 8 presents five different link quality classes, while on the y-axis it presents the percentage of correctly classified links. The comparison reveals that, autoencoder [40], which is a type of deep learning method, on average performs best with above 95% correctly classified *very bad*, *bad* and *very good* links and about 87% correctly classified *intermediate* quality link classes. Autoencoders are outperformed by the non-ML baseline [12] and SVM with RBF kernel [74] on the *very good* link quality class by about 4 percentage points, by over 30 percentage points on the *good* quality link class and by about 12 percentage points on the *intermediate* quality link class. As autoencoders are known to be powerful methods, we speculate that such high performance difference on those three classes might be due to insufficient training data or other experimental artifacts.

Tree-based methods and SVM [39] as well as the customized online learning algorithm TALENT [38] follow the performance of the autoencoders very closely with a tiny margin on *very bad*, *very good* and *intermediate* link quality classes. Next, the offline version of TALENT [18] exhibits very similar performance to tree-based methods and SVM on the *intermediate* class and about 17 percentage points worse on the *very good* class. Moreover, traditional artificial neural networks, logistic regression and Naive Bayes [17] follow next with almost 20 percentage points difference compared to autoencoders on the *very good* and *very bad* link quality classes and almost 30 percentage points on the *intermediate* link quality class. The relative performance difference of the work reported in [17] might be due to the poor data preprocessing practices, such as the lack of interpolation, which can significantly influence the final model performance that is discussed later in Section IV.

To summarize, the analysis of Fig. 8 reveals that autoencoders, tree based methods and SVM tend to consistently perform better than logistic regression, naive Bayes and ANNs while the non-ML TRIANGLE estimator performs very well on two of the classes, namely *very good* and *good* link quality classes.

Discussion: The observations from Fig. 8 also conform to the general performance intuitions regarding ML approaches. Namely, fuzzy logic and Naive Bayes are generally comparable with the latter being far more practical and popular. Neither of the two are known to exhibit better relative performance against logistic or linear regression. As shown in [17], [38], Naive Bayes tends to exhibit reduced performance compared to logistic regression, whereas ANNs are usually superior. Fuzzy logic, Naive Bayes, linear and logistic regression are relatively simple and require modest computational load and memory consumption. Therefore, these ML methods can be suitable for implementation in embedded devices, especially for small-dimensional feature spaces. Besides, ANNs can be designed to optimize computational load and memory consumption, particularly by simplifying their considered topologies, which in turn, comes with a cost to their performance.

For classification in constrained embedded devices, the authors of [17], [38] selected logistic regression for its simplicity among other three candidates. The selection was based on practical considerations, but their experiments proved that ANNs were superior compared to other LQE models. The reason behind this is because logistic and linear regressions are linear models that tend to be more suitable to approximate linear phenomena. Contrarily, LQE models do not follow linear models and therefore ANN-based model outperformed its counterpart LQE models in [17], [38].

SVMs, part of the so-called Kernel Methods, were popular and frequently used at the beginning of the century before significant breakthroughs brought by deep learning (deep neural networks (DNN)). SVMs often exhibit at least similar performance to ANNs and also to decision/regression trees [68]. However, there are only a paucity of contributions on adapting them for embedded devices [84]. In [72], the authors performed an in-depth comparison of ML algorithms including SVM, decision trees and k nearest neighbors (k-NN) from several perspectives, such as accuracy, computational load and training time. Their results showed that SVMs are constantly superior in terms of accuracy to k-NN and regression trees at the expense of significant resource consumption.

While many of the traditional ML methods including decision/regression trees and k-NN typically require an explicit, often manual feature engineering step, SVMs are able to automatically weight the features according to their importance automatizing part of the effort allocated for manual feature engineering. SVMs are known to be highly customizable through hyperparameter tuning, which is a dedicated research area within the ML community. Through appropriate selection of the kernel and parameter space [85], they are able to perform very well on both linear and non-linear problems. Therefore, from this particular perspective, SVMs and the broader Kernel Methods are indeed favorable choices for developing LQE models.

Deep learning, represented by DNNs are a new class of ML algorithms that are currently under intense investigation in various research communities penetrating also wireless and LQE [40]. These algorithms are very powerful and accurate for approximating both linear and non-linear problems, albeit requiring high memory and computational cost. Such models are prohibitive for embedding in constrained devices. However, there are a number of research efforts [86] invested in employing transfer learning approaches [87]. When an LQE based data processing occurs on a non-constrained device, such as the case in [6], DNNs can show an outstanding performance. While the authors of [6] proposed a novel and visionary approach for the development of an LQE model and accomplished robust results using SVMs, employing DNNs might assist in surpassing those existing results.

B. Application Quality Aspects

Following the analyses from Sections II-B and II-F, we have identified five important link quality aspects to consider when choosing or designing an LQE model (estimator). These aspects are often used to indirectly evaluate the performance of LQE models, by evaluating the behavior of the application that relies on LQE versus the one that does not rely on it.

1) *Reliability* - The LQE model should perform estimations that are as close as possible to the values observed. More explicitly, LQE models should maintain high accuracy.
2) *Adaptivity/Reactivity* - The LQE model should reach and adapt to persistent link quality changes. This indicates that when a link changes its quality for a longer period of time, the LQE model should be able to capture these changes and accordingly perform the estimations. Changes in estimation subsequently unveil routing topology changes.
3) *Stability* - The LQE model should be immune to transient link quality changes. This immunity ensures a relatively stable topology leading to reduced cost of routing overheads.
4) *Computational cost* - The computational complexity of LQE models should be considerate of the target devices, where computational load can be appropriately apportioned among constrained and powerful devices.
5) *Probing overhead* - LQE models consider a diverse set of metrics to estimate the link quality, as discussed in Section II-C, which are gathered through probing. LQE models should be designed in an optimal way so that the probing overhead is minimized.

A comprehensive classification of the ML-based LQE studies according to the aforementioned five application quality aspects is exhibited in Fig. 9, which reveals that most of the LQE studies explicitly consider *computational cost* and *reliability* aspects in their evaluations, whilst only a paucity of the studies considers *probing overhead*, *adaptability* and *stability*. With respect to *computational cost*, it can be readily observed from the figure that tree- and neural network-based methods tend to have higher computational cost, whereas online logistic regression has medium cost, and Naive Bayes, fuzzy logic and offline logistic regression have relatively low computational cost. With regards to the *probing overhead* for trace-set collection, it is perceived from Fig. 9 that some LQE models are designed to incur zero-overhead, and one incurs both asynchronous and synchronous (async. & sync.) probing, whereas the other is devised to use an adaptive probing rate. As far as *reliability* is concerned, some LQE studies focus on the reliability of the routing tree topology, and on the link prediction/estimation, whereas others put emphasis on the traffic. *Adaptability* is explicitly taken into consideration mostly in studies employing online learning algorithms, while *stability* is considered for those studies focusing on offline learning algorithms.

Discussion: To support a more in-depth understanding, Table VI presents an aggregated and elaborated view of the papers that are systematically categorized in Figs. 7 and 9. The first column of the table shows the purpose for which LQEs have been developed, the second column of the table lists the problem that is being solved using ML-based LQE models, the third provides the relevant research papers solving those respective problems, column four includes the ML type and method, while the last five columns correspond to

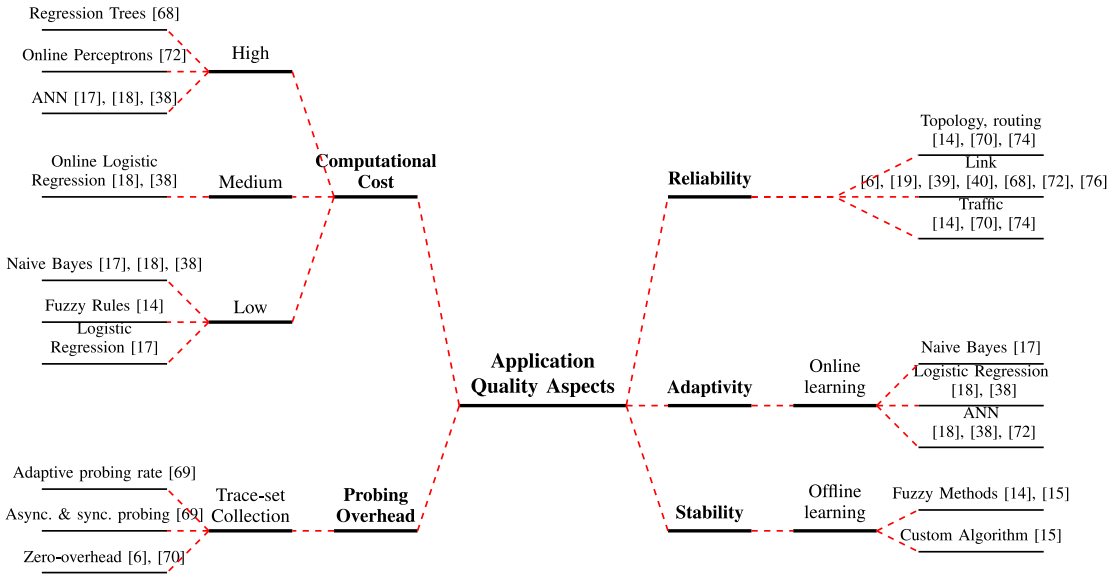


Fig. 9. Classification of the surveyed LQE papers by taking into consideration the identified application quality aspects.

the link quality metrics previously enumerated in this section. The last five columns are filled in, if those quality aspects are given consideration in these respective research papers and left empty otherwise.

The first line of Table VI indicates that the problem solved by [17], [18], [38] is to reduce the cost of packet delivery with a well-known multi-hop protocol, the so-called collection tree protocol (CTP). In their first approach, [17] achieve this by developing three batch ML models that, according to their evaluation, perform better than 4BIT. However, ML models are trained in batch mode and remain static after training, therefore the estimator is not adaptive to persistent changes in the link. Batch or offline training of ML algorithms [88] means that the model is trained, optimized and evaluated once on available training and testing sets, and has to be completely re-trained later in order to adapt the possible changes in the distribution of the updated data. In practice, this corresponds to sporadic updates, e.g., once in few hours and once per day depending on how the overall system is engineered. For the case of embedded devices, the device has to be fully or partially reprogrammed [89]. In the specific case of [17], it is clear that the coefficients of the linear regression model learned during training are hard-coded on the target device and reprogramming is required for obtaining the updates.

When the behavior of the links changes significantly, especially for wireless networks having mobility, the offline model is expected to decrease in performance, since those link changes may not be recognized by the ML model residing on the devices. In [18], [38], they improve their previously proposed offline modeling by introducing adaptivity to their models and thus developing online versions of the learning algorithms. Online ML algorithms are capable of updating their model [88] as new data points arrive during regular operation. The authors of [18], [38] also address reliability and computational cost aspects in their evaluation, as can be readily seen in the respective columns of Table VI.

Realizing the shortcomings of the offline-models [68] for estimating LQE in community networks and then developing on-line [72] models can be also noticed in the sixth line of Table VI. This research problem is formulated as a regression problem, while the previous one addressed in [17], [18], [38] is formulated as a classification one. Both approaches are suitable for the purpose and both need to implement a threshold- or class-based decision making on whether to use the link or not. ML methods used in [68] and [72] target WiFi devices (routers) and are thus more expensive in terms of memory and computational cost than those that target constrained devices (sensors), as outlined at the first line of Table VI. Generally speaking, ML algorithms, such as SVM and k-NN used in [68], [72] and outlined at line six of Table VI are computationally more expensive than naive Bayes and logistic regression utilized in [17], [18], [38] and outlined at the first line of Table VI.

In addition to the adaptivity trade-offs noticed in research papers at the first and sixth rows of Table VI, reactivity trade-offs can be perceived from research papers outlined in the second, third and seventh rows of Table VI. More explicitly, in the second row, LQE model is used to improve network reliability by reducing topology changes and the depth of the routing tree [14], while still maintaining high reliability, and in the third and seventh rows, [15] and [76] enhance reliability, stability and reactivity, respectively. The application requirements of these studies seem to favor reliable and cost effective routing with minimal routing topology changes. To sum up, the LQE model has to be as accurate as possible, update the model on significant link changes and remain immune to short-term

TABLE VI
OVERVIEW OF THE APPLICATIONS OF THE ML-BASED LQE MODELS FOR THE RELEVANT PAPERS SURVEYED IN TABLES II AND III

Purpose	Specific Problems	Research Papers	ML Type and Method	Reliability	Adaptivity	Stability	Computational Cost	Probing Overhead
LQE for protocol performance	1. Reduce the cost of delivering a packet in multihop networks (CTP protocol)	[17]	Classification: Naive Bayes, Logistic regression, Artificial neural networks		No (offline)		Low	
		[18], [38]		Yes	Yes (online)		Medium	
	2. Improve network reliability, reduce topology changes and routing depth	[14]	Regression: Fuzzy logic (2 inference rules, defuzzification)	Yes		Yes	Low	
	3. Improve reliability and reactivity in an application specific network	[15]	Classification: Custom algorithm based on fuzzy logic	Yes	Yes	Yes		
	4. Minimize the overhead caused by active probing operations	[69]	Regression: Reinforcement learning					Yes
	5. Select links that maximize the delivery rate and minimize traffic congestion for routing.	[74]	Classification: SVM	Yes				
New or improved LQE	6. Prediction the quality of link in community network (WiFi)	[68]	Regression: SVM, regression trees, k-nearest neighbor, Gaussian process for regression	Yes	No (offline)		High	
		[72]	Regression: perceptron, regression trees, incremental model trees with drift detection and adaptive model rules	Yes	Yes (online)		High	
	7. Link prediction quality, stability and reactivity	[76]	Classification: custom algorithm + 2 extensions		Yes	Yes		
	8. Reliable link quality estimation using probability-guaranteed estimation result	[19]	Regression: Wavelet Neural networks	Yes				
	9. Improved LQE	[40]	Classification: Deep learning (autoencoders)	Yes				
	10. No overhead throughput estimation in mmWaves using RGB imaging	[70]	Regression: Adaptive regularization of weight vectors	Yes				Yes
	11. Accurate estimation of LoRA transmissions using multispectral imaging	[6]	Classification: SVMs with Radial Basis Function (RBF) kernel	Yes				Yes
	12. On Designing a Machine Learning Based WirelessLink Quality Classifier	[39]	Classification: Logistic regression, decision trees, random forest, SVM, multi-layer perceptron	Yes				

variations for the sake of a stable topology. To achieve such goal, the right tuning of on-line learning algorithms that ensure a good stability vs adaptivity trade-offs has to be performed.

The computation of LQE models involves probing overhead to collect relevant metrics, as discussed in Section II-C and Table IV. Minimizing the probing overhead has also been a major concern for a number of research papers [6], [69], and [70], as it can be readily observed from rows four, ten and eleven of Table VI. In row four, probing overhead is reduced by using reinforcement learning to guide the probing process [69], while in [6] and [70], network related information obtained via probing is replaced with external non-networking sources based on imaging. Replacing the probing overhead with additional hardware components that involve learning from image data, image capturing and processing, consequently leads to increased computational complexity of the system.

The remaining research papers [19], [40], and [74] outlined at lines five, eight and nine of Table VI address the aspects of developing more accurate estimators against predetermined baseline models. Additionally, the LQE model proposed by [19] provides probability-guaranteed estimation using packet reception ratio for satisfying reliability requirements of the smart grid communication standards.

IV. DESIGN PROCESS PERSPECTIVE OF ML-BASED LQES

For the development of any ML model, the researchers have to follow some very precise steps that are well established in the community, defined in the Knowledge Discovery Process (KDP) [63], [90], namely data pre-processing, model building and model evaluation. The data pre-processing stage is known to be the most time-consuming process, tends to have a major influence on the final performance of the model and is applied on the training and evaluation data collected based on the input metrics discussed in Section II-C. This stage includes several steps, such as data cleaning and interpolation, feature selection and resampling. The model building and selection steps usually take a set of ML methods, train them using the available data and evaluate their results, as discussed in Section II-F.

Analyzing the existing works from the perspective of the design process is equally important and complements the analysis from the application perspective performed in Section III.

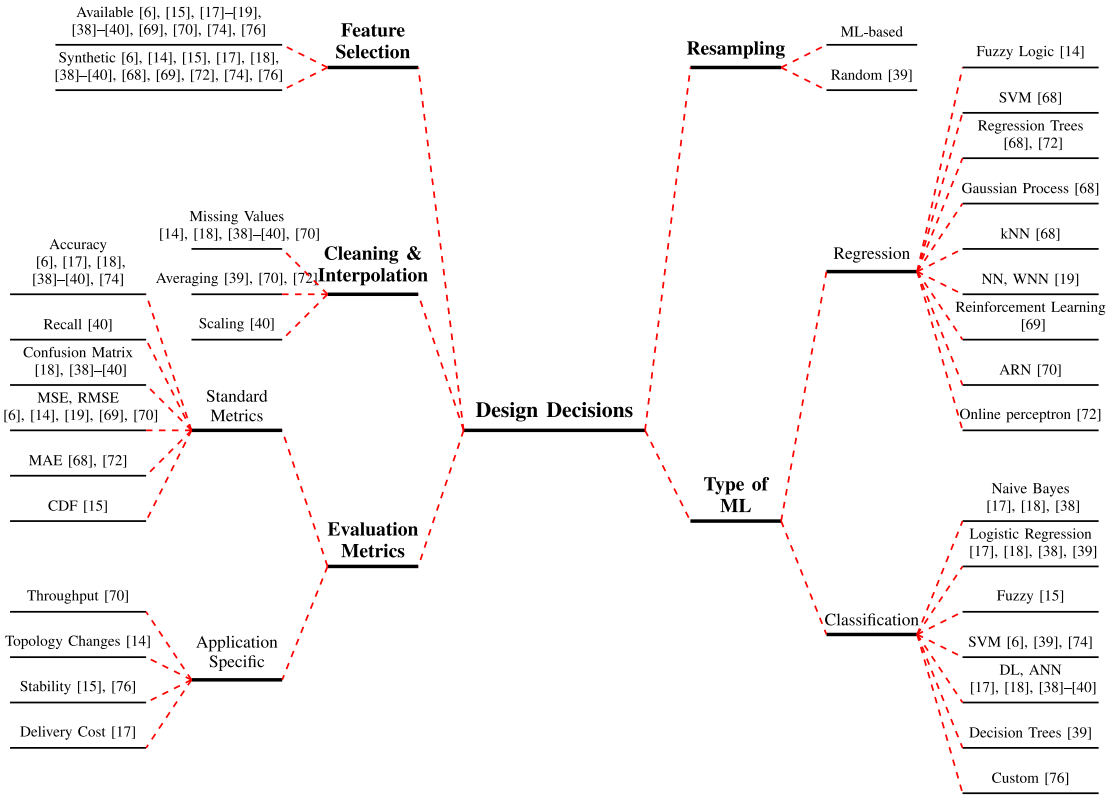


Fig. 10. Overview of the design decisions taken during the development of the ML-based LQE models for the relevant papers surveyed in Tables II and III.

Fig. 10 classifies the studies based on the reported design decisions taken while developing ML-based LQE models, namely cleaning and interpolation, feature selection, re-sampling strategy and ML model selection. Fig. 11 compares the reported influence of the respective steps on the final model considering accuracy as the metric while Fig. 12 depicts the trade-off for the process considering the F1 score[2] and the precision[3] and recall[4] metrics.

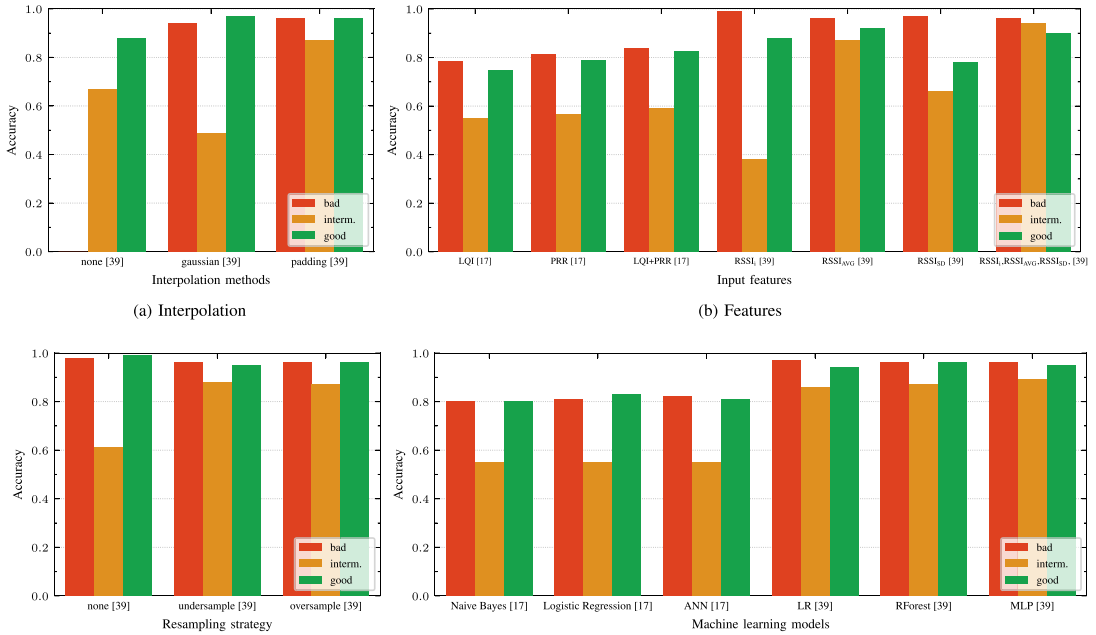
A. Cleaning & Interpolation Steps

From the Cleaning & Interpolation branch of the mind map depicted in Fig. 10 it can be seen that only seven of the ML-based LQE models provide explicit consideration of the cleaning and interpolation step. While in the general ML practice that use real world datasets, the cleaning step is very difficult to avoid and LQE-based research papers mostly leverage carefully collected datasets, often generated in-house from existing testbeds, as discussed in Section II-A. For instance, Okamoto *et al.* [70] perform cleaning on the image data they selected to use as part of the model training.

With respect to interpolation, however, several works [14], [18], [38], [40] fill in missing values with zeros. Their design decision with respect to this step of the process can also be referred to as interpolation using domain knowledge as they replace the missing RSSI values with 0, which represents a poor quality link with no received signal, yielding PRR equal to 0. It is not clear how [72] handle the missing data, however, they drop measurement data if there are not enough variations in their values.

Explicitly mentioning the design decision with respect to cleaning and interpolation is important for reproducibility (discussed in Section II-G) as well as for its potential influence on the final performance of the ML model. For instance, it can be readily seen from Fig. 11(a) that, all the other settings kept the same, domain knowledge interpolation denoted by "padding" can increase the accuracy of a classifier on *good* classes from 0.88 to 0.95, while also increasing the performance on the minority classes from 0.49 to 0.87 for *intermediate* and nearly 0 to 0.98 for *bad*, which can also be perceived from the findings of [39]. Going beyond accuracy as an evaluation metric, Fig. 12 shows significant performance increase, measured with F1 score which is the harmonic mean of the precision and recall, if the type of used interpolation is optimized for a particular scenario. More specifically, F1 score for no-interpolation is about 0.43 on the left lower part of the

[2] $F1 = 2 * precision * recall/(precision + recall)$.
[3] $precision = true\ positives/(true\ positives + false\ positives)$.
[4] $recall = true\ positives/(true\ positives + false\ negatives)$.



(a) Interpolation (b) Features

(c) Resampling (d) ML models

Fig. 11. Accuracy performance analyses for various steps of the design process as an exemplifying three-class LQE classification problem with unbalanced training data.

figure, then increases to 0.80 with Gaussian interpolation, and finally reaching 0.94 with constant interpolation (denoted by "padding" in Fig. 11(a)) that utilizes domain knowledge.

B. Feature Selection

According to the feature selection branch of the mind map depicted in Fig. 10, all research papers provide details on their feature selection. Often, all the features directly collected from testbed and simulator are used, as discussed in Section IV-B. Part of the literature, i.e., [17], [18] and [38] also considers the performance of the final model as a function of the input features as part of their analysis, while others only report a fixed set of features that are then used to develop and evaluate models. It may be because, the authors implicitly considered the feature selection step and solely reported the final features selected for their models to keep their paper concise. In such cases, the influence of other features or synthetic features [91] cannot be readily assessed in the related works surveyed.

Perceived from an extensive comparative evaluation in [39] and from another study that explicitly quantifies the impact of the feature selection on an LQE classification problem in [17], we summarize the reported performances with respect to the feature selection step in Fig. 11(b). While the works of the aforementioned figure leverage different datasets and distinct ML approaches, therefore they cannot be fairly benchmarked against each other, it is clear that the feature engineering can significantly increase the accuracy of a classifier within the same work by keeping all the other settings the same. Liu and Cerpa [17] reports up to 9 percentage points classification improvement in all classes by using LQI+PRR compared to the scenario using LQI only and PRR only, while Cerar [39] reports on average classification performance increases from 0.89 to 0.95, while also increasing the performance on the minority class from 0.38 to 0.87. Furthermore, according to Fig. 12, classification performance of F1 score ranges from 0.61 to 0.93, of precision ranges from 0.62 to 0.93 and of recall ranges from 0.63 to 0.93.

C. Resampling Strategy

Resampling is used in ML communities when the available input data is imbalanced [92], [93]. For instance, assume a classification problem where the aim is to classify links into *good*, *bad* and *intermediate* classes, similar to the problem approached in [16], [76]. If the *good* class would represent 75% of the examples in the training dataset, *bad* would represent 20% and *intermediate* would represent the remaining 5%, then a ML model would likely be well trained to recognize the *good* classes as it has been exposed to many such instances. However, it might be difficult for the model to recognize the other two classes, as they are scarcely populated instances in the dataset.

According to the resampling branch of the mind map in Fig. 10, only one very recent research papers elaborates on their resampling strategy. In other works it is often not clear whether they employed a resampling strategy in the case of

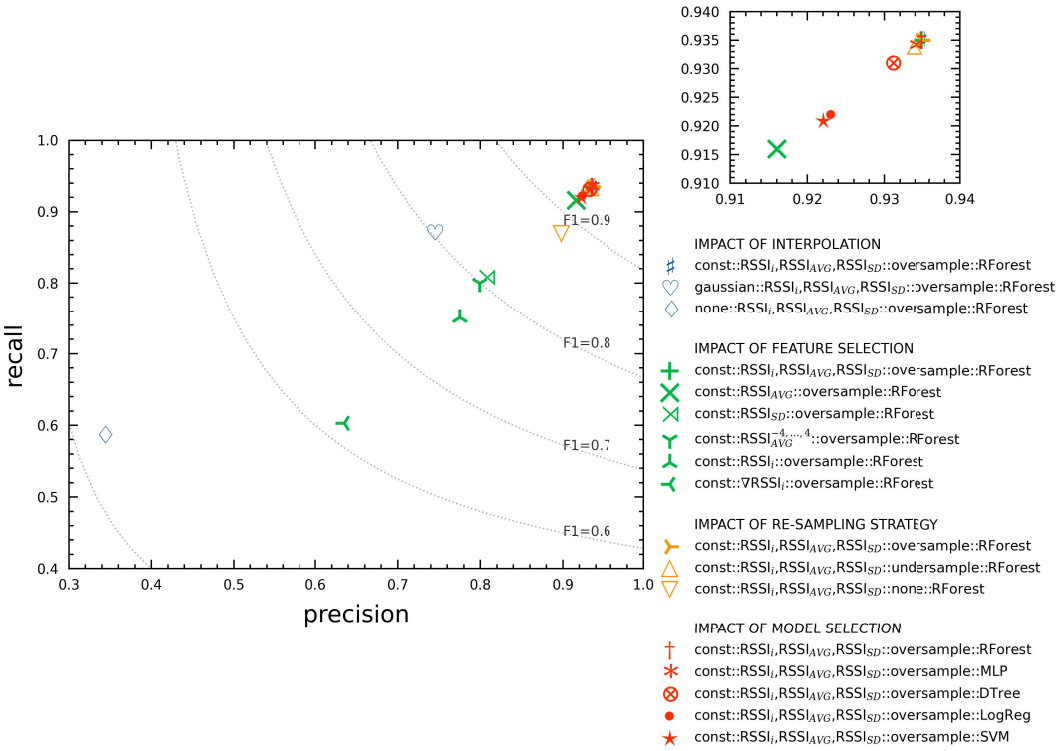


Fig. 12. Precision vs. recall performance trade-off for various design decisions including interpolation, feature selection, resampling and model selection, where the figure situated at the top-right corner is a zoomed-in portion of the closest region to F1=1 of the main figure.

imbalanced datasets. For instance, the performance of the predictor on two of the five classes is modest in [40]. It would be interesting to understand whether employing a resampling strategy would provide a better discrimination of the considered classes. Resampling could also improve other surveyed estimators in [6], [18], [38], [74].

From Fig. 11(c), it can be seen that, all the other settings being the same, performing resampling can slightly decrease the accuracy of a classifier on the two majority classes from 0.97 to 0.95, albeit it can yield a dramatic increase in the classification performance of the minority *intermediate* class with an accuracy raise from 0.61 to 0.88, which can also be worked out from the findings of [39]. Going beyond accuracy as an evaluation metric, Fig. 12 exhibits significant precision, recall and F1 score increase for the minority class, when a resampling strategy is leveraged. More specifically, an LQE model without resampling yields an F1 score of about 0.87, which then increases to about 0.93 with undersampling and remains at 0.93 when oversampling is considered.

D. Machine Learning Method

According to the ML method branch of the mind map shown in Fig. 10 seven of the works estimate the link quality in terms of discrete values, therefore they perform classification, while the remaining seven estimate it as actual values, hence regression is employed. The preferred ML method is chosen according to the specific application considered. It can be seen from this branch that the same type of algorithm can be adopted for classification and regression, respectively. For example, SVMs are exploited for regression in [68] and for classification in [6], [74]. Besides, every ML algorithm can be adapted to work in an on-line mode by means of retraining the model with every new incoming value during its operation. As discussed in Section III, online learning are particularly suitable for LQE models that also optimize the adaptivity in [18], [70], [72].

For classification, the most frequently used ML algorithms are naive Bayes, logistic regression, artificial neural networks (ANNs) and SVMs. The first three are used in [17], [18], [38], while SVMs are used in [6], [74]. The ML algorithms used for regression are more diverse ranging from fuzzy logic to reinforcement learning. While the performance of the classification algorithms is often evaluated according to the precision/recall and F1 scores in ML communities, potentially via complementary confusion matrices, the performance of regression are evaluated using distance metrics, such as RMSE and MAE.

TABLE VII
PUBLICLY AVAILABLE TRACE-SETS FOR THE ANALYSIS OF LQE

Origin of Trace-sets	HW. & Technology	Measurements	Data Points	Type	Additional Notes
MIT, Roofnet, [94], [95], 2002	Cisco Aironet 350, IEEE 802.11b, mesh, custom Roofnet protocol	Source, destination, sequence, time, signal, noise and so on	21 258 359 (1725 links, 4 bitrates)	1-to-N	Which packets were lost on a link is not provided.
Rutgers University, ORBIT testbed, [96], 2007	29x PC + Atheros 5212, IEEE 802.11abg	Seq. number, RSSI	611 632 (406 links, 300 packets/link, 1 packet/100 ms, 5 levels of noise)	1-to-N	Minor preprocessing is involved.
"Packet-metadata", [97], 2015	2x TelosB, IEEE 802.15.4	RSSI, LQI, noise floor, packet size, no. retries, energy, Tx power, ACK, queue size and so on	14 515 200 (300 packets per 80646 runs per 6 distances)	1-to-1	It requires minor preprocessing.
Colorado, [98], 2009	5x listeners, IEEE 802.11	Signal strength, data rate, channel, time-stamp and so on	29 000 (500 packets per 58 locations)	1-to-1	It requires preprocessing.
University of Michigan, [99], 2006	14x Mica2, proprietary protocol, sub-GHz ISM	RSSI	580 762 (1 packet/0.5s, 30 min/device, 3191 records/link)	1-to-N	MATLAB's binary format is considered and inconsistent data is observed (leading zeros and no units). Source and destination nodes are not clearly identified.
EVARILOS, UGent, [100], 2015	6 nodes, Bluetooth	RSSI, time-stamp	5 938 ($<$2 000 records/link)	N-to-1	Hospital environment is considered in the absence of interference.
EVARILOS, UGent, [100], 2015	5 nodes, IEEE 802.15.4	RSSI, time-of-arrival, time-stamp	110 126 ($<$35 000 records/link)	1-to-N	Hospital environment is considered in the absence of interference.
University of Colorado, [101], [102], 2009	6x PC with omni-directional antennas, 1x distinctly configured omni-directional antenna for transmitter, IEEE 802.11	Seq. number, coordinates, direction, TX power, 5x RSSI values per log	5x 623 207 (500 packets per 180 positions per 4 directions per 11 Tx levels per 5 nodes)	1-to-N	Experiment is composed of nodes equipped with antennas that are capable of serving 4 different directions. Tx power is variable and extensive documentation is available.
Brussels University, [103], 2007	19x Tmote Sky, IEEE 802.15.4	Seq. number, RSSI, LQI, time-stamp	112 793 ($<$1 600 packet/link)	1-to-N	It requires advanced preprocessing. Sequence numbers are rarely inconsistent. There are three more trace-sets available from this experiment that is intended for localization.

Fig. 11(d) shows that, all the other settings being the same, the selection of the ML method for a selected classification problem has a relatively smaller impact on the accuracy of a classifier compared to the other steps of the design process. As reported in both [17] and [39], the accuracy changes by up to 3 percentage points between the considered models. The zoomed portion of Fig. 12 exhibits the negligible impact of the model selection on the F1 score, which is up to around 0.02.

V. OVERVIEW OF MEASUREMENT DATA SOURCES

To complement the survey of the LQE models developed using data, we perform a survey of the publicly available trace-sets that have already been used or could be used for LQE. The data collected for a limited period of time on a given radio link, is referred to as *traces* in this section. When a set of these traces is recorded using more links and/or periods in several rounds of tests for a given testbed, we refer for it as a *trace-set*. Traces and trace-sets, in general, are prone to have irregularities and missing values that need to be preprocessed, especially when ported into ML algorithms. In this article, we refer to a trace-set that has been preprocessed as *dataset*. Ideally, a trace-set should include all the information available that is directly or indirectly related to the packets' trip.

To support our analysis, Tables VII and VIII summarize the publicly available trace-sets and the available features in each trace-set respectively. Our survey only analyzes publicly available trace-sets for LQE research that we were able to look into, however we mention other applicable trace-sets that are not publicly available. Table VII reviews the source of the trace-sets and the estimated year of creation along with the hardware and technology used for the trace-set gathering. Additionally, data that each trace relies on, the size of the trace/trace-set, the type of communication used in the measurement campaign, and additional notes on the specification and characteristic of the trace-sets can also be found in Table VII. Table VIII lists the trace-sets in the first column while the remaining columns refer to various metrics contained within the trace-set. This table maps the available metrics, also referred to as features, to the analyzed trace-sets.

To summarize the important points of these trace-sets, they were collected by the research teams at various universities worldwide using their own testbeds [94], [96], [100] or via conducting one-time deployments [97]–[99], [101], [103]. This confirms that the trace-sets were likely generated on testbeds developed and maintained in universities, which is consistent also with our findings in Section II-A. According to the second column of Table VII, four of the trace-sets are based on IEEE 802.11, three utilize IEEE 802.15.4, one is based on IEEE 802.15.1, and one operates on a proprietary radio technology. According to the fourth column of the table, the number of entries, i.e., data points, ranges from only 6 thousand up to 21 million, whereas the number of measured data

TABLE VIII
AVAILABLE FEATURES OF THE TRACE-SETS SURVEYED IN TABLE VII FOR THE SAKE OF LQE

Trace-set	Seq. Numbers	Time-stamp	RSSI	LQI	SNR (Signal/Noise)	Location	Queue (Size/Length)	Frame Size	HW. Specs.
Roofnet [94], [95]		✓(implicit)			✓/ ✓				✓
Rutgers [96]	✓	✓	✓		✗/ ✓	✓			✓
"Packet-metadata" [97]	✓	✓	✓	✓	✓/ ✓	✓	✓/ ✓	✓	✓
Colorado [98]	✓	✓	✓			✓		✓	✓
University of Michigan [99]	✓		✓						✓
EVARILOS [100]	✓	✓	✓			✓			✓
Colorado [101], [102]	✓	✓	✓			✓			✓
Brussels [103]	✓	✓	✓	✓		✓			

per entry ranges from one to about fifteen. The third column of the table lists the measurements available in each trace-set. For more clarity, the measurements are summarized in Table VIII for each trace-set and their meaning and importance for LQE is summarized as follows:

- A sequence number holds key information on the consecutive orders of the received packets and/or frames. With the aid of the sequence number, reconstruction of time series is enabled and thus it inherently provides information on packet loss and duplicated packets. It is already part of the frame headers owing to the standardization efforts. Sequence numbers can be processed to provide PRR and its counterpart PER that are useful input for LQE model.
- A time-stamp, which can be relative or absolute, is a suitable addition to the aforementioned sequence number. It reveals the amount of elapsed time between measurements. Therefore, it can help for deciding on whether a previous data point is still relevant and thus improving LQE in a dynamic environment. If a high precision timer and dedicated radio hardware are available, time-stamps can also empower localization.
- Measurement points indicating the quality of received signal on the links are mainly described by SNR, RSSI and LQI. SNR represents the ratio between the signal strength and the background noise strength. Compared to all other features, it allows the most clear-cut observation of the radio environment. However, some hardware, especially constrained devices, might not support direct SNR observation. In contrast to SNR, RSSI is the most widely-used measurement data and it can be accessed on the majority of radio hardware. It shows high correlation with SNR, since it is obtained in a similar way. Researchers may argue on its inaccuracy due to the low precision, i.e., quantization is around 3dB on most hardware. As opposed to the SNR and RSSI, LQI is a score-based measurement data and mostly found in radios of ZigBee-like (IEEE 802.15.4) technologies, which provides an indication of the quality of a communication channel for the transmission and the flawless reception of signals. However, the drawback of LQI is the lack of strict definition, leaning it to the vendor to decide its way of implementation and it may lead to the difficulty of cross-hardware comparison across vendors.
- For a more dynamic environment of wireless networks, where nodes are mainly mobile, information regarding the physical (geographical) locations can be beneficial.
- Additionally, there are other software related measurements data including queue size, queue length and frame length just to name few. If we refer to domain knowledge,[5] shorter frames tend to be more prone to errors, while queuing statistics can reveal information concerning buffer congestions.
- For the interpretation of the technical research outcome, revealing which hardwares were utilized during data collection is important to help diagnosing potential erratic behaviors of some hardware, including sensitivity degradation with time.

As can be seen from Table VIII, no single metrics appears in all trace-sets, however, sequence numbers, time stamps, RSSI, location and hardware specifications are available in the majority.

The Roofnet [94] is a well known WiFi-based trace-set built by MIT. It contains the largest number of data points among the trace-sets listed in Table VII. However, it is difficult to obtain the exact Roofnet setup/configuration used during the collection of the measurement data, since it has evolved with other contributions. One particular drawback of Roofnet is that PRR, as a potential LQE candidate, can only be computed as an aggregate value per link without the knowledge of how the link quality varied over time. Table VIII shows that this particular trace-set strictly depends on SNR values for the analysis of LQE.

The Rutgers trace-set [96] was gathered in the ORBIT testbed. It is large enough for ML models, requires only moderate preprocessing and is appropriately formed for data-driven LQE. It contains the overall packet loss of 36.5%. The meta-data contains information regarding physical positions, timestamps and hardware used. The trace-set for each node contains raw RSSI value along with the sequence number, as depicted in Table VIII. From the surveyed papers, [18] relies on both Rutgers and Colorado, while [11] considers only Rutgers.

The "packet-metadata" [97] comes with a plethora of features convenient for LQE research, as indicated in Table VIII. 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. One of the major characteristic of this trace-set is to enable the observation of packet queue. Packet

[5]Domain knowledge is the knowledge relating to the associated environment in which the target system performs, where the knowledge concerning the environment of a particular application plays a significant role in facilitating the process of learning in the context of ML algorithms.

loss can only be observed in rare cases with very small packet queue length.

Upon closer investigation for the remaining six trace-sets listed in Table VII, they are not primarily targeted for data-driven LQE research. The trace-set from the University of Michigan [99] 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 [100] are mainly well formated, whereas each contains fewer than 2,000 entries, and thus both are not well suited for data-driven LQE research. In Colorado trace-set [101], the diversity of the link performance is missing as all links seem to exhibit less than 1% packet loss. Finally, the trace-set of Brussels University [103], at the time of writing, is inadequate for data-driven LQE analysis, and suffers from an inconsistent data structure and deficient documentation.

After careful evaluation of the candidate trace-sets, we can conclude that the most suitable candidate for data-driven analysis of LQE is the Rutgers trace-set. Roughly speaking, all the other candidates lack sufficient size, are structured in improper format, contain negligible packet loss hindering from practical LQE investigation and/or rely on deficient documentation. However, these are the main characteristics required for ML-based LQE investigation, where it's classification primarily depends on PRR. Even though we concluded that the Rutgers trace-set is the most suitable one for data-driven LQE research, it also lacks some critical aspects for near-perfect data-driven LQE research including explicit time-stamps and non-artificial noise sources just to name a few. We take this conclusion in account later in Section VI-C where we suggest industry and research community a design guideline on how a good trace-set should be collected.

VI. Findings

In this section, we present our findings as a result of the comprehensive survey of data-driven LQE models, publicly available trace-sets and the design of ML-based LQE models. First, we elaborate on the lessons learned from the aforementioned survey of the literature, then we suggest design guidelines for developing ML-based LQE models based on application quality aspects and for generic trace-set collection to the industry and research community.

A. Lessons Learned

Having surveyed the comprehensive literature for LQE models using ML algorithms in Section II, we now outline the lessons we have learned throughout this section.

- While traditionally, most LQE models were developed to be eventually used by a routing protocol, recently researchers have also identified their potential application in single hop networks, particularly with the intention of reducing network planning costs via automation [6].
- Recently, new sources of information or input metrics, such as topological- and imaging-based are considered for the development of LQE models, as noted in Section II-C.
- From Sections II-D and III, it can be concluded that reinforcement learning is a relatively less popular ML method for LQE research.
- A number of LQE models provide categorization (grade) for link quality rather than continuous values. The analysis in Section II-E shows that the number of categories or classes (link quality grades) varies between 2 and 7.
- There is no standardized and easy way of evaluating and benchmarking LQE models against each other, as it is evident from the analysis in Section II-F.
- Only a small number of research papers provide all the details and datasets so that the results can be readily reproduced by the research community to improve upon and to be utilized as a baseline/benchmarking model for the sake of comparative analysis, as discussed in Section II-G.

We highlight the following lessons learned from the application perspective analysis of the ML-based LQE models performed in Section III.

- From the application that uses LQE, such as a multi-hop routing protocol, we were able to identify five application quality metrics that are indispensable for the development of an ML-based LQE model: reliability, adaptivity/reactivity, stability, computational cost and probing overhead. These application quality metrics are outlined and explained in Section III and distilled from the extensive survey in Section II. These metrics are sometimes used to evaluate the performance of the application with/without using LQE.
- Only a paucity of contributions explicitly considers adaptivity, stability, computational cost and probing overhead in their evaluation for the performance of an LQE model, as perceived from the analysis in Section III. No research paper considers all five aspects together.
- To develop LQE models for wireless networks with dynamic topology, adaptivity can be enabled with the aid of online learning algorithms. Important link changes are difficult to capture with offline models, resulting in a degradation of the performance of the LQE model, as the up-to-date link state is unknown to the intended devices.

The lessons learned from design decisions taken for developing existing ML-based LQE models as analyzed in Section IV can be summarized as follows.

- Training data for ML models often miss data points, for example no records for the lost packets can be found. The approach adopted for compensating the missing data, such as interpolation, may have significant impact on the final performance of the LQE model and explicitly describing the process is important for enabling reproducibility.
- The feature sets that are utilized for LQE research are not always explicitly reported nor identical among different LQE models, which hinders fair comparative analysis for diverse parameter settings.
- Training data for ML models can be highly imbalanced. Classification-wise, for example, the training dataset can be dominated by one type of link quality class (grade), which consequently leads to a highly biased LQE model

that is unable to recognize minority classes. To counter this artifact, resampling has to be employed for highly imbalanced datasets. No research papers explicitly state their resampling strategy, as readily observed in Fig. 10 of Section IV-C.
- Logistic and linear regressions are linear models that tend to be more suitable to approximate linear phenomena. In practical scenarios, LQE models do not obey linearity and therefore ANN-based models outperform linear models. However, ANN- and DNN-based models usually require high memory and computational resources, which is unfavorable for constrained devices, albeit they may be tuned to necessitate less resources but at the expense of proportional performance.

From the overview of measurement data sources in Section V, we have learned the following lessons.
- Only a limited number of publicly available datasets record overlapping/identical metrics, which can indeed empower fair comparative analyses between diverse LQE models.
- Measurement points indicating the quality of the received signal on links are commonly defined by SNR, RSSI and LQI.

B. Design Guidelines for ML-Based LQE Model

Due to a very large decision space for developing a ML-based LQE model, it can be challenging to provide a universal decision diagram or methodology. However, showing how application requirements affect design decisions, and by reflexivity, how certain design decisions can favor some application requirements can be invaluable for the development of ML-based LQE models. In this section, we provide design guidelines on developing a ML-based LQE model starting from the five application quality aspects identified in Section III and their implications on decisions during the design steps of the ML process discussed in Section IV. The visual relationship of how the application quality aspects influence the design decisions for developing LQE models is illustrated in Fig. 13.

1) Reliability: When *reliability* is *the only* application quality aspect to be optimized for developing a ML-based LQE model, trace-set collection, data pre-processing and ML method selection should be carefully considered, as depicted in the Reliability branch of the mind map in Fig. 13.

Trace-set collection: The trace-set collection and subsequent probing mechanism utilized during the actual operation of an LQE model, can collect all the input metrics listed in Table VIII and perhaps even other inventive metrics that have not been used up-to-date in the existing literature.

Data pre-processing: During data pre-processing, high dimensional feature vectors using recorded input metrics as well as synthetically generated ones (see Section IV-B) can be used as there are no constraints on the memory use or computational power of the machine used to train the subsequent model.

ML method selection: During ML method selection, more computationally expensive methods, such as DNN, SVMs with non-linear kernel as well as ensemble methods, such as random forests can be considered. For accurate models that provide very good *reliability*, these methods are able to train on high dimensional feature vectors. However, they will also require many training data-points, possibly hours or days of measurements. While DNNs are known to be very powerful, they are also excessively data hungry. Their performance can be significantly diminished if the data-points are not sufficient.

2) Adaptivity: When *adaptivity* is *the only* application quality aspect to be optimized for developing an ML-based LQE model, data pre-processing and ML method selection are the two aspects to be examined, as illustrated in the Adaptivity branch of the mind map in Fig. 13.

Data pre-processing: Adaptivity requires LQE model to capture non-transient link fluctuations, therefore it has to monitor temporal aspects of the link. This is usually realized by introducing time windows on which the pre-processing is done. As opposed to pre-processing all available data in a bulk mode for subsequent offline development as employed for *reliability* aspect, each window is pre-processed separately for the *adaptivity*. The size of the window then influences the *adaptivity* of the model, where a smaller window size yields a more adaptive model.

ML method selection: During the ML method selection, online versions of ML methods or reinforcement learning are more suitable for capturing the changes in time. Generally, the online version of an offline ML method may be slightly more expensive computationally and its performance may be slightly reduced. Reinforcement learning is a class of ML algorithms that learn from experience and these are inherently designed to adapt to changes. The higher the required *adaptivity*, the faster the model has to change, leading to a more reactive ML (method) parameter tuning.

3) Stability: When *stability* is *the only* application quality aspect to be optimized for developing an ML-based LQE model, the same ML design steps are affected as outlined in the *Adaptivity* aspect, namely data pre-processing and ML method selection, as portrayed in the Stability branch of the mind map in Fig. 13. However, they are reversely affected when compared to the *adaptivity* aspect.

Data pre-processing: Stability requires LQE model to be immune to transient link behavior. While it may assume changes over time, it encourages only relevant changes. The size of the window chosen in this case typically represents a compromise between the batch approach mentioned for *reliability* and the relatively small reactive window that maximizes *adaptivity*.

ML method selection: During the ML method selection, online versions of ML methods or reinforcement learning are more suitable for capturing changes in time, however, they need to be optimized to detect persistent link changes, while being immune to transient ones.

4) Computational Cost: When *computational cost* is *the only* application quality aspect to be optimized for developing an ML-based LQE model, data pre-processing and ML method selection should be carefully contemplated, as outlined in the Computational Cost branch of the mind map in Fig. 13.

Data pre-processing: Computational cost optimization requires reducing memory and energy consumption as well as processor performance aspects required for the

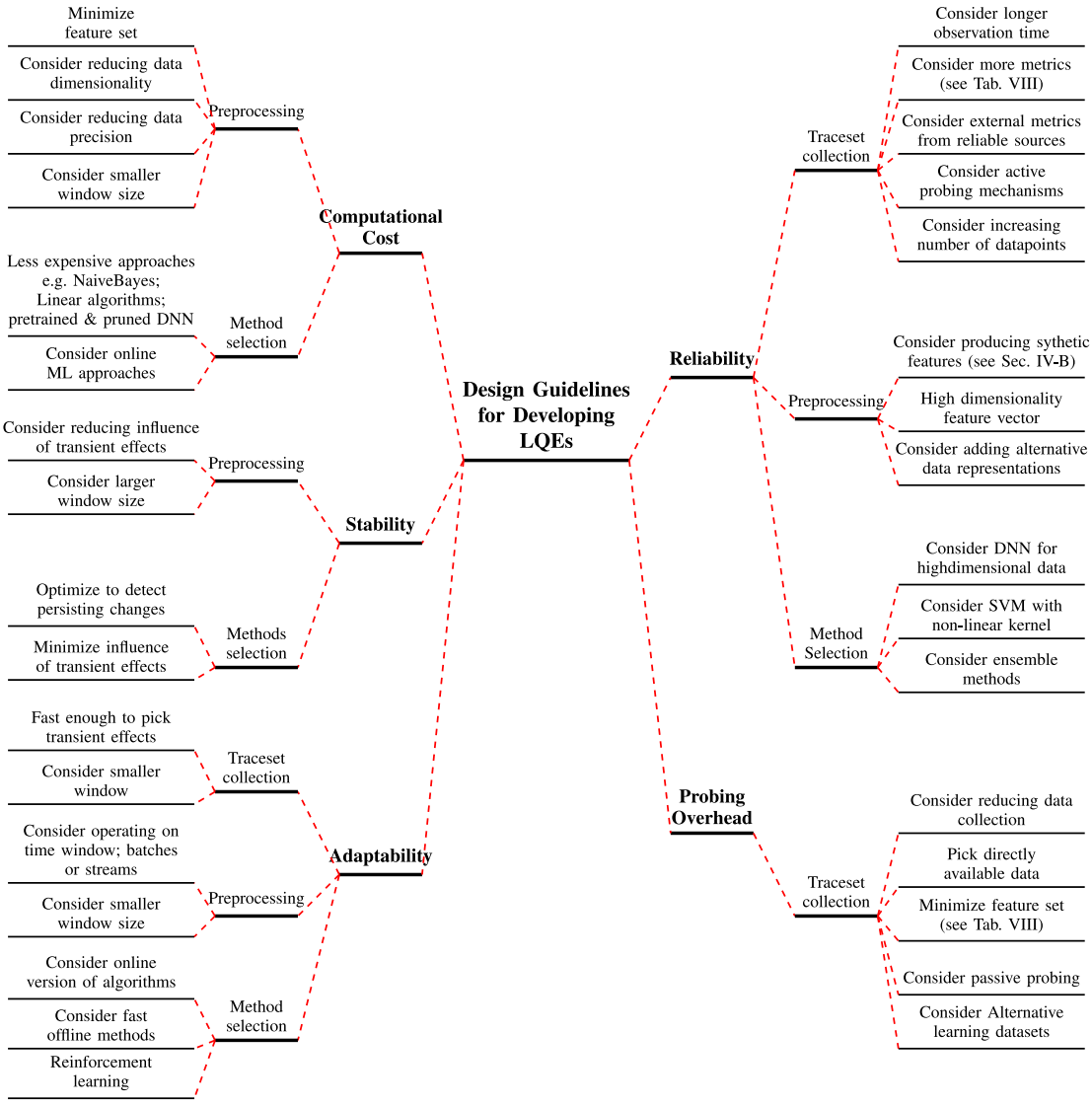


Fig. 13. Mind map representation of design guidelines for LQE model development.

LQE model development. For offline or batch processing, the size of the feature vectors should be kept to a minimum, therefore it has to include only the most relevant real or synthetic features. Alternatively, projecting large feature vectors to a lower dimensional space might help for training. Additionally, for online processing, smaller time windows that minimize RAM consumption are favored.

ML method selection: During ML method selection, less intensive methods, such as naive Bayes or linear/logistic regression are preferred. When online versions of the ML methods are utilized, their configurations should be appropriately adjusted so that the resource usage is kept at minimum. For instance, transfer learning [87] approaches enable stripped down versions of a complete model that was previously learned on a powerful machine, which is then deployed to the production environment. Transfer learning is becoming a relatively popular way of deploying DNN-based models on flying drones for instance [87].

5) Probing Overhead: When *Probing overhead* is *the only* application quality aspect to be optimized for developing an ML-based LQE model, trace-set collection is the only design process that requires careful attention, as illustrated in the probing overhead branch of Fig. 13.

Trace-set collection: Trace-set collection and subsequent probing mechanism utilized during actual operation of the

LQE model should only collect few and most important metrics from the ones listed in Table VIII. Ideally, LQE model can be engineered to work on passive probing so that it can only use the metrics that the transmitter captures.

6) Practical Scenarios: A practical application using LQE will likely request optimizing more than one of the five identified application quality aspects. As a result, the guideline and its illustrations for such cases would be more sophisticated and interconnected than in Fig. 13. However, the proposed guideline provides an overview of the measures to be taken and presents an invaluable trade-off between these application quality aspects that require careful attention for the development of an ML-based LQE model.

For example, when the application requires high *reliability* and *adaptivity*, large feature spaces can be used with powerful online algorithms on appropriately identified time windows. However, if *computational cost* is appended to the requirements, the feature space should be limited and the algorithm parameters should be optimized. If the LQE model is still computationally expensive, transfer learning or other out-of-the-box ML methods should be employed. When *probing overhead* is also appended to the previously-mentioned application quality aspects, then the feature set should only include locally available data (passive probing) and limited number of metrics (possibly none) involving active probing, as discussed in Section II-C. In brief, this guideline can be used as a reference for the development of an ML-based LQE model depending on the combination or quality aspects relevant for the application.

C. Design Guidelines for Trace-Set Collection

We now attempt to provide a generic guideline on how to design and collect an LQE trace-set, as portrayed in Fig. 14. It is worth noting that this design guideline comprises of plausible and reasonable observations gleaned from this survey of LQE and trace-sets, and from the analysis of ML methods reviewed for the sake of LQE models. Our plausible recommendations on how to design and collect an LQE trace-set can be summarized as follows, which can also be followed as in Fig. 14.

1) Core Components of a Trace-Set: Deciding on the data collection strategy, the application and the environment is a crucial stage, since the development of an LQE model is strictly dependent on the trace-set environment including industrial, outdoor, indoor and "clean" laboratory environments. State of the radio spectrum and interference level are important metrics to be taken into account before collecting a trace-set. For example, for an LQE model to work efficiently in a particular environment that is exposed to interference, then the LQE model has to be developed and trained over this kind of trace-set. More explicitly, one cannot expect an ML-based LQE model to perform well in an interference-exposed environment without having it implemented and tested on a trace-set containing interference measurement data, which leads us to data collection strategy and the application.

2) Availability and Documentation: Making trace-set publicly available is also another important stage, which can indeed empower better cross-testbed comparisons and provide good support/foundation from research community to conduct and disseminate research on LQE models. There are numerous ways to make trace-sets publicly available. One well known repository for wireless trace-sets is CRAWDAD,[6] although researchers can also take advantage of other methods like public version control systems, e.g., GitHub, GitLab and BitBucket just to name a few. Moreover, a systematic description on how the trace-set was collected is also required for research community to understand, test and improve upon. This will indeed help in capacity building between research groups.

3) Essential Measurements Data: Plausible logic dictates that a generic trace-set that can be utilized for any kind of LQE research is infeasible considering numerous features induced by the wireless communication parameters. By interpreting our overall observations gleaned from this survey paper, some of the most important measurements data or features that are recommended for an effective LQE research are already included in the design guideline of Fig. 14 with a notice that other application-dependent features may be required for a strong analysis of the LQE model. The elaborated details of these essential measurements data can be found in Section V.

There may be other application-dependent metrics and features (measurements data) related to the set of parameters of wireless communication that could be taken into account for a healthy investigation of a particular LQE model. We observe from the outcomes of this survey paper that each application can have unique characteristics and requirements for maintaining reliability, for satisfying a certain QoS and more generally for accomplishing a target objective, such as in smart grid, wireless sensor network, mobile cellular communication, air-to-air communication, air-to-ground communication, traditional terrestrial communication, underwater communication and other wirelessly communicating networks. Explicitly, for each application of these networks, determining a suitable evaluation metric is vitally important for the sake of maintaining a reliable and adequate communication. Therefore, trace-sets have to be designed and collected based on not only applications but also on evaluation metrics considering diverse environments, settings and technologies in order to be able to derive the properly effective metrics for an efficient development of the link quality estimation models.

Nonetheless, from the perspective of innovative data sources, a trace-set can be built without on-site measurements and before embarking on hardware deployments in order to provide a good estimate for the link quality for the sake of maintaining reliable communications. To achieve such goal, Demetri *et al.* [6] exploited readily available multi-spectral images from remote sensing, which are then utilized to quantify the attenuation of the deployment environment based on the classification of landscape characteristics. This particular research demonstrates that the quantification and classification of links can be conducted via solely relying on the image-based data source rather than the traditional on-site measurements data.

[6] A repository for archiving wireless data at Dartmouth: https://crawdad.org.

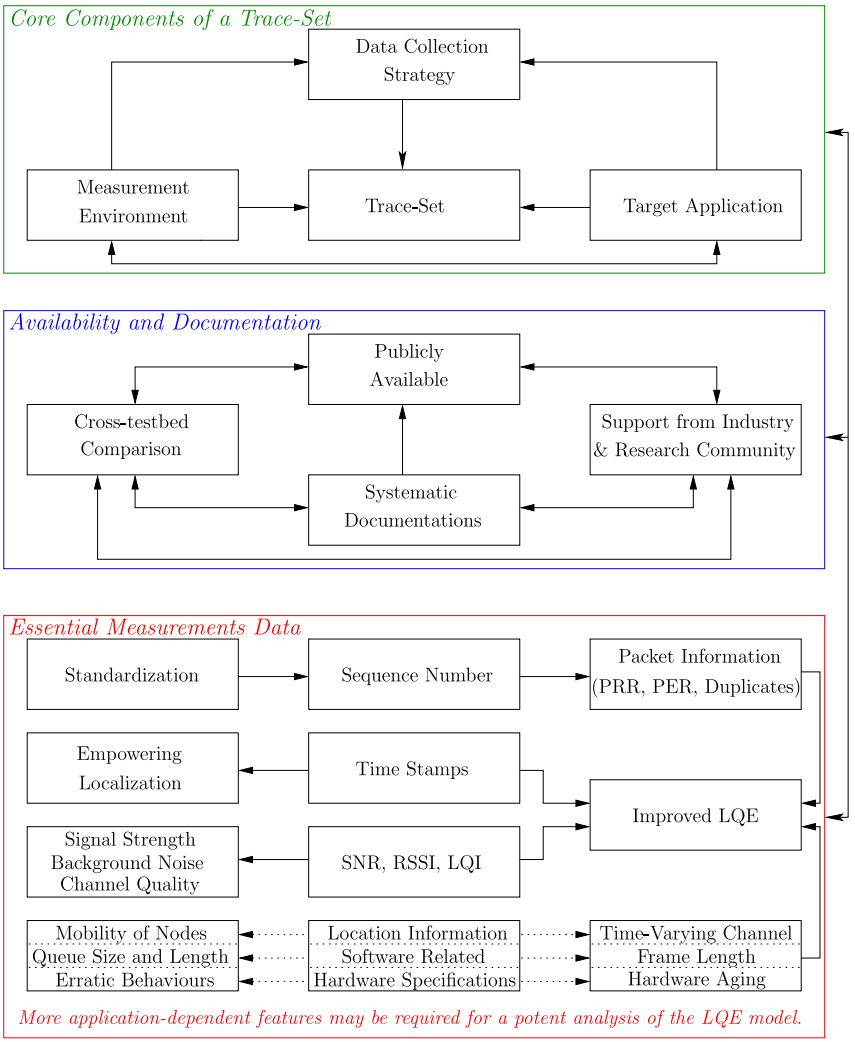


Fig. 14. Design guidelines recommended for the industry and research community to follow in order to design and collect trace-sets for the sake of LQE research.

For urban area applications, the aforementioned technique can also be leveraged for maintaining up to a certain degree of the link quality, but only considering the stationarity of the deployment environment. This is mainly because the spectral images obtained via remote sensing represent a stationary instance of the landscape and thus this technique would dramatically fail, since the LQE model developed using remote sensing would not be able to cope with the high mobility in such a scenario with moving vehicles, slowly-fading pedestrian channels, mobile UAVs and so on.

Besides, 3D model of large buildings can also be leveraged for the optimal indoor deployment of access points and wireless devices in order to supply with the adequate connectivity and coverage. The trace-set built from this indoor deployment can be utilized for other large and similar indoor buildings along with an indoor-generic LQE model to understand the characteristics of indoor links and to provide high quality link performance. Similarly, the same strategy can be implemented for a particular city to understand the link behavior in different weather conditions. One study for such scenario is conducted using high frequency [104], [105], where the impact of rainfall on wireless links was researched. They utilized rain gauges and their models are demonstrated to contain large bias, and rainfall predictions were underestimated, which indicates that a long-lasting and realistic measurement conditions are required along with a plethora of measurements data before developing a healthy LQE model.

Finally, recording hardware related metrics on a trace-set could also help in diagnosing potential problems during the model development. This would indeed require commercial radio chips that are capable of reporting the chip errors or chip related issues in order to pinpoint problems that may be encountered at the time of measurements data collection [106].

VII. Summary

Having outlined the lessons learned along with a comprehensive design guideline derived for ML-based LQE model development and trace-set collection, we now provide our concluding remarks and future research directions along with challenging open problems.

A. Conclusion

The data-driven approaches have been long ago adopted in the study of LQE. However, with the adoption of ML algorithms, it has recently gained new momentum stimulating for a broader and deeper understanding of the impact of communication parameters on the overall link quality. In this treatise, we first provide an in-depth survey of the existing literature on LQE models built from data traces, which reveals the expanding use of ML algorithms. We then analyze ML-based LQE models using performance data with the perspective of application requirements as well as with the ML-based design process that is commonly utilized in the ML research community. We complement our survey with the review of publicly available datasets relevant for LQE research. The findings from the analyses are summarized and design guidelines are provided to further consolidate this area of research.

B. Future Research Directions

Finally, we conclude the paper with a discussion on the open challenges, followed by several directions for future research, regarding (i) data sources utilized for developing LQE models, (ii) applicability of LQE models to heterogeneous networks incorporating multi-technology nodes, and (iii) a broader and deeper understanding of the link quality in various environments.

It is highly likely that commercial markets will leverage either pre-built LQE models for a particular application or entire training data to develop models from scratch. The potential opportunity of "model stores" and "dataset stores" can follow a similar way to conventional application stores/markets, distributing models for diverse applications. The competition will gradually become ripe as time elapsed. However, data-driven models are still in their infancy and several critical open challenges await concerning LQE models, which are outlined as follows.

1) A significant challenge is to directly compare different wireless link quality estimators. As discussed in Section II-F, there is no standardized approach to evaluate the performance of the estimators, and only a very small subset of estimators are compared directly in existing works. Establishing a uniform way of benchmarking new LQE models against existing ones using standard datasets and standard ML evaluation metrics, such as practiced in various ML communities, would greatly contribute to the ability to reproduce and compare innovative ML-based LQE models.
2) The performance of the existing LQE models using classifiers are solely evaluated based on the *accuracy* metric, possibly in addition to another application-specific metric, as discussed in Section II-F. However, it is well-known in the ML communities that *accuracy* is a misleading performance evaluation metric, especially for imbalanced datasets [107]. Adopting standardized metrics for classification, e.g., *precision*, *recall*, $F1$ and, where necessary, the detailed *confusion matrix* would lead to a more in-depth understanding of the actual performance and behavior of the LQE models for all the target classes. The same challenge applies to LQE models solving a regression problem.
3) Another challenge is to encourage researchers and industry to share trace-sets collected from real networks. More suitable public trace-sets would allow algorithms and machine learning models to be properly evaluated across different networks and scenarios considering the important metrics discussed in Section V. Indeed, trace-sets collected in an industrial environment could better represent a realistic communication network potentially with a broad number of parameters.
4) The other challenge is to go beyond one-to-one trace-sets. Research community is required to extend the scope to a more realistic measurement setup, e.g., considering multi-hop, non-static networks representing several wireless technologies. Such instances of trace-sets are scarce due to the necessity of exhausting efforts to monitor and record a packet's travel through a particular communication network.
5) Another challenge is that certain types of trace-sets are very expensive and time-consuming to gather. One way to overcome this is to conduct a synthesis of artificial data using generative adversarial neural networks as pointed out in [108]. Roughly speaking, this open challenge is a formidable task, since conducting such synthesis could potentially introduce unwanted bias to existing data, even though for specific applications a number of suitable examples of this method can be found in the literature, such as wireless channel modeling [109], [110].
6) The traditional approach to measure interference is mainly conducted through SNR or RSSI measurement data, which strictly relies on the data collection at certain intervals, and communication established from other nodes is mainly treated as a background noise for the sake of simplicity. The aim of interference measurement as part of this challenge is to develop LQE models that are aware of the on-going communication within a heterogeneous communication environment. None of the trace-set layouts surveyed in Section V is designed for such asynchronous information. Therefore, research community and industry have to pay attention to collecting such realistic trace-sets in order to be able to develop robust, agile and flexible LQE models that can

readily adapt in dynamic and realistic communication environments.

7) The wireless link abstraction comprised of channel, physical layer and link layer represents a complex system affected by a multitude of parameters, but most of the LQE datasets and research only leverages a small number of observed parameters. While recently additional image-based and topological-based contextual information has been incorporated in LQE models, it would be necessary in future large scale multi-parameter measurement campaigns to also capture the type of antenna, modulation and coding utilized, producer of the transceiver, firmware versions, to name a few. Such efforts would lead to a more in-depth understanding of the real-world operational networks and potential use of the findings to make well-informed decisions for the design of next-generation wireless systems, even beyond ML-based LQE model development.

In order to realize beyond simple decision making, i.e., channel and radio behavior modeling, *hand-tuning* of communication parameters within transceivers must be avoided. It is anticipated that the transceivers' internal components will be gradually replaced by software-based counterparts. Therefore, an inevitable incorporation of software-defined radio (SDR), FPGAs and link quality estimators is expected for intelligently handling parameters and operations through self-contained smart components. These joint LQE models can be designed in a similar manner to [111], particularly for heterogeneous networks involving the 5G and beyond communications.

The recent advancements in data-driven approaches in the form of machine learning and deep learning have already proven to be successful for the applications of communication networks. For example, attempts to use neural network-based autoencoders for channel decoding provide promising solutions [112], which can also be adopted for data-driven LQE investigation as it is discussed in [40].

The performance of link quality estimator is constrained by the dynamic network topology and one can keep track of the network topology changes considering replay-buffer-based deep Q-learning algorithm developed in [113], where authors control the position of UAVs, acting as relays, to compensate for the deteriorated communication links.

Additionally, LQE models involved in the optimization problems may become very large in size, and thus algorithms that can reduce complexity have to be developed to tackle with the scale of the problem. For example, a similar deep learning approach to [114] can be adopted for improving the performance of the proposed LQE model by means of eliminating the links from optimization problem that are not utilized for transmission.

Referring back to Section II-F, we discussed the convergence rate of LQE models. While some contributions [9], [11], [12], [17] focus their attention on the convergence of their LQE model, majority of the papers tend to neglect it. Motivated by this premise, we suggest the research community to pay particular attention on the LQE model convergence in order to prove the validity of their proposed models.

In addition to finding other new sources of data, a challenging task would be to analyze a large set of measurements in various environments and settings, from a large number of manufacturers to understand how measurements vary across different technologies and differ for various implementations within the same technology, and derive truly effective metrics for an efficient development of the link quality estimation model.

ACRONYMS

4B	Four-Bit
4C	Foresee
AI	Artificial Intelligence
BER	Bit Error Rate
CDF	Cumulative Distribution Function
ETX	Expected Transmission count
FLI	Fuzzy-logic Link Indicator
F-LQE	Fuzzy-logic based LQE
KDD	Knowledge Discovery and Data mining
KDP	Knowledge Discovery Process
LQ	Link Quality
LQE	Link Quality Estimation
LQI	Link Quality Indicator
MAE	Mean Absolute Error
ML	Machine Learning
MSE	Mean Squared Error
NLQ	Neighbor Link Quality
PER	Packet Error Rate
PRR	Packet Reception Ratio
PSR	Packet Success Ratio
RMSE	Root-Mean-Square Error
RNP	Required Number of Packets
ROC	Receiver Operating Characteristic
RSS	Received Signal Strength
RSSI	Received Signal Strength Indicator
SGD	Stochastic Gradient Descent
SNR	Signal-to-Noise Ratio
SVM	Support Vector Machine
TCP	Transmission Control Protocol
WMEWMA	Window Mean with an Exponentially Weighted Moving Average
WNN-LQE	Wavelet Neural Network based LQE.

ACKNOWLEDGMENT

The authors would like to thank Timotej Gale and Matjaž Depolli for their valuable insights.

REFERENCES

[1] H. Bai and M. Atiquzzaman, "Error modeling schemes for fading channels in wireless communications: A survey," *IEEE Commun. Surveys Tuts.*, vol. 5, no. 2, pp. 2–9, 4th Quart., 2003.

[2] N. Baccour *et al.*, "Radio link quality estimation in wireless sensor networks: A survey," *ACM Trans. Sens. Netw.*, vol. 8, no. 4, p. 34, Sep. 2012.

[3] A. Zanella, "Best practice in RSS measurements and ranging," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 4, pp. 2662–2686, 4th Quart., 2016.

[4] H. Yetgin, K. T. K. Cheung, M. El-Hajjar, and L. H. Hanzo, "A survey of network lifetime maximization techniques in wireless sensor networks," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 2, pp. 828–854, 2nd Quart., 2017.

[5] G. T. Nguyen, R. H. Katz, B. Noble, and M. Satyanarayanan, "A trace-based approach for modeling wireless channel behavior," in *Proc. Winter Simulat. Conf.*, Coronado, California, USA, Dec. 1996, pp. 597–604.

[6] 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 *Proc. 18th ACM/IEEE Int. Conf. Inf. Process. Sens. Netw. (IPSN)*, Montreal, QC, Canada, Apr. 2019, pp. 145–156.

[7] H. Balakrishnan and R. H. Katz, "Explicit loss notification and wireless web performance," in *Proc. IEEE Globecom Internet Mini-Conf.*, Sydney, NSW, Australia, Nov. 1998.

[8] A. Woo, T. Tong, and D. Culler, "Taming the underlying challenges of reliable multihop routing in sensor networks," in *Proc. 1st Int. Conf. Embedded Netw. Sens. Syst.*, Nov. 2003, pp. 14–27.

[9] M. Senel, K. Chintalapudi, D. Lal, A. Keshavarzian, and E. J. Coyle, "A Kalman filter based link quality estimation scheme for wireless sensor networks," in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, Washington, DC, USA, Nov. 2007, pp. 875–880.

[10] R. Fonseca, O. Gnawali, K. Jamieson, and P. Levis, "Four-bit wireless link estimation," in *Proc. ACM SIGCOMM 6th Workshop Hot Topics Netw. HotNets-VI)*, Atlanta, Georgia, Nov. 2007.

[11] K. Srinivasan, M. A. Kazandjieva, M. Jain, and P. Levis. (2008). *PRR is Not Enough*. [Online]. Available: http://sing.stanford.edu/pubs/sing-08-01.pdf

[12] C. A. Boano, M. A. Zúñiga, T. Voigt, A. Willig, and K. Römer, "The triangle metric: Fast link quality estimation for mobile wireless sensor networks," in *Proc. 19th Int. Conf. Comput. Commun. Netw.*, Zurich, Switzerland, Aug. 2010, pp. 1–7.

[13] N. Baccour *et al.*, "F-LQE: A fuzzy link quality estimator for wireless sensor networks," in *Proc. Eur. Conf. Wireless Sens. Netw.*, Coimbra, Portugal, Feb. 2010, pp. 240–255.

[14] 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 J.*, vol. 13, no. 10, pp. 3605–3615, Oct. 2013.

[15] S. Rekik, N. Baccour, M. Jmaiel, and K. Drira, "Low-power link quality estimation in smart grid environments," in *Proc. Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, Dubrovnik, Croatia, Aug. 2015, pp. 1211–1216.

[16] H.-J. Audéoud and M. Heusse, "Quick and efficient link quality estimation in wireless sensors networks," in *Proc. 14th Annu. Conf. Wireless On-Demand Netw. Syst. Serv. (WONS)*, Isola, France, 2018, pp. 87–90.

[17] T. Liu and A. E. Cerpa, "Foresee (4C): Wireless link prediction using link features," in *Proc. 10th Int. Conf. Inf. Process. Sens. Netw. (IPSN)*, Chicago, IL, USA, Apr. 2011, pp. 294–305.

[18] T. Liu and A. E. Cerpa, "Temporal adaptive link quality prediction with online learning," *ACM Trans. Sens. Netw.*, vol. 10, no. 3, p. 46, 2014.

[19] 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. 12788–12797, 2017.

[20] V.-S. Feng and S. Y. Chang, "Determination of wireless networks parameters through parallel hierarchical support vector machines," *IEEE Trans. Parallel Distrib. Syst.*, vol. 23, no. 3, pp. 505–512, Mar. 2012.

[21] K. Bregar and M. Mohorčič, "Improving indoor localization using convolutional neural networks on computationally restricted devices," *IEEE Access*, vol. 6, pp. 17429–17441, 2018.

[22] A. Khan, S. Wang, and Z. Zhu, "Angle-of-arrival estimation using an adaptive machine learning framework," *IEEE Commun. Lett.*, vol. 23, no. 2, pp. 294–297, Feb. 2019.

[23] A. Caciularu and D. Burshtein, "Blind channel equalization using variational autoencoders," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, Kansas City, MO, USA, May 2018, pp. 1–6.

[24] P. M. Olmos, J. J. Murillo-Fuentes, and F. Perez-Cruz, "Joint nonlinear channel equalization and soft LDPC decoding with Gaussian processes," *IEEE Trans. Signal Process.*, vol. 58, no. 3, pp. 1183–1192, Mar. 2010.

[25] T. J. O'Shea, T. Roy, and T. C. Clancy, "Over-the-air deep learning based radio signal classification," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 1, pp. 168–179, Feb. 2018.

[26] E. Nachmani, E. Marciano, L. Lugosch, W. J. Gross, D. Burshtein, and Y. Be'ery, "Deep learning methods for improved decoding of linear codes," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 1, pp. 119–131, Feb. 2018.

[27] P. Siyari, H. Rahbari, and M. Krunz, "Lightweight machine learning for efficient frequency-offset-aware demodulation," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 11, pp. 2544–2558, Nov. 2019.

[28] N. Farsad and A. Goldsmith, "Neural network detection of data sequences in communication systems," *IEEE Trans. Signal Process.*, vol. 66, no. 21, pp. 5663–5678, Nov. 2018.

[29] H. Ye, G. Y. Li, and B.-H. Juang, "Power of deep learning for channel estimation and signal detection in OFDM systems," *IEEE Wireless Commun. Lett.*, vol. 7, no. 1, pp. 114–117, Feb. 2018.

[30] D. Neumann, T. Wiese, and W. Utschick, "Learning the MMSE channel estimator," *IEEE Trans. Signal Process.*, vol. 66, no. 11, pp. 2905–2917, Jun. 2018.

[31] M. Sanchez-Fernandez, M. de-Prado-Cumplido, J. Arenas-Garcia, and F. Perez-Cruz, "SVM multiregression for nonlinear channel estimation in multiple-input multiple-output systems," *IEEE Trans. Signal Process.*, vol. 52, no. 8, pp. 2298–2307, Aug. 2004.

[32] J. A. Cal-Braz, L. J. Matos, and E. Cataldo, "The relevance vector machine applied to the modeling of wireless channels," *IEEE Trans. Antennas Propag.*, vol. 61, no. 12, pp. 6157–6167, Dec. 2013.

[33] R. He *et al.*, "A kernel-power-density-based algorithm for channel multipath components clustering," *IEEE Trans. Wireless Commun.*, vol. 16, no. 11, pp. 7138–7151, Nov. 2017.

[34] C. Di, B. Zhang, Q. Liang, S. Li, and Y. Guo, "Learning automata-based access class barring scheme for massive random access in machine-to-machine communications," *IEEE Internet Things J.*, vol. 6, no. 4, pp. 6007–6017, Aug. 2019.

[35] T. Joshi, D. Ahuja, D. Singh, and D. P. Agrawal, "SARA: Stochastic automata rate adaptation for IEEE 802.11 networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 19, no. 11, pp. 1579–1590, Nov. 2008.

[36] S. M. Srinivasan, T. Truong-Huu, and M. Gurusamy, "Machine learning-based link fault identification and localization in complex networks," *IEEE Internet Things J.*, vol. 6, no. 4, pp. 6556–6566, Aug. 2019.

[37] P. Lin and T. Lin, "Machine-learning-based adaptive approach for frame-size optimization in wireless LAN environments," *IEEE Trans. Veh. Technol.*, vol. 58, no. 9, pp. 5060–5073, Nov. 2009.

[38] T. Liu and A. E. Cerpa, "TALENT: Temporal adaptive link estimator with no training," in *Proc. 10th ACM Conf. Embedded Netw. Sens. Syst.*, Toronto, ON, Canada, Nov. 2012, pp. 253–266.

[39] G. Cerar, H. Yetgin, M. Mohor˘ cič, and C. Fortuna, "On designing a machine learning based wireless link quality classifier," in *Proc. 31st Int. Symp. Pers. Indoor Mobile Radio Commun.*, London, U.K., 2020, pp. 1–7.

[40] 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. 21572–21583, 2019.

[41] Y. Wang, Y. Xiang, J. Zhang, W. Zhou, G. Wei, and L. T. Yang, "Internet traffic classification using constrained clustering," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 11, pp. 2932–2943, Nov. 2014.

[42] X. Yun, Y. Wang, Y. Zhang, and Y. Zhou, "A semantics-aware approach to the automated network protocol identification," *IEEE/ACM Trans. Netw.*, vol. 24, no. 1, pp. 583–595, Feb. 2016.

[43] Q. Scheitle, O. Gasser, M. Rouhi, and G. Carle, "Large-scale classification of IPv6-IPv4 siblings with variable clock skew," in *Proc. Netw. Traffic Meas. Anal. Conf. (TMA)*, Dublin, Ireland, Jun. 2017, pp. 1–9.

[44] J. Dowling, E. Curran, R. Cunningham, and V. Cahill, "Using feedback in collaborative reinforcement learning to adaptively optimize MANET routing," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 35, no. 3, pp. 360–372, May 2005.

[45] P. Charonyktakis, M. Plakia, I. Tsamardinos, and M. Papadopouli, "On user-centric modular QoE prediction for VoIP based on machine-learning algorithms," *IEEE Trans. Mobile Comput.*, vol. 15, no. 6, pp. 1443–1456, Jun. 2016.

[46] R. Ferdous, R. L. Cigno, and A. Zorat, "On the use of SVMs to detect anomalies in a stream of SIP messages," in *Proc. 11th Int. Conf. Mach. Learn. Appl.*, vol. 1. Boca Raton, FL, USA, Dec. 2012, pp. 592–597.

[47] H. Li, K. Ota, and M. Dong, "Learning IoT in edge: Deep learning for the Internet of Things with edge computing," *IEEE Netw.*, vol. 32, no. 1, pp. 96–101, Jan./Feb. 2018.

[48] A. Azarfar, J.-F. Frigon, and B. Sanso, "Improving the reliability of wireless networks using cognitive radios," *IEEE Commun. Surveys Tuts.*, vol. 14, no. 2, pp. 338–354, 2nd Quart., 2012.

[49] Q. Dong and W. Dargie, "A survey on mobility and mobility-aware MAC protocols in wireless sensor networks," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 1, pp. 88–100, 1st Quart., 2013.

[50] H. Shi, R. V. Prasad, E. Onur, and I. G. M. M. Niemegeers, "Fairness in wireless networks:Issues, measures and challenges," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 1, pp. 5–24, 1st Quart., 2014.

[51] L. Gupta, R. Jain, and G. Vaszkun, "Survey of important issues in UAV communication networks," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 2, pp. 1123–1152, 2nd Quart., 2016.

[52] S. Jiang, "On reliable data transfer in underwater acoustic networks: A survey from networking perspective," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 2, pp. 1036–1055, 2nd Quart., 2018.

[53] I. A. Alimi, A. L. Teixeira, and P. P. Monteiro, "Toward an efficient C-RAN optical fronthaul for the future networks: A tutorial on technologies, requirements, challenges, and solutions," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 1, pp. 708–769, 1st Quart., 2018.

[54] Q. Mao, F. Hu, and Q. Hao, "Deep learning for intelligent wireless networks: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 2595–2621, 4th Quart., 2018.

[55] C. Zhang, P. Patras, and H. Haddadi, "Deep learning in mobile and wireless networking: A survey," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2224–2287, 3rd Quart., 2019.

[56] M. Amjad, L. Musavian, and M. H. Rehmani, "Effective capacity in wireless networks: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 4, pp. 3007–3038, 4th Quart., 2019.

[57] S. K. Sharma and X. Wang, "Toward massive machine type communications in ultra-dense cellular IoT networks: Current issues and machine learning-assisted solutions," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 1, pp. 426–471, 1st Quart., 2019.

[58] C. Fortuna, E. De Poorter, P. Škraba, and I. Moerman, "Data driven wireless network design: A multi-level modeling approach," *Wireless Pers. Commun.*, vol. 88, no. 1, pp. 63–77, 2016.

[59] J. Jiang, V. Sekar, I. Stoica, and H. Zhang, "Unleashing the potential of data-driven networking," in *Proc. Int. Conf. Commun. Syst. Netw.*, 2017, pp. 110–126.

[60] M. Z. Zheleva *et al.*, "Enabling a nationwide radio frequency inventory using the spectrum observatory," *IEEE Trans. Mobile Comput.*, vol. 17, no. 2, pp. 362–375, Feb. 2018.

[61] S. Rajendran *et al.*, "Electrosense: Open and big spectrum data," *IEEE Commun. Mag.*, vol. 56, no. 1, pp. 210–217, Jan. 2018.

[62] C. Fortuna and M. Mohorcic, "Trends in the development of communication networks: Cognitive networks," *Comput. Netw.*, vol. 53, no. 9, pp. 1354–1376, 2009.

[63] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, "From data mining to knowledge discovery in databases," *AI Mag.*, vol. 17, no. 3, p. 37, 1996.

[64] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, *Data Mining: Practical Machine Learning Tools and Techniques*. Cambridge, MA, USA: Morgan Kaufmann, 2016.

[65] Y. Gil *et al.*, "Examining the challenges of scientific workflows," *Computer*, vol. 40, no. 12, pp. 24–32, Dec. 2007.

[66] J. J. Van Bavel, P. Mende-Siedlecki, W. J. Brady, and D. A. Reinero, "Contextual sensitivity in scientific reproducibility," *Proc. Nat. Acad. Sci.*, vol. 113, no. 23, pp. 6454–6459, 2016.

[67] M. Baker, "1,500 scientists lift the lid on reproducibility," *Nat. News*, vol. 533, no. 7604, pp. 452–454, 2016.

[68] P. Millan *et al.*, "Time series analysis to predict link quality of wireless community networks," *Comput. Netw.*, vol. 93, no. 2, pp. 342–358, Dec. 2015.

[69] 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," *Comput. Commun.*, vol. 112, pp. 1–13, Nov. 2017.

[70] H. Okamoto, T. Nishio, M. Morikura, K. Yamamoto, D. Murayama, and K. Nakahira, "Machine-learning-based throughput estimation using images for mmWave communications," in *Proc. IEEE 85th Veh. Technol. Conf. (VTC Spring)*, Sydney, NSW, Australia, 2017, pp. 1–6.

[71] R. Van Nee, G. Awater, M. Morikura, H. Takanashi, M. Webster, and K. W. Halford, "New high-rate wireless LAN standards," *IEEE Commun. Mag.*, vol. 37, no. 12, pp. 82–88, Dec. 1999.

[72] 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," *Comput. Netw.*, vol. 132, pp. 68–80, Feb. 2018.

[73] P. Levis *et al.*, "TinyOS: An operating system for sensor networks," in *Ambient Intelligence*. Heidelberg, Germany: Springer, 2005, pp. 115–148.

[74] 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," *Chin. J. Electron.*, vol. 26, no. 2, pp. 377–384, Apr. 2017.

[75] N. Baccour *et al.*, "RadiaLE: A framework for designing and assessing link quality estimators in wireless sensor networks," *Ad Hoc Netw.*, vol. 9, no. 7, pp. 1165–1185, Sep. 2011.

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

[77] D. S. J. De Couto, D. Aguayo, J. Bicket, and R. Morris, "A high-throughput path metric for multi-hop wireless routing," *Wireless Netw.*, vol. 11, no. 4, pp. 419–434, 2005.

[78] A. Cerpa, J. L. Wong, M. Potkonjak, and D. Estrin, "Temporal properties of low power wireless links: Modeling and implications on multi-hop routing," in *Proc. 6th ACM Int. Symp. Mobile Ad Hoc Netw. Comput.*, 2005, pp. 414–425.

[79] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proc. Int. Conf. Comput. Stat. (COMPSTAT')*, 2010, pp. 177–186.

[80] L. B. Almeida, T. Langlois, J. D. Amaral, and A. Plakhov, "Parameter adaptation in stochastic optimization," in *On-Line Learning in Neural Networks*. Cambridge, U.K.: Cambridge Univ. Press, 1999, pp. 111–134.

[81] M. H. Alizai, O. Landsiedel, J. Á. B. Link, S. Götz, and K. Wehrle, "Bursty traffic over bursty links," in *Proc. 7th ACM Conf. Embedded Netw. Sens. Syst.*, 2009, pp. 71–84.

[82] J. Luo, L. Yu, D. Zhang, Z. Xia, and W. Chen, "A new link quality estimation mechanism based on LQI in WSN," *Inf. Technol. J.*, vol. 12, no. 8, pp. 1626–1631, Apr. 2013.

[83] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis, "Collection tree protocol," in *Proc. 7th ACM Conf. Embedded Netw. Sens. Syst.*, 2009, pp. 1–14.

[84] R. Pedersen and M. Schoeberl, "An embedded support vector machine," in *Proc. IEEE Int. Workshop Intell. Solutions Embedded Syst.*, Vienna, Austria, 2006, pp. 1–11.

[85] B. Guo, S. R. Gunn, R. I. Damper, and J. D. B. Nelson, "Customizing kernel functions for SVM-based hyperspectral image classification," *IEEE Trans. Image Process.*, vol. 17, pp. 622–629, 2008.

[86] J. Ahmad, I. Mehmood, S. Rho, N. Chilamkurti, and S. W. Baik, "Embedded deep vision in smart cameras for multi-view objects representation and retrieval," *Comput. Elect. Eng.*, vol. 61, pp. 297–311, Jul. 2017.

[87] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.

[88] A. Banerjee and S. Basu, "Topic models over text streams: A study of batch and online unsupervised learning," in *Proc. SIAM Int. Conf. Data Min.*, 2007, pp. 431–436.

[89] P. Ruckebusch, E. De Poorter, C. Fortuna, and I. Moerman, "GITAR: Generic extension for Internet-of-Things architectures enabling dynamic updates of network and application modules," *Ad Hoc Netw.*, vol. 36, pp. 127–151, Jan. 2016.

[90] M. Kulin, C. Fortuna, E. De Poorter, D. Deschrijver, and I. Moerman, "Data-driven design of intelligent wireless networks: An overview and tutorial," *Sensors*, vol. 16, no. 6, p. 790, 2016.

[91] A. A. Freitas, "Understanding the crucial role of attribute interaction in data mining," *Artif. Intell. Rev.*, vol. 16, no. 3, pp. 177–199, Nov. 2001.

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

[93] B. W. Yap, K. A. Rani, H. A. A. Rahman, S. Fong, Z. Khairudin, and N. N. Abdullah, "An application of oversampling, undersampling, bagging and boosting in handling imbalanced datasets," in *Proc. 1st Int. Conf. Adv. Data Inf. Eng. (DaEng)*, Dec. 2014, pp. 13–22.

[94] D. Aguayo, J. Bicket, S. Biswas, G. Judd, and R. Morris, "Link-level measurements from an 802.11b mesh network," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 34, no. 4, pp. 121–132, 2004.

[95] D. Gokhale, S. Sen, K. Chebrolu, and B. Raman, "On the feasibility of the link abstraction in (rural) mesh networks," in *Proc. IEEE INFOCOM 27th Conf. Comput. Commun.*, 2008, pp. 61–65.

[96] S. K. Kaul, M. Gruteser, and I. Seskar, "Creating wireless multi-hop topologies on space-constrained indoor testbeds through noise injection," in *Proc. 2nd Int. Conf. Testbeds Res. Infrastruct. Develop. Netw. Communities (TRIDENTCOM)*, Barcelona, Spain, Mar. 2006, p. 10.

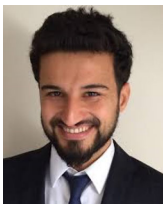
[97] S. Fu, Y. Zhang, Y. Jiang, C. Hu, C.-Y. Shih, and P. J. Marrón, "Experimental study for multi-layer parameter configuration of WSN links," in *Proc. IEEE 35th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Columbus, OH, USA, 2015, pp. 369–378.

[98] K. Bauer, D. McCoy, B. Greenstein, D. Grunwald, and D. Sicker, "Physical layer attacks on unlinkability in wireless LANs," in *Proc. Int. Symp. Privacy Enhancing Technol. Symp.*, 2009, pp. 108–127.

[99] Y. Chen, A. Wiesel, and A. O. Hero, "Robust shrinkage estimation of high-dimensional covariance matrices," *IEEE Trans. Signal Process.*, vol. 59, no. 9, pp. 4097–4107, Sep. 2011.

[100] T. Van Haute *et al.*, "Platform for benchmarking of RF-based indoor localization solutions," *IEEE Commun. Mag.*, vol. 53, no. 9, pp. 126–133, Sep. 2015.
[101] E. Anderson, G. Yee, C. Phillips, D. Sicker, and D. Grunwald, "The impact of directional antenna models on simulation accuracy," in *Proc. IEEE 7th Int. Symp. Model. Optim. Mobile Ad Hoc Wireless Netw. (WiOPT)*, Seoul, South Korea, 2009, pp. 1–7.
[102] E. Anderson, C. Phillips, D. Sicker, and D. Grunwald, "Modeling environmental effects on directionality in wireless networks," *Math. Comput. Model.*, vol. 53, nos. 11–12, pp. 2078–2092, 2011.
[103] Y.-A. Le Borgne, J.-M. Dricot, and G. Bontempi, "Principal component aggregation for energy efficient information extraction in wireless sensor networks," *Knowl. Discov. Sens. Data*, to be published.
[104] F. Fenicia *et al.*, "Microwave links for rainfall estimation in an urban environment: Insights from an experimental setup in Luxembourg-City," *J. Hydrol.*, vols. 464–465, pp. 69–78, Sep. 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0022169412005483
[105] A. Kelmendi *et al.*, "Rain attenuation prediction model based on hyperbolic cosecant copula for multiple site diversity systems in satellite communications," *IEEE Trans. Antennas Propag.*, vol. 65, no. 9, pp. 4768–4779, Sep. 2017.
[106] M. Spuhler, V. Lenders, and D. Giustiniano, "BLITZ: Wireless link quality estimation in the dark," in *Wireless Sensor Networks*. Heidelberg, Germany: Springer, Feb. 2013, pp. 99–114.
[107] L. A. Jeni, J. F. Cohn, and F. De La Torre, "Facing imbalanced data–recommendations for the use of performance metrics," in *Proc. IEEE Humaine Assoc. Conf. Affect. Comput. Intell. Interact.*, Geneva, Switzerland, 2013, pp. 245–251.
[108] I. Goodfellow *et al.*, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*. Red Hook, NY, USA: Curran, 2014, pp. 2672–2680.
[109] H. Ye, G. Y. Li, B.-H. F. Juang, and K. Sivanesan, "Channel agnostic end-to-end learning based communication systems with conditional GAN," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Abu Dhabi, UAE, Dec. 2018, pp. 1–5.
[110] Y. Yang, Y. Li, W. Zhang, F. Qin, P. Zhu, and C.-X. Wang, "Generative-adversarial-network-based wireless channel modeling: Challenges and opportunities," *IEEE Commun. Mag.*, vol. 57, no. 3, pp. 22–27, Mar. 2019.
[111] H. Haggui, S. Affes, and F. Bellili, "FPGA-SDR integration and experimental validation of a joint DA ML SNR and doppler spread estimator for 5G cognitive transceivers," *IEEE Access*, vol. 7, pp. 69464–69480, 2019.
[112] T. Gruber, S. Cammerer, J. Hoydis, and S. Ten Brink, "On deep learning-based channel decoding," in *Proc. IEEE 51st Annu. Conf. Inf. Sci. Syst. (CISS)*, Baltimore, MD, USA, 2017, pp. 1–6.
[113] A. M. Koushik, F. Hu and S. Kumar, "Deep Q -learning-based node positioning for throughput-optimal communications in dynamic UAV swarm network," *IEEE Trans. Cogn. Commun. Netw.*, vol. 5, no. 3, pp. 554–566, Sep. 2019.
[114] L. Liu, Y. Cheng, L. Cai, S. Zhou, and Z. Niu, "Deep learning based optimization in wireless network," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Paris, France, May 2017, pp. 1–6.

Gregor Cerar (Graduate Student Member, IEEE) received the master's degree in telecommunications from the Faculty of Electrical Engineering, University of Ljubljana in 2016. He is currently pursuing a Ph.D. degree with the Jožef Stefan International Postgraduate School. He is also a Research Assistant with the Department of Communication Systems, Jožef Stefan Institute. His main research interests are in wireless networking of constrained devices, anomaly detection, and machine learning and deep learning applications in IoT.



Halil Yetgin (Member, IEEE) received the B.Eng. degree in computer engineering from Selcuk University, Turkey, in 2008, the M.Sc. degree in wireless communications from the University of Southampton, U.K., in 2010, and the Ph.D. degree in wireless communications from the Next Generation Wireless Research Group, University of Southampton in 2015. He is an Assistant Professor of the Department of Electrical and Electronics Engineering, Bitlis Eren University, Turkey, and a Research Fellow of the Department of Communication Systems, Jožef Stefan Institute, Ljubljana, Slovenia. His research interests include the development of intelligent communication systems, energy efficient cross-layer design, resource allocation of the future wireless communication networks and machine learning for wireless networks. He was a recipient of the Full Scholarship granted by the Republic of Turkey, Ministry of National Education. He is an Associate Editor of IEEE ACCESS. He was a TPC member for IEEE VTC-2018, VTC-2019, VTC-2020, and IEEE Globecom-2020

Mihael Mohorčič (Senior Member, IEEE) received the B.Sc., M.Sc., and Ph.D. degrees in electrical engineering from the University of Ljubljana, Ljubljana, in 1994, 1998, and 2002, respectively, and the M.Phil. degree in electrical engineering from the University of Bradford, Bradford, U.K., in 1998. He is currently the Head of the Department of Communication Systems and a Scientific Counsellor with the Jožef Stefan Institute, and an Associate Professor with the Jožef Stefan International Postgraduate School. He has been participating in many EC co-funded research projects since 1996, and he is currently involved in H2020 projects Fed4FIRE+, SAAM, RESILOC, and BD4OPEM. He has authored or coauthored over 200 refereed journal and conference papers, coauthored three books and contributed to nine book chapters. His research interests include the development and performance evaluation of network protocols and architectures for mobile and wireless communication systems, and resource management in terrestrial, stratospheric, and satellite networks. His recent research interest is focused on cognitive radio networks, smart applications of wireless sensor networks, dynamic composition of communication services, and wireless experimental testbeds.

Carolina Fortuna received the B.Sc. degree in 2006, and the Ph.D. degree in 2013. She was a Postdoctoral Research Associate with IBCN, Ghent University, from 2014 to 2015, and visited Stanford Infolab in 2017. She is currently a Research Fellow and the Head of the Networked Embedded Systems Laboratory, Department of Communication Systems, Jožef Stefan Institute. She has participated in H2020, FP7, and FP6 Projects, including in various leadership roles. Her research is interdisciplinary, focusing on data and knowledge driven modeling of communication and sensor systems. She has coauthored over 50 peer-reviewed publications, edited a book, was a TPC member at IEEE ICC from 2011 to 2021, ESWC in 2012, IEEE Globecom from 2011 to 2021, VTC in 2010, 2016, and IEEE WCNC in 2009.

Chapter 3

Designing a Machine Learning Based Wireless Link Quality Classifier

In the previous chapter, we comprehensively studied and analyzed existing data-driven estimators of wireless link quality developed from empirical data, focusing on ML-based approaches. We considered them from the perspective of how they address quality requirements and how they approach standard design steps commonly used in the ML community. We found that the existing literature rarely describes in detail the design decisions authors made when developing ML-based LQE models. This inspired us to suggest design guidelines for developing ML-based LQE models and for generic trace-set collection as well as to investigate whether different design decisions would make a difference in the effectiveness of link quality estimation.

Effective link quality estimation is particularly important for wireless networks in dynamic propagation environments, where radio signal varies in time and space. Various phenomena affecting the wireless link quality in such environments can only be adequately described by data-driven link quality models, where those based on ML algorithms that conduct classification of wireless links in different quality classes recently became preferred to statistical ones, but since they are less explainable, they require particular attention in the design phase as we show in this chapter.

This chapter extends Chapter 2 and focuses on the methodology of designing an ML-based wireless link quality classifier to achieve accurate short-term prediction of wireless link quality. The methodology consists of three major stages, namely, data pre-processing, model building and model evaluation.

The data pre-processing is the most time-consuming stage, but it tends to have a significant influence on the final performance of a model. We divide the pre-processing stage into five steps: cleaning and interpolation step, feature engineering step, observation window selection step, resampling strategy step, and finally model selection step. We show that using domain knowledge at cleaning and interpolation gives better results than interpolation with Gaussian noise or dropping invalid values. We demonstrate 8% overall difference in accuracy, where the per-class difference is up to 77%. At the feature engineering step, we investigated more meaningful features from a time-series of received signal strength indicator (RSSI) values for more accurate estimation. We show that the combination of current value, rolling average and standard deviation of signal strength over an observation time window gives the best results. When comparing performance against using only instant RSSI, the measured difference in accuracy is about 6%. Next, we learn that the observation window size balances between stability and reactivity. In other words, larger observation window size produces more stable, while shorter observation window produces more reactive estimator. At the resampling strategy step, we investigate the importance

of balancing the size of classes. In a case where all classes are equally important, we show that balancing the classes with random resampling improves the underrepresented classes' classification performance for 44%.

We evaluate the performance across five different well-known algorithms and dummy majority classifier at the algorithm selection step. The evaluation shows similar performance between linear and non-linear algorithms, where non-linear algorithms offer slightly better performance than linear algorithms.

From the hypotheses outlined in Chapter 1.1, in this chapter, we address and partially confirm hypotheses **H1** and **H3**:

H1 Wireless link quality can be efficiently and accurately estimated using machine learning approaches.

H3 Data pre-processing and algorithm parametrization have significant impact on the performance of wireless link quality estimation and wireless link anomaly detection.

Similar to the state of the art presented in Chapter 2, this chapter also confirms hypothesis **H1**. With the presented classification performance, we demonstrate that machine learning approaches can efficiently and accurately estimate wireless link quality.

A large part of this chapter is dedicated to the data pre-processing steps. We demonstrate how significant influence these steps have on the final performance of LQE, which confirms hypothesis **H3**.

As to the contributions outlined in Chapter 1.3, this chapter represents part of contribution **C2** by providing a systematic investigation of the impact of common pre-processing steps in KDP methodology on the final performance of data-driven ML-based LQE. The performance is measured using standard classification metrics.

The publication included in this chapter is:

- G. Cerar, H. Yetgin, M. Mohorčič and C. Fortuna, *On Designing a Machine Learning Based Wireless Link Quality Classifier*, 2020 IEEE 31st Annual International Symposium on Personal, Indoor and Mobile Radio Communications, London, UK, 2020, pp. 1-7, doi: 10.1109/PIMRC48278.2020.9217171.

On Designing a Machine Learning Based Wireless Link Quality Classifier

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*—Ensuring a reliable communication in wireless networks strictly depends on the effective estimation of the link quality, which is particularly challenging when propagation environment for radio signals significantly varies. In such environments, intelligent algorithms that can provide robust, resilient and adaptive links are being investigated to complement traditional algorithms in maintaining a reliable communication. In this respect, the data-driven link quality estimation (LQE) using machine learning (ML) algorithms is one of the most promising approaches. In this paper, we provide a quantitative evaluation of design decisions taken at each step involved in developing a ML-based wireless LQE on a selected, publicly available dataset. Our study shows that, re-sampling to achieve training class balance and feature engineering have a larger impact on the final performance of the LQE than the selection of the ML method on the selected data.**

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

I. Introduction

Machine learning (ML) is becoming an increasingly popular way of solving various aspects 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 ML algorithms that approximate the distribution of the underlying random variable and are thus able to learn the quality of a link [4], [5].

LQE developed using ML can estimate the quality of a link in a continuous value space, in this case the ML performs a regression [4], [6]–[10]. Alternatively, if they estimate the quality in a discrete value space, the ML performs classification [5], [11]–[13]. By analyzing the existing body of work developing classification models for LQE, we notice the following approaches: binary, two class or multi-class.

The first type is a *binary or a two-class output*, which is produced by the classification model. This type of output can be found in [3], [11], [12], [14], [15]. The applications noticed are mainly (binary) decision making [3] and above/below threshold estimation [11], [12], [14], [15].

The second type is *multi-class output* value. Similar to the first type, it is also produced by the classification model. The multi-class output values are utilized in [5], [13], [16]–[19], where [17], [19] use a three-class, [16] utilizes a four-class, [13], [18] rely on a five-class, and [5] leverages a seven-class output. The applications observed are the categorization and estimation of the future LQE state, which is expressed through labels/classes. It is not always clear from the related work how the authors select the number of classes. However, according to [20], a wireless link seems to follow a non-linear S-shaped curve with three regions. All works using a three class output model seem to consider this characteristic of the link.

For developing ML models in any application area, generally some very precise steps that are well established in the community are followed [21], [22], namely data preprocessing, model building and model evaluation. The data preprocessing stage is known to be the most time-consuming process and tends to have a major influence on the final performance of the model. This stage includes several steps such as data cleaning and interpolation, feature selection and re-sampling. While most of the identified research developing LQEs explicitly mention aspects of cleaning and interpolation and feature selection, none evaluate the impact of the design decision taken at these steps on the final performance of the ML-based LQE. However, the majority evaluate the impact of the ML method selection on the final performance of the ML-based LQE.

Additionally, none of the works mentions aspects of the re-sampling step, that is particularly critical for the generalization capability of a model. Re-sampling is used in ML communities when the available input data is imbalanced [23], [24]. For instance, assume a classification problem where the aim is to classify links into *good*, *bad* and *intermediate* classes, similar to the problem approached in [17], [19]. If the *good* class would represent 75% of the examples in the training dataset, *bad* would represent 20% and *intermediate* would represent the remaining 5%, then a ML model would likely be well trained to recognize the *good* as it has been exposed to many such instances, however it might have difficulties in recognizing the other two minority classes.

In this paper, we aim to show the impact of design decisions taken at each step of the process of designing a ML-based LQE model on the final performance of the model. To realize our aim, we first select the Rutgers publicly available dataset [25]

978-1-7281-4490-0/20/$31.00 © 2020 IEEE

and a decision tree[1] as a representative ML-based classification model. Then, we systematically perform each step of the knowledge discovery process [21] on the selected dataset using the selected model, meanwhile varying the design parameters at each step. This way, we are able to systematically quantify the influence of each of the design steps on the final performance of our classifier, therefore providing an in-depth step by step understanding on the process of learning to classify wireless links.

The main contributions of this paper are:

- A systematic quantification of the influence of the design steps on the final performance of our wireless link quality classifier is provided. The highlights of the quantification are that, for the chosen problem and dataset, the generation of synthetic features from the only available training feature received signal strength indicator ($RSSI$), yields up to 6% higher accuracy and is able to better discriminate the intermediate class up to 49%. The choice of ML method has less, relatively smaller impact on the final model performance with all the selected algorithms yielding an accuracy performance between 94% and 95% and minority class is detected between 87% and 89%.
- A first time evaluation of the impact of re-sampling on wireless link quality classification is realized using ML. In the case of the chosen imbalanced dataset, by using standard re-sampling, the minority class is correctly detected in over 87% of the instances, yielding more than 25 percentage points increase in the performance and comes at a small 2% decrease in overall accuracy.

The remainder of the paper is organized as follows. Section II elaborates on the selected dataset and Section III analyses the importance of the cleaning and interpolation steps applied to the selected dataset. Then, the importance of feature engineering, window selection and re-sampling strategy is emphasized in Section IV, while Section V examines the influence of the model selection on the final performance of the LQE classifier. Then, potential generalization perspectives of the proposed models and findings for other datasets are discussed in Section VI. Finally, Section VII concludes the paper.

II. Rutgers dataset summary

The Rutgers trace-set [25] 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. In this particular experiment, we observe that 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 ranges from 0 to 128, where the value 0 represents a bad link with no signal and 127 indicates a good link with strong signal, while observed value of 128 represents an error. Nonetheless, 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.

All the scripts developed for the comparative performance analyses are publicly available [2] for researchers to reproduce, re-use on other data-sets and improve upon our analyses.

III. Analysis of cleaning & interpolation steps

The first step of the process of developing a ML-based LQE involves data cleaning and interpolation. The reason for that is because models that are automatically created using ML algorithms can be significantly biased as a result of invalid and missing data. First of all, a valid time series corresponding to each link has to be extracted, which is referred to as a series of ordered tuples each of which contains a packet sequence number and corresponding measured link metrics. The obtained values in the tuples have to be within valid ranges. For instance, the sequence numbers have to be identical with the packets sent during the trace collection, and the values of the link metrics have to remain within the valid ranges that are specified by the transceiver data sheets. Roughly speaking, link metrics with regard to the received radio signals, i.e., RSSI and link quality indicator (LQI) can be extracted directly from the hardware registers of the corresponding transceivers, whereas link metrics concerning packet data transmission, i.e., packet reception ratio (PRR) and packet success rate (PSR) are computed with suitable software procedures.

As described in Section II, the Rutgers trace-set contains invalid values and a considerable number of missing sequence numbers due to the lost packets. Most of the available out-of-the-box data mining algorithms cannot handle these invalid values, e.g., *NaN* and $\pm\infty$ of IEEE 754 standard, or they are simply ignored. To quantify the impact of selected cleaning and interpolation approaches to the final performance of the model, we assume the use of a decision tree algorithm trained with a trio of instant RSSI, averaged RSSI and standard deviation RSSI values, stratified k-fold[3] and pruning[4], standard normalization, and random oversampling (ROS) approach, as discussed in Sections IV-A, IV-B and IV-C.

From the perspective of data preprocessing steps for ML models, there are many approaches for handling missing data [26], [27]. To reveal the impact of the approach to missing values on link quality classification, we train the same model, i.e., decision trees, stratified k-fold and pruning, along with the same feature set for the following cases; a) without handling the missing values, b) using a simple time series approach where we interpolate missing data with Gaussian noise, and c) with the aid of domain knowledge. In the case

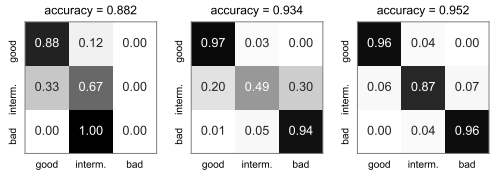
[1]Decision tree is one of the predictive modeling approaches utilized in ML.

[2]ML LQE scripts: https://github.com/sensorlab/link-quality-estimation

[3]K-fold cross-validation is a statistical procedure of splitting data into training and test chunks used for evaluating the ability of ML models.

[4]Pruning is a technique used for shrinking the size of decision trees by discarding sections of the tree that may be too specific and thus lead to overfitting.

of interpolation with Gaussian noise, gaps of missing data are filled with random values based on the previous and next valid values. Regarding domain knowledge, we replace the missing RSSI values with 0, which represents a poor quality link with no received signal, yielding PRR equal to 0. Recalling that possible RSSI values are integers ranging between 0 (bad link with no signal) and 127 (good link with strong signal), while observed value of 128 represents an error.



(a) No interpolation (b) Gaussian (c) Domain knowledge

Fig. 1. Different interpolation cases with a nonlinear decision tree algorithm and ROS.

Fig. 1 presents the relative performance of the models for all three interpolation cases using the form of a confusion matrix[5], i.e., indicating how well the model classifies individual instances. The better the classifier the darker the diagonal of the confusion matrix and the whiter the non-diagonal squares. For this particular case, the best performing model in terms of accuracy (95.2%) is the model using domain knowledge in Fig. 1c. *While the difference in accuracy between the best two models is approximately 2 percentage points, their respective confusion matrices indicate that the model using interpolation with domain knowledge is superior as it better discriminates between the three link types.* This comparison confirms that [11], [13], [30] took the best design decision by using domain knowledge for cleaning and interpolation.

IV. Analysis of feature engineering

The second step of the process of developing a ML-based LQE involves feature engineering. The feature engineering step may involve several sub-steps depending on application requirements, type of data and type of ML problem. For our purpose of learning to classify LQE, we distinguish three sub-steps discussed in the following subsection.

A. Analysis of feature selection

Feature selection is the process of selecting relevant raw features and/or creating synthetic features to be used for training ML models. When the number of possible input features is very large, then usually only the most relevant ones are selected to be used for the model. On the other hand, when the number of possible input features is very low, then creating synthetic features starting from the available ones to aid in better model development is employed. Feature selection is a fundamental step and can be performed manually or, in some cases, can be built automatically by existing algorithms, such as support vector machines (SVMs)[6].

To analyze and understand the influence of feature selection on the model performance, we consider a set of standard feature engineering procedures on the selected dataset. The Rutgers trace-set has only two available attributes useful for LQE, i.e., the instant (raw) RSSI value and the sequence number, therefore exploring synthetic feature generation for model improvement seems to be the only feasible option for this step. The sequence number is leveraged for the computation of PRR which represents the target value, therefore leaving only RSSI as possible training feature. This classification obeys the following rules:

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

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

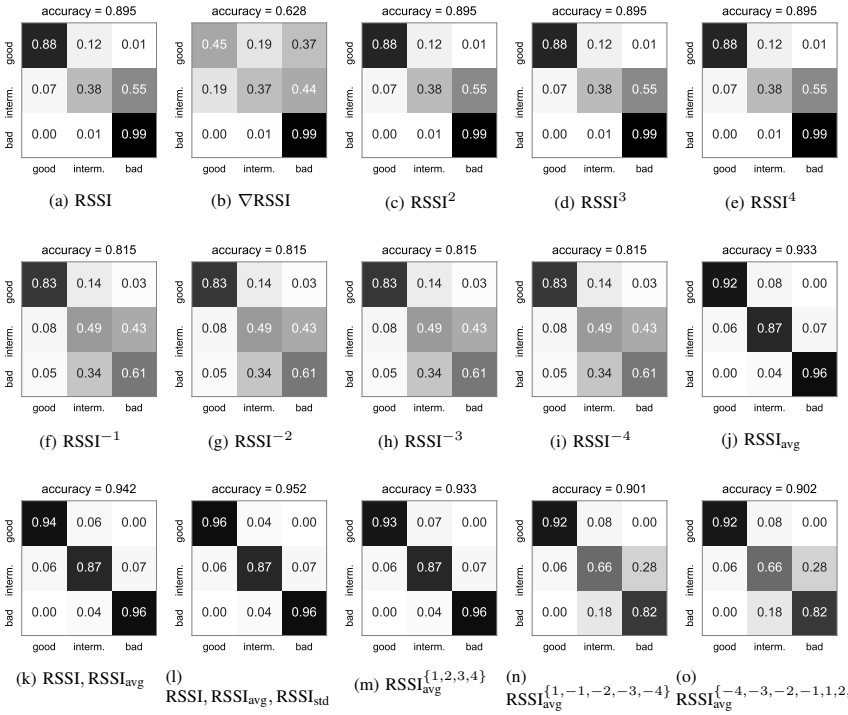
A typical approach in ML for such limited trace-sets is to investigate whether synthetic features, such as average RSSI over a time window or polynomial interactions [31][7], can aid in training to acquire more accurate models compared to that of the instant RSSI values. Fig. 2 shows the influence of the best-performing feature combinations on the classification performance. For this analysis, we assume interpolation based on domain knowledge, i.e., replacing missing values with zeros, as discussed in Section III. Additionally, synthetic feature creation with prediction window size W_{PRR} and historical window size $W_{history}$ are set to 10, while utilizing standard normalization and ROS approach, as discussed in Sections IV-C and IV-B. In this analysis we predict the link quality as per Eq. (1) for the next prediction window W_{PRR}. Noting that the windows W_{PRR} and $W_{history}$ are utilized for computing link quality labels and features, respectively.

We can see from the results listed in Fig. 2(a) that the decision tree based model, trained using stratified k-fold and pruning, that uses the only available feature, $RSSI$, yields 89% accuracy and 38% correctly identified *intermediate* class, we can call this the baseline performance. *The best performing feature combination that uses two synthetically generated features in $RSSI_{avg}$, $RSSI_{std}$ addition to RSSI yields an accuracy of 95.2% and 87% correctly identified intermediate class as can be seen in Fig. 2(l).* Moreover, Fig. 2(j) also shows that $RSSI_{avg}$ alone yields great results, i.e., 93% accuracy and 87% correctly identified *intermediate* class. Positive powers of $RSSI$ have no major advantage over the baseline as can be seen from Fig. 2(c), (d) and (e), while negative powers

[5]Confusion matrix is a table layout, with rows for the instances of a predicted class and columns for the instances of an actual class, used for the problem of statistical classification in order to exhibit the performance of an algorithm. Readers are referred to [28] and [29] for further details.

[6]Support Vector Machine is a prominent ML algorithm seeking solutions for both classification and regression problems.

[7]Polynomial interactions are employed for generating new features from the already available ones with the intention of improving the performance of ML model.



(a) RSSI (b) ∇RSSI (c) RSSI^2 (d) RSSI^3 (e) RSSI^4

(f) RSSI^{-1} (g) RSSI^{-2} (h) RSSI^{-3} (i) RSSI^{-4} (j) RSSI_{avg}

(k) $\text{RSSI}, \text{RSSI}_{\text{avg}}$ (l) $\text{RSSI}, \text{RSSI}_{\text{avg}}, \text{RSSI}_{\text{std}}$ (m) $\text{RSSI}_{\text{avg}}^{\{1,2,3,4\}}$ (n) $\text{RSSI}_{\text{avg}}^{\{1,-1,-2,-3,-4\}}$ (o) $\text{RSSI}_{\text{avg}}^{\{-4,-3,-2,-1,1,2,3,4\}}$

Fig. 2. The influence of feature selection on the performance of the nonlinear model using decision trees.

lower overall accuracy, albeit they perform better then the baseline for the *intermediate* class and significantly worse for the *bad* class as per Fig. 2(f), (g), (h) and (i). In the last row of the table, Figs. 2(k-o) show that other synthetic combinations of RSSI_{avg} perform relatively better than the baseline. *As a conclusion, it can be seen that the generation of synthetic features from the only available training feature RSSI, yields up to 6% higher accuracy and is able to discriminate the intermediate class up to 49% better.*

B. Analysis of window selection

For examining the influence of the window selection on the performance of the model, we need to distinguish between two types of windows. The first one is the historical window $\text{W}_{\text{history}}$ that is used for computing features such as RSSI_{avg}. The second one is the prediction window W_{PRR} that is used for computing the link quality labels. The majority of related works mention details about the window selection step. However, many of them fail to specify the size of the window used for the models they propose and evaluate. Additionally, the window size tends to be smaller for more reactive or online models, such as in [6], [12], while for less reactive models, as proposed in [5], [18], the window size is likely larger.

Given that the investigated Rutgers trace-set consists of 300 packets per link, the size limits for the two windows are within $[0, 300]$ packets, where opting for the value 0 indicates no windowing and favoring the value 300 suggests per link labeling. Therefore, we restrict the range of the window sizes to $[2, 100]$ packets, within which we investigate the performance with a discrete set of nine values $\{2, 5, 10, 15, 20, 30, 50, 80, 100\}$. In this analysis, we predict the link quality for the next prediction window $\text{PRR}(\text{W}_{\text{PRR}})$ considering the Rutgers trace-set with domain knowledge interpolation, the decision tree algorithm, with stratified k-fold and pruning,

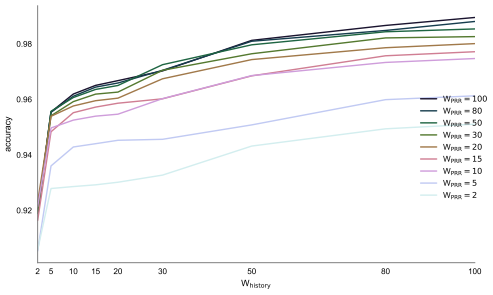


Fig. 3. Overview of the influence of a discrete set of window sizes on the accuracy of the proposed nonlinear model.

the feature vector (RSSI, $RSSI_{avg}(W_{history})$, $RSSI_{std}(W_{history})$), standard normalization and the ROS approach.

As portrayed in Fig. 3, the best performing model is the one utilizing $W_{PRR} = 100$, which predominantly outperforms the models based on other W_{PRR} settings, although all results for window size above 30 are rather similar.

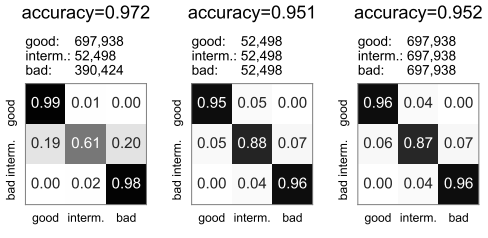
The results, in general, reveal that; (i) a longer historical window improves prediction because there is more information about how the link performed in the past, and (ii) increasing the prediction window (computing the future value of the classes for link quality) also leads to an improvement of the accuracy. Both observations, however, can also be a side-effect of "smoothing"/averaging data from a relatively static trace-set. More explicitly, larger prediction windows are unable to inform on short-term effects, although they can help better in identifying the overall link behavior. It is worth noting that the optimal combination of values for historical and prediction windows is data dependent, however, the trade-offs discussed in this section can be adopted for general models. While the Rutgers trace-set is relatively static, for a more dynamic trace-set the optimal window sizes are likely smaller.

To develop a suitable LQE model, the agility of the model has to be specified by the designer considering dynamically changing environments, e.g., for designing a routing algorithm in a largely mobile wireless network. Additionally, the practical memory limitations of the devices have to be taken into account when developing a suitable LQE model. This is mainly because more agile estimators use smaller window sizes, and therefore they tend to consume less memory, and yet yield low accuracy. Even though larger window sizes assist in attaining high accuracy, the cold start period, during which the historical window is initialized, leads to an estimation delay.

C. Re-sampling strategy

From the analysis of the actual values in the considered Rutgers trace-set, it can be readily observed that there are 61% *good*, 34% *bad* and only 5% *intermediate* class entries. This distribution of data is largely imbalanced due to the artifact of the experiment, where the nodes were close to each other and the interference level was relatively low. Therefore, the majority of the links were actually good as expected and this was not due to the missing values within one particular class category of link quality. Additionally, it has been acknowledged in the literature [20] that the *intermediate* region of the receivers tends to be relatively narrow compared to the *good* and *bad* regions, and therefore this naturally forms a scarcely populated class for intermediate regions in such trace-sets, yet having an important influence to ML-based LQE models although, as mentioned in the introduction as part of the motivation for this work, this aspect has been neglected by all the related work we have reviewed.

Imbalanced trace-sets are often encountered in ML and data mining communities and they are typically dealt with an appropriate re-sampling strategy. For studying the influence of the re-sampling strategy on the performance of the model for link quality classification, we employ the standard ROS and



(a) No resampling (b) Undersampling (c) Oversampling

Fig. 4. Different re-sampling strategies on the pipeline with a standard normalization and nonlinear decision tree algorithm using (RSSI, $RSSI_{avg}$ and $RSSI_{std}$ features.

the random undersampling (RUS) approaches. The ROS [23], [24] approach equalizes all class sizes to the size of the majority class by duplicating the trace-set entries of the minority classes; therefore the resulting re-sampled dataset becomes larger. On the contrary, the RUS [23], [24] approach equalizes all class sizes to the size of the minority class by randomly discarding instances from other larger classes. Hence, the new resampled dataset becomes smaller. It is observed from the numbers of good, intermediate and bad links in Figs. 4(b) and (c) that with both approaches, i.e., ROS and RUS, we are able to acquire a training dataset with balanced classes, about 50k examples for each class of RUS and 690k examples for ROS.

Fig. 4 illustrates that re-sampling strategies on the Rutgers trace-set decrease the overall accuracy of the classification model from 97.2% to slightly above 95%. Some more advanced re-sampling strategies [32] may limit this decrease in performance. However, when no re-sampling is performed, the minority class, i.e., *intermediate* is only correctly detected in 61% of the instances, indicating that the model is over-fitted to the majority of the classes. *In the case of re-sampling, the minority class is correctly detected in over 87% of the instances, yielding more than 25 percentage points increase in the performance.* This improvement comes at a relatively small performance cost for the majority classes, inducing 3-4 percentage points decline for the *good* links and 2 percentage points reduction for the *bad* links.

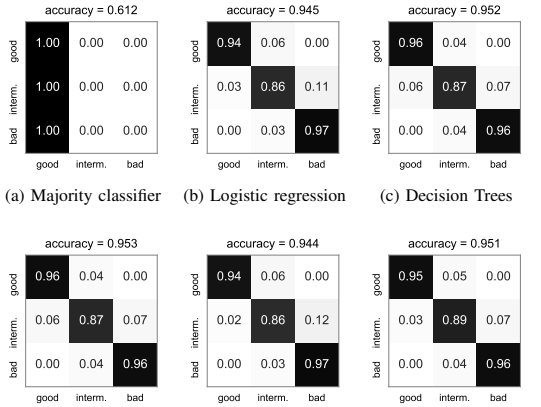
Considering this analysis, we may hint that, in the case of [13], where the performance of the predictor on two of the five classes is modest, employing a resample strategy might lead to better discrimination of those classes. Re-sampling may also improve other proposed estimators, for example the ones in [5], [11], [18], [30].

The results for the selected Rutgers trace-set reveal that there is no significant distinction between the two re-sampling strategies, i.e., RUS and ROS. This is likely due to the relatively large size of the intermediate class. Although the intermediate class only represents 5% of the population, it still contains more than 52,000 samples. However, looking beyond this particular trace-set, the RUS approach may suffer from excluding a certain number of majority class instances

and may affect the representativeness of the remaining data points, especially for more dynamic trace-sets. On the other hand, due to the enlarged number of data points, the ROS approach requires more computing resources for building a model. Note that the results obtained in this section are based on interpolation and cleaning using domain knowledge, instant RSSI, $RSSI_{avg}$ and $RSSI_{std}$ as features and W_{PRR} and $W_{history}$ of size 10.

V. Analysis of model selection

The final step of this systematic analysis is concerned with the influence of the ML algorithm selection on the performance of LQE models. To provide a comparative analysis of the impact, we examine logistic regression and linear SVM as representatives of linear ML algorithms, and decision trees, random forests and a multilayer perceptron[8], as representatives of nonlinear model. As a baseline reference model, we leverage the majority classifier, which in our case, classifies all links in the *good* class.



(a) Majority classifier (b) Logistic regression (c) Decision Trees

(d) Random Forest (e) SVM classifier (f) Multilayer perceptron

Fig. 5. The influence of the choice of ML algorithm on the effectiveness of LQE models.

The analysis in this section is conducted by using domain knowledge interpolation, the feature vector consisting of instant RSSI, $RSSI_{avg}$ and $RSSI_{std}$, windowing with $W_{PRR} = 10$, $W_{history} = 10$, and a ROS approach over the Rutgers trace-set. The selected ML algorithms are evaluated using 10-times stratified K-fold cross-validation [33], [34]. Note that we obey the rule of cumulative parameterization throughout the data preprocessing steps in order to reveal the impact of each step on the ML algorithms for the sake of the LQE model proposed.

Fig. 5 shows that all the selected ML models apart from the reference majority classifier have comparable performance, with an accuracy above 94%. Decision tress, random forests and multi-layer perceptrons, non-linear ML models, are very similar at 95% accuracy. SVM with linear kernel and logistic regression are then at 94%. Slightly lower performance of the linear models such as logistic regression and SVM conforms to the findings in the literature that LQE is a nonlinear function [7], [10], [18], [20]. Looking at the ability of the ML algorithms to identify the minority class, the multilayer perceptron outperforms all the other ML algorithms considered for this analysis.

One of our major observation from the analysis of ML-based LQE models is that the slightly better performance of nonlinear ML-based LQE models to the linear counterparts conforms to the findings in the state-of-the-art literature as it can be observed in [7], [10]. Besides, upon the conclusions drawn in [6], [11], [12], [30], which are mainly compared to 4B [35], we can see that ML-based LQE models consistently outperform the traditional analytical estimators.

VI. Threats to validity

In this paper, we quantify the influence of various design steps of the ML process on the final results for the case of LQE estimation on a single dataset. However, by using alternative datasets, the general conclusions that the preprocessing steps have a relatively higher influence on the final result than the model selection would still hold for any other dataset, albeit the exact numbers will differ from the ones presented herein. Consider linear regression as a representative of linear ML models and decision trees as a representative of non-linear ML models. Roughly speaking, it is expected that both models will perform well when learning to approximate linear problems, but that the decision trees will perform significantly better with non-linear problems. However, with the aid of data preprocessing, more specifically feature interactions, non-linearities can be captured by the engineered features. When fed into the linear model, these engineered features will compensate for the shortcomings of the model and the final performance will be comparable to the non-linear model that typically benefits less from feature interactions.

Similarly, both in statistics and ML, various re-sampling approaches are leveraged for a better understanding of the underlying distribution that generated the available observations. Without sufficient training examples, ML models are unable to learn about certain classes, therefore using re-sampling for balancing out the datasets to eventually attain superior results on minority classes is essential. This is also a generally applicable conclusion, albeit the expected improvement in performance will be strictly dependent on the dataset.

VII. Conclusions

In this paper, we provided a systematic quantification of the influence of the design steps on the final performance of a wireless link quality classifier. Among others, we found that, for the chosen problem and dataset, the generation of synthetic features from the only available training feature $RSSI$, yields up to 6% higher accuracy and is able to discriminate the intermediate class up to 49% better. The choice of ML method

[8]Multilayer perceptron is a class of feed-forward neural networks, where information travels through a directed non-cyclic graph of computational units called perceptrons (artificial neurons).

has relatively smaller impact on final model performance with all the selected algorithm yielding accuracy between 94% and 95% and minority class is detected between 87% and 89%.

We also provided a first time evaluation of the impact of re-sampling on wireless link quality classification using ML. In the case of the chosen imbalanced dataset, by using standard re-sampling, the minority class was correctly detected in over 87% of the instances, yielding more than 25 percentage points increase in the performance and comes at a small decrease in accuracy that can be mitigated with more advanced re-sampling techniques.

ACKNOWLEDGMENT

This work was funded by the Slovenian Research Agency (Grant no. P2-0016 and J2-9232) and 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] P. Millan, C. Molina, E. Medina, D. Vega, R. Meseguer, B. Braem, and C. Blondia, "Time series analysis to predict link quality of wireless community networks," Computer Networks, vol. 93, no. 2, pp. 342–358, Dec. 2015.

[8] 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," Computer Communications, vol. 112, pp. 1–13, Nov. 2017.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] N. Baccour, A. Koubâa, M. B. Jamâa, D. Do Rosario, H. Youssef, M. Alves, and L. B. Becker, "Radiale: A framework for designing and assessing link quality estimators in wireless sensor networks," Ad Hoc Networks, vol. 9, no. 7, pp. 1165–1185, September 2011.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] U. Fayyad, G. P.-Shapiro, and P. Smyth, "From data mining to knowledge discovery in databases," AI magazine, vol. 17, no. 3, p. 37, 1996.

[22] M. Kulin, C. Fortuna, E. De Poorter, D. Deschrijver, and I. Moerman, "Data-driven design of intelligent wireless networks: An overview and tutorial," Sensors, vol. 16, no. 6, p. 790, 2016.

[23] 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.

[24] B. W. Yap, K. A. Rani, H. A. A. Rahman, S. Fong, Z. Khairudin, and N. N. Abdullah, "An application of oversampling, undersampling, bagging and boosting in handling imbalanced datasets," in Proceedings of the First International Conference on Advanced Data and Information Engineering (DaEng'13) - Lecture Notes in Electrical Engineering, December 2014, pp. 13–22.

[25] 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.

[26] J. M. Jerez, I. Molina, P. J. García-Laencina, E. Alba, N. Ribelles, M. Martín, and L. Franco, "Missing data imputation using statistical and machine learning methods in a real breast cancer problem," Artificial intelligence in medicine, vol. 50, no. 2, pp. 105–115, 2010.

[27] R. J. Little and D. B. Rubin, Statistical analysis with missing data. John Wiley & Sons, 2014, vol. 333.

[28] W. Castro, J. Oblitas, M. De-La-Torre, C. Cotrina, K. Bazn, and H. Avila-George, "Classification of cape gooseberry fruit according to its level of ripeness using machine learning techniques and different color spaces," IEEE Access, vol. 7, pp. 27 389–27 400, 2019.

[29] R. A. Alshinina and K. M. Elleithy, "A highly accurate deep learning based approach for developing wireless sensor network middleware," IEEE Access, vol. 6, pp. 29 885–29 898, 2018.

[30] T. Liu and A. E. Cerpa, "TALENT: temporal adaptive link estimator with no training," in Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems, Toronto, Canada, November 2012, pp. 253–266.

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

[32] L. Lusa et al., "Smote for high-dimensional class-imbalanced data," BMC bioinformatics, vol. 14, no. 1, p. 106, 2013.

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

[34] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg et al., "Scikit-learn: Machine learning in Python," Journal of machine learning research, vol. 12, no. Oct, pp. 2825–2830, 2011.

[35] R. Fonseca, O. Gnawali, K. Jamieson, and P. Levis, "Four-bit wireless link estimation," in ACM SIGCOMM Sixth Workshop on Hot Topics in Networks (HotNets-VI'07), Atlanta, Georgia, 14-15 November 2007.

Chapter 4

Classifying Imbalanced Wireless Link Data

In the previous chapter, we presented the importance of data pre-processing steps and their contribution to the classification performance of data-driven LQE models. One of the most common problems with machine learning algorithms is an imbalance between classes in the available dataset. If ignored, this problem leads to unfair classification biased towards a majority class.

This chapter extends the investigation in data pre-processing steps by focusing on the influence of design steps on per-class classification performance and its fairness toward minority classes. In literature, LQE classifiers utilize two (binary), three, four and up to seven target classes with distinct wireless link quality. The performance of those classifiers is evaluated using various, but often only single averaged metrics. In our research, we show that relying on a single metric, such as accuracy, can be misleading in presenting fairness of classification.

We present a novel tree-based classifier. The classifier exhibits high performance and fairly classifies the minority class while incurring low training costs. We compare the tree-based model to a non-linear multi-layer perceptron model and two linear models, namely logistic regression and support vector machine. As opposed to other studies, we evaluate their results using five different performance metrics, *i.e.* accuracy, precision, recall, F1-score, and confusion matrix.

The comparison shows that non-linear models perform slightly better than linear models in general. The new tree-based model shows the best trade-off considering F1-score, training time and fairness of classification. Relying solely on a single aggregated metric, such as accuracy, can hide poor performance and discrimination toward minority classes. In our case, we demonstrate that it is possible to improve the performance on minority classes by over 40% through feature selection and by over 20% through resampling strategies, leading to notably more fair classification results.

From the hypotheses outlined in Section 1.1, this chapter partially addresses and confirms hypotheses **H1** and **H3**:

H1 Wireless link quality can be efficiently and accurately estimated using machine learning approaches.

H3 Data pre-processing and algorithm parametrization have significant impact on the performance of wireless link quality estimation and wireless link anomaly detection.

Similar to the state of the art presented in Chapter 2 and ML-based LQE model design in Chapter 3, this chapter also confirms hypothesis **H1**. With the emphasis on the fairness of classification, we show that machine learning approaches can achieve high performance, high accuracy and low training costs in successfully estimating wireless link quality even under a more fair classification premise.

This chapter emphasises fairness of classification and demonstrates how significant is the influence of data pre-processing steps on the final performance, which confirms hypothesis **H3**.

As to the contributions outlined in Chapter 1.3, this chapter represents parts of contributions **C2** and **C3**. For contribution **C2**, this chapter provides an analysis of design decisions to improve the fairness of classification for the minority classes. For contribution **C3**, we present our novel supervised tree-based classifier trained on cross-layer data obtained from a real-world wireless network testbed, and evaluate its performance with standard classification metrics.

The publication included in this chapter is:

- G. Cerar, H. Yetgin, M. Mohorčič and C. Fortuna, *Learning to Fairly Classify the Quality of Wireless Links*, 16th Conference on Wireless On-demand Network Systems and Services (WONS 2021).

2021 16th Annual Conference on Wireless On-demand Network Systems and Services (WONS)

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

2021 16th Annual Conference on Wireless On-demand Network Systems and Services Conference (WONS) | 978-3-903176-35-5/21/$31.00 ©2021IEEE 10.23919/WONS51326.2021.9415540

***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

2021 16th Annual Conference on Wireless On-demand Network Systems and Services (WONS)

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

2021 16th Annual Conference on Wireless On-demand Network Systems and Services (WONS)

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].

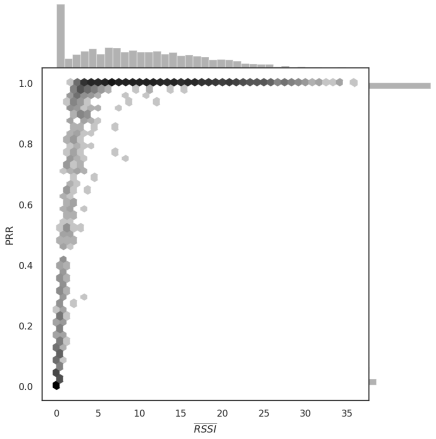


Fig. 1: PRR and average $\overline{\text{RSSI}}$ relationship for Rutgers trace-set (log-scale).

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

2021 16th Annual Conference on Wireless On-demand Network Systems and Services (WONS)

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.

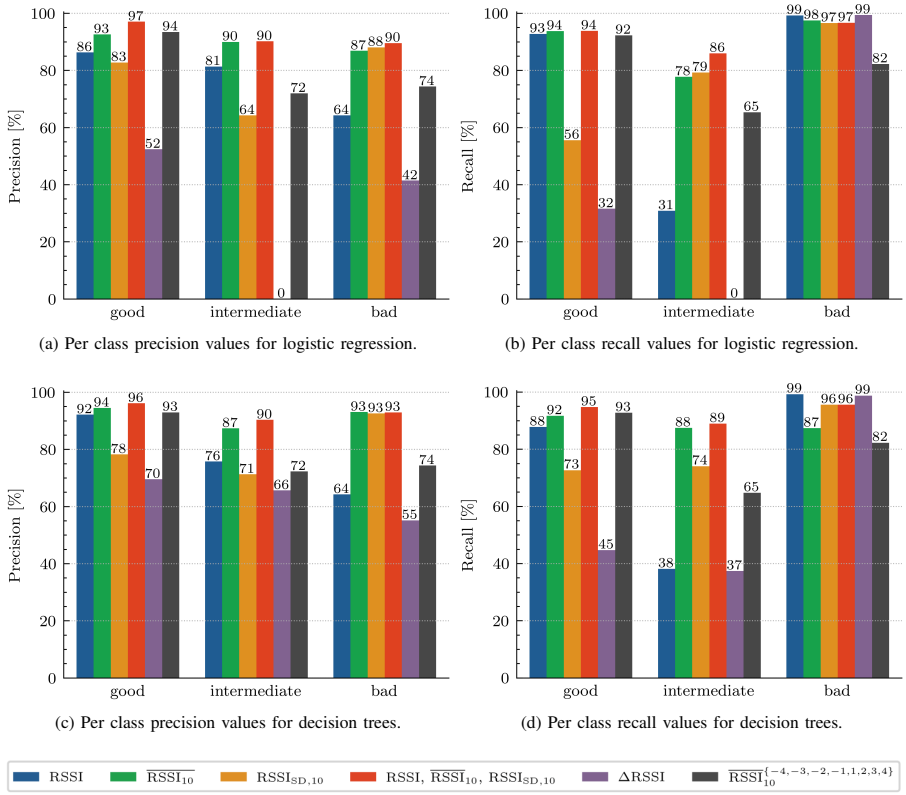
The results show that using only $RSSI$ yields 74% *accuracy* for the linear model and 75% for the non-linear one as per the first line corresponding to each algorithm in Table II, while the F1 scores are about 70% and 72%, respectively, confirming the fact that *accuracy* overestimates the performance 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

2021 16th Annual Conference on Wireless On-demand Network Systems and Services (WONS)

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

2021 16th Annual Conference on Wireless On-demand Network Systems and Services (WONS)

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.

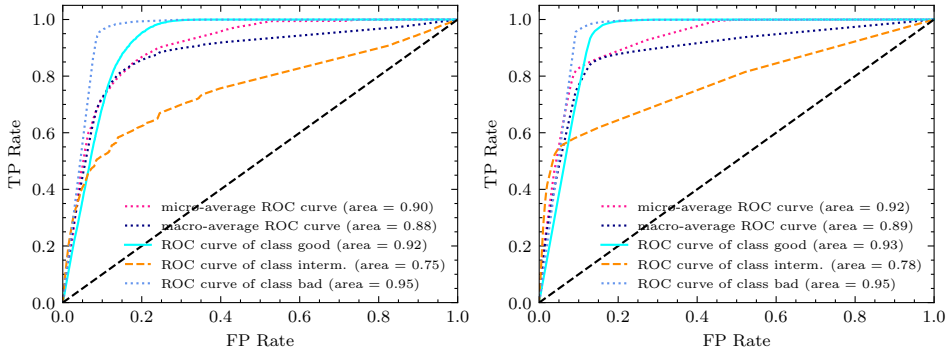
The analyses in this section demonstrate that when optimizing the overall performance of the classifier without considering per-class fairness, the best results are obtained on the actual dataset resulted in 97% accuracy. However, in this case the performance of recognizing the minority classes, namely

ISBN 978-3-903176-35-5 © IFIP

2021 16th Annual Conference on Wireless On-demand Network Systems and Services (WONS)

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

2021 16th Annual Conference on Wireless On-demand Network Systems and Services (WONS)

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.

ISBN 978-3-903176-35-5 © IFIP

Part II

Anomaly Detection in Wireless Links

Chapter 5

Detecting Anomalous Wireless Links in IoT Networks

In the previous chapters (*i.e.* 2, 3 and 4) we presented the first of two applications of link state information studied in depth in this thesis, namely link quality estimation. In this chapter, we present anomaly detection, which is a different application of link state information. Both applications evaluate wireless links to make decisions. However, the difference lies in their goals. On one hand, link quality estimation assesses wireless links to optimize data transmission in terms of throughput, latency, or reliability. On the other hand, anomaly detection evaluates wireless links to detect malfunctions that directly or indirectly affect wireless links to make monitoring and maintenance more efficient.

Large-scale automated and semi-automated deployments of wireless devices are becoming a new norm with the advent of IoT technologies. However, once deployed, IoT nodes become part of the operational infrastructure that needs to be maintained and serviced similarly to any other infrastructure. To keep the maintenance cost sustainable and avoid extensive downtime in case of failures, it thus becomes essential to monitor the devices and wireless links for any malfunctions and anomalies, especially because certain anomalies or symptoms can be detected directly or indirectly through the quality of wireless links.

This chapter defines four distinctive types of anomalies that can appear on wireless links, their possible causes and symptoms. Ordered by their duration, we distinguish very short in duration referred to as instantaneous anomalies (InstaD), sudden degradation with recovery (SuddenR), sudden degradation without recovery (SuddenD), and finally slow degradation (SlowD), where wireless links degrade at a slow pace.

Although each anomaly in the wireless network appears unique, it follows one of the four basic patterns, and recognizing patterns in data is where machine learning excels. We evaluate several different machine learning approaches to detect those anomalous patterns that do not comply with typical wireless link behaviour. In our evaluation, we consider supervised as well as unsupervised approaches. Ideally, unsupervised approaches with sufficiently high accuracy are preferred since they do not require preparation of an annotated training dataset that proves to be a labour-intensive and tedious task.

Following the methodology from Chapters 3 and 4, we investigate feature engineering, where we experiment with four different data representations to maximize the detection rate. We evaluate raw time series, aggregated (summarized) features, histogram features, and frequency domain representation as input. We also introduce encoded representation of input features using deep learning autoencoders, where the input transformation of the autoencoder acts as a generalized dimensionality reduction process.

In the performance analysis, we explain the ML algorithm decision process with explainable AI approaches. We show the analysis for each type of anomaly independently

and present some of the detection limitations. None of the manually generated features dominate in terms of performance, however the use of automatically generated encoded representations shows an improvement in F1 score of up to 40% compared to non-encoded representations. Supervised models achieve near-perfect performance, while unsupervised approaches, such as the one-class support vector machine (OC-SVM), perform best with an average F1 score of 99% for detecting SuddenD, 95% for detecting SuddenR, 93% for InstaD, and 95% for SlowD.

From the hypotheses outlined in Chapter 1.1, in this chapter, we addressed and partially confirmed hypotheses **H2**, **H3** and **H4**:

H2 Wireless link anomalies can be effectively and reliably detected using machine learning approaches which can outperform rule-based approaches.

H3 Data pre-processing and algorithm parametrization have significant impact on the performance of wireless link quality estimation and wireless link anomaly detection.

H4 Wireless link anomaly detection based on traditional machine learning approaches can be further improved by using deep learning neural networks in the pre-processing step.

This chapter shows that anomalies in wireless links can effectively and reliably be detected using machine learning algorithms. Furthermore, it demonstrates that machine learning approaches outperform rule-based approaches, and the top-performing model of supervised and unsupervised groups surpass 90% F1 score mark, which confirms hypothesis **H2**.

The use of careful pre-processing and algorithm parametrization shows significant influence on the ML performance for anomaly detection. As demonstrated, combining traditional machine learning and deep learning as a pre-processing stage to produce an encoded representation of input data shows promising results. We observe a boost in performance of up to 40% in most cases for supervised and unsupervised algorithms alike, which confirms hypotheses **H3** and **H4**.

As to the contributions outlined in Chapter 1.3, this chapter represents parts of contributions **C4** and **C5**. We compared the performance of novel supervised and unsupervised anomaly detection classifiers based on cross-layer data obtained from real-world wireless network testbeds (**C4**). In the process, we experimented with several data representations and pre-procesing steps from KDP to achieve better detection rate. The evaluation and comparison were done through standard classification metrics. For contribution **C5**, we demonstrate performance enhancements for anomaly detection by utilizing autoencoders for encoding input features. Autoencoders' feature space reduction and denoising capabilities were able to improve detection rate for up to 40% for selected ML models.

The publication included in this Chapter is:

- G. Cerar, H. Yetgin, B. Bertalanic and C. Fortuna, *Learning to Detect Anomalous Wireless Links in IoT Networks*, in IEEE Access, vol. 8, pp. 212130-212155, 2020, doi: 10.1109/ACCESS.2020.3039333.

IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received November 3, 2020, accepted November 16, 2020, date of publication November 19, 2020,
date of current version December 8, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3039333

Learning to Detect Anomalous Wireless Links in IoT Networks

GREGOR CERAR[1,2], (Graduate Student Member, IEEE), HALIL YETGIN[1,3], (Member, IEEE), BLAZ BERTALANIC[1,4], (Member, IEEE), AND CAROLINA FORTUNA[1]

[1]Department of Communication Systems, Jožef Stefan Institute, SI-1000 Ljubljana, Slovenia
[2]Jožef Stefan International Postgraduate School, SI-1000 Ljubljana, Slovenia
[3]Department of Electrical and Electronics Engineering, Bitlis Eren University, 13000 Bitlis, Turkey
[4]Faculty of Electrical Engineering, University of Ljubljana, 1000 Ljubljana, Slovenia

Corresponding author: Halil Yetgin (halil.yetgin@ijs.si)

This work was supported by the Slovenian Research Agency under Grant P2-0016 and Grant J2-9232.

ABSTRACT After decades of research, Internet of Things (IoT) is finally permeating real-life and helps improve the efficiency of infrastructures and processes as well as our health. As massive number of IoT devices are deployed, they naturally incurs great operational costs to ensure intended operations. To effectively handle such intended operations in massive IoT networks, automatic detection of malfunctioning, namely anomaly detection, becomes a critical but challenging task. In this paper, motivated by a real-world experimental IoT deployment, we introduce four types of wireless network anomalies that are identified at the link layer. We study the performance of threshold- and machine learning (ML)-based classifiers to automatically detect these anomalies. We examine the relative performance of three supervised and three unsupervised ML techniques on both non-encoded and encoded (autoencoder) feature representations. Our results demonstrate that; i) selected supervised approaches are able to detect anomalies with F1 scores of above 0.98, while unsupervised ones are also capable of detecting the said anomalies with F1 scores of, on average, 0.90, and ii) OC-SVM outperforms all the other unsupervised ML approaches reaching at F1 scores of 0.99 for SuddenD, 0.95 for SuddenR, 0.93 for InstaD and 0.95 for SlowD.

INDEX TERMS Anomaly detection, Internet of Things (IoT), machine learning (ML), wireless links, wireless networks.

I. INTRODUCTION

The Internet of Things (IoT) has received a plethora of attention from both industry and academia due to the market release of a variety of smart devices on a regular basis, e.g. the devices retrofitted in home appliances, wearables, healthcare, vehicles and industrial machinery, just to name a few [1]. To this end, extensive research efforts have been put forward for their active deployment and development to enable increasingly efficient and more automated operations in manufacturing, agriculture, transportation and healthcare, but also due to their massive economic contributions [2].

Valid business cases [3] and successful real-world large-scale IoT deployments are emerging as a way to improve existing business processes as well as enable new applications [2]. However, once the network of sensors is deployed, it becomes part of the operational infrastructure of a business, and needs to be maintained and serviced similar to any other infrastructure, such as legacy IT infrastructure, robots and machines just to name a few. Minimizing maintenance costs while ensuring the reliability of IoT network [4] becomes prohibitive when the number of sensors are in their thousands or tens of thousands. To efficiently manage such massive IoT networks, automatic IoT network monitoring [5] and malfunction detection [6] solutions that automatically report relevant malfunctions and filter them out without influencing the business process are required.

IoT network or node malfunctioning can also be referred to as network or node anomaly and to date, it has been defined in various ways, often from the perspective of monitored networking aspects. For instance, Sheth *et al.* [6] define and identify anomalies from the IEEE 802.11 physical layer perspective, namely, hidden terminal, capture effect, noise and signal strength variation anomalies, whereas Gupta *et al.* [7] define anomalies from multihop networking perspective with the aspects, such as black hole, sink hole, selective forwarding

The associate editor coordinating the review of this manuscript and approving it for publication was Celimuge Wu.

and flooding. Alipour *et al.* [8] define the anomalies from IEEE 802.11 link layer security perspective with the focus on aspects, such as injection test, deauthentication attack, disassociation attack, association flood and authentication flood. Generally speaking, anomaly detection research in IoT networks can be found in the form of intrusion, fraud and fault detection, system health monitoring, event detection in sensor networks and detecting ecosystem disturbances [9], where most studies mainly concerned with a certain type of anomaly within a specific scenario.

In this paper, motivated by a real-world experimental IoT deployment, we define four types of IoT anomalies that can be identified at the link layer, namely *sudden degradation*, *sudden degradation with recovery*, *instantaneous degradation* and *slow degradation*. Rather than focusing on the cause of an anomaly as realized in [6] and [7], we focus our attention on the observable symptoms of link measurements, namely the changes in the expected received signal. Based on the type of anomaly, we identify possible root causes that may be related to hardware, firmware and the channel, and develop models for automatically classifying the introduced anomalies. By accurately detecting these four types of anomalies, a wireless network operator is able to quickly and proactively detect issues within the operation of the network without waiting to be explicitly alerted by users. Proactively detecting and mitigating malfunctions can increase user satisfaction, reduce churn and ultimately show significant improvements in business KPIs. Additionally, the detected and classified anomaly type can aid technical staff with the well-informed decisions so as to diagnose and resolve the issues. For instance, *sudden degradation with recovery* is observed frequently after updating the firmware of devices in the network, which is highly likely related to the bugs of the firmware that prevent devices from working as intended and trigger the watchdog to reset. Therefore, discriminating between four of those types of anomalies and automatizing this process can speed up the real-time resolution of the network-related issues, in turn diminishing the allotted personnel and their efforts, and network-wide operational costs of mobile operators. The major contributions of this paper are as follows.

1) We define four types of anomalies that can appear on wireless links and are representative for narrowing down the causes and enabling more efficient mitigation. Driven by a real-world operational wireless infrastructure, for each of the defined anomalies we identify their symptoms from the application perspective and potential underlying causes.
2) We study the performance of standard manually-engineered features and a proposed autoencoder-based automatic feature generation approach, and show the performance improvement brought by the latter.
3) We also analyse the relative performance of three supervised and three unsupervised ML techniques. More explicitly, we consider regression-based, tree-based and kernel-based methods as part of our supervised techniques, while nearest neighbours, tree- and kernel-based methods are leveraged as their unsupervised counterpart techniques.

Additionally, minor contributions are outlined as follows:

1) Based on the gained knowledge while operating the LOG-a-TEC wireless experimentation testbed [10], we provide an analysis on real-world operational measurements that further stresses the need for automated anomaly detection in massive IoT networks.
2) We produce a publicly available anomaly detection tool-set[1] including entire procedures, e.g., anomaly injection into trace-sets, feature generation out of data representations, and model training and development.

This paper is structured as follows. Section II summarizes the related work and Section III presents an analysis of the real-world testbed measurements motivating our contributions, while Section IV introduces the four types of IoT network anomalies. Then, Section V elaborates on various data representations that can be used to generate features for training the proposed ML models, whereas Section VI discusses the threshold-based approach as well as the selected supervised and unsupervised ML techniques. Section VII describes the relevant methodological and experimental details, while Section VIII provides thorough analyses of the results and discusses the limitations. Finally, Section IX concludes the paper.

II. RELATED WORK

We provide related work to the main contributions of this paper as follows. First, we discuss related works that define anomalies in wireless and IoT networks, then we stress on the use of autoencoders for improving various aspects of wireless networks including anomaly detection, and finally, we focus on ML models that support for improved operations of wireless networks.

A. ANOMALY DEFINITIONS IN WIRELESS NETWORKS

Generally speaking, *an anomaly* is defined as an outlier, a distant object, an exception, a surprise, an aberration or a peculiarity, depending on the domain, research community and specific application scenario [9], [11]–[15]. A widely used classification of anomalies, including in wireless sensor network research is provided in [9], [16], where three classes of anomalies are defined based on their nature; point anomalies, contextual anomalies and collective anomalies. In [14], Gupta *et al.* classify relevant studies on outlier detection for time series data, one of which is the point outlier as defined in [9], and others are subsequence outliers, global and local outliers. More recently, Lavin and Ahmad *et al.* [17] introduce a benchmark for anomaly detection, and target mainly at cloud networks and associated services, where they provide reference datasets to be used when evaluating the performance of anomaly detection algorithms. While they do

[1]Script for the design and development of anomaly detection models: https://gist.github.com/gcerar/0b03e55f41147a7b7230f45d1f1209d6

not specifically define the type of anomalies, their benchmark datasets include several anomalies.

Due to the spatio-temporal nature of wireless sensor network monitoring and data collection, Jurdak *et al.* [18] introduce temporal, spatial and spatio-temporal anomalies as well as node, network and data anomalies, followed by even finer grained anomalies, such as node resets, node failures, etc. A number of studies then introduce more focused and application specific anomalies. For instance, Sheth *et al.* [6] define and identify anomalies from the IEEE 802.11 physical layer perspective namely; hidden terminal, capture effect, noise and signal strength variation anomalies. Moreover, Gupta *et al.* [7] define anomalies with the aspects of multihop networking, such as black hole, sink hole, selective forwarding and flooding, whereas Alipour *et al.* [8] define anomalies from IEEE 802.11 link layer security aspects, such as injection test, deauthentication attack, disassociation attack, association flood and authentication flood. For further details, motivated readers are referred to [18] for the diagnosis and detection of wireless network anomalies.

B. AUTOENCODERS FOR IMPROVING WIRELESS NETWORK OPERATIONS AND ANOMALY DETECTION

With the advent of deep learning, one class of techniques belonging to this class of ML, referred to as autoencoders, has been proven to be particularly useful at performing automatic feature engineering also for time series data [19]. Autoencoders attempts to learn a lossless compression of the data and the code resulting from that compression represents a superior feature set.

Generally in wireless, autoencoders have been successfully applied by [20] and their subsequent works, such as [21] to accurately reconstruct physical layer signals and [22] signal denoising for more accurate localization. For anomaly detection in wireless and IoT networks, Wang *et al.* [23] proposed autoencoders for more accurate identification of faulty parts of WSNs, as well as faulty antennas in antenna arrays, whereas Shahid *et al.* [24] and Chen *et al.* [25] proposed autoencoders for identifying anomalies in wireless and IoT networks based on transport layer traces, and recently, Yin *et al.* [26] proposed recurrent autoencoders for time series anomaly detection for IoT networks. However, they used a synthetic dataset with metrics derived from several Yahoo services. Unlike the state-of-the-art, this work proposes autoencoders as an automatic feature generation method for link layer anomaly detection and uses a real-world wireless dataset in which the introduced four types of anomalies are synthetically injected.

C. ML TECHNIQUES FOR WIRELESS AND IoT NETWORK ANOMALY DETECTION

In the literature, it is often a good practice that when a ML solution to a specific problem is considered, several counterpart ML models are evaluated against each other for performance analyses. For instance, Kieu *et al.* [19] compare the performance of ten different ML techniques, such as Support Vector Machines, Local Outlier Factor, Isolation Forest, just to name a few, on six different datasets that are suitable for anomaly detection.

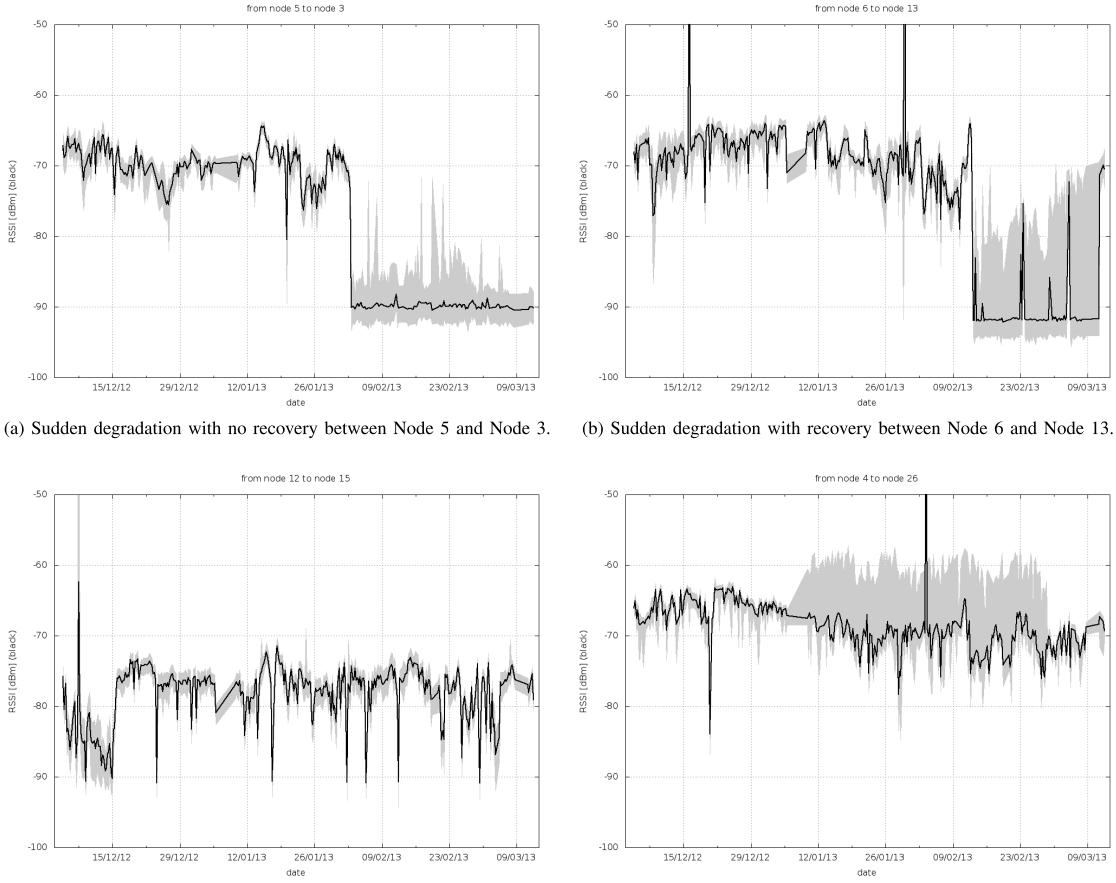
With respect to wireless and IoT network anomalies, Thing [27] evaluate the relative performance of four deep learning and one decision tree models for anomaly detection and attack classification in IEEE 802.11 networks, whereas Chen *et al.* [25] evaluate the relative performance of principal component analysis, standard and convolutional autoencoder for detecting anomalies in transport layer traces, i.e., TCP, UDP and ICMP of wireless networks. Moreover, Ran *et al.* [28] evaluate the relative performance of their proposed semi-supervised approach of IEEE.802.11 anomaly detection, and similarly Salem *et al.* [29] evaluate the relative performance of five ML techniques, i.e., SVM, decision trees (J48), logistic regression, Naïve Bayes, and Decision Table for anomaly detection in WSNs. Additionally, the previous authors [30] also evaluate the performance of their proposed algorithm against selected three ML techniques, namely linear regression, additive regression, and J48 decision tree for anomaly detection in WSNs. However, in most of the ML-based network anomaly detection research discussed in this section as well as in [31] provide only limited relative performance evaluation results. To the best of our knowledge, this paper is the first attempt to provide relative comparisons between three supervised and three unsupervised ML techniques based on various data representations and their encoded counterpart features.

III. MOTIVATION

Our lab runs the LOG-a-TEC [2] testbed that has empowered wireless experimentation for more than ten years. The first version of the testbed comprised of our custom embedded platform [32] was mounted on public light poles in a small municipality of Slovenia [33]. It included more than fifty nodes, most of which were situated in hard-to-reach locations. A sensor management system [10] is used to keep the record of each node for its hardware and software versions, configurations, and locations. This system also performs a number of management and diagnosis related tasks to monitor the operation of the devices.

Over time, the users of the testbed had difficulties in reaching some of the nodes or noticed unexplainable measurements collected during their testbed experimentation. For instance, the transceivers on some of the nodes were degraded significantly for their receiver sensitivity and transmit power performances, and in some cases to such a degree that they became inoperative. As depicted in Figure 1a, third node (ID-3) sensed transmissions from fifth node with received signal strength indicator (RSSI) of about −70 [dBm] on average till 2nd February of 2013. Following that, either fifth node's transmit power or third node's receiver sensitivity was degraded significantly, which was reduced to about −90 [dBm] on average. After investing a good amount of time

[2]LOG-a-TEC testbed with sensor platforms `http://log-a-tec.eu`



(a) Sudden degradation with no recovery between Node 5 and Node 3.

(b) Sudden degradation with recovery between Node 6 and Node 13.

(c) Spike-like instantaneous degradation between Node 13 and Node 15.

(d) Slow degradation between Node 4 and Node 26.

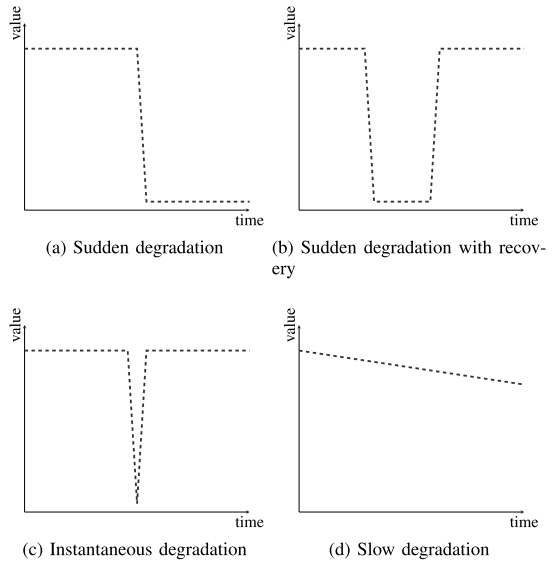
FIGURE 1. Anomalies observed in operational environment, where solid black lines represent average RSSI and greyed areas show maximum/minimum values.

and effort in understanding and reproducing the anomaly, the fifth node was diagnosed with a hardware failure, and it could only be restored to normal operation by replacing the integrated circuit for transceiver (TI CC2500).

Similarly, another anomaly type is experienced in Figure 1b with a sudden degradation and there were several recovery attempts between February 15th and March 9th 2013. In this particular case, we figured out that the sixth node was accidentally downgraded in February to an older version of the firmware that had a bug in the spectrum sensing code, which directly affected the operations of the sixth node and degraded its transmit power. Figure 1c presents several spike-like instantaneous degradation anomalies between nodes 12 and 15. We were not able to discover anything technically wrong with these respective nodes. Therefore, we assumed that these anomalies were probably due to weather and/or large objects moving around the radios, since these two devices were mounted in an industrial zone, where moving large trucks and massive long-term standing objects were not an uncommon occurrence, which can indeed incur spikes due to the instantaneous non-line-of-sight channels experienced. Finally, Figure 1d also exhibits two distinguishable rapid drops and climbs, but most importantly, on average, shows a slightly degrading performance in sensitivity and/or transmit power between nodes 4 and 26 after December 2012. We were not able to readily justify such behaviour of the device, but ageing of electronic components may induce such behaviour, which is a well-known issue [34].

IV. WIRELESS NETWORK ANOMALIES

Wireless networks are designed to exchange data between two communicating parties, e.g., video, voice and sensor measurements. As long as the network remains functional and is not interrupted, all the devices within the network are considered ordinarily operable. When the devices are compromised as exemplified in Section III, then a degradation in the service quality is experienced. The way how anomalies affect the user's service quality experience is stringently



(a) Sudden degradation (b) Sudden degradation with recovery

(c) Instantaneous degradation (d) Slow degradation

FIGURE 2. Visual representation of anomalies abbreviated as; a) SuddenD, b) SuddenR, c) InstaD, d) SlowD.

associated with the type of anomaly. Therefore, in this section, we introduce four types of anomalies that can be observed in communication links of wireless networks, which were mainly discovered in our evaluation of a real-world experimentation, as discussed in Section III: a) sudden degradation, b) sudden degradation with recovery, c) spike-like instantaneous degradation and d) slow degradation.

A. SUDDEN DEGRADATION (SuddenD)

The sudden degradation anomaly can be mathematically represented by a step function with decreasing slope, as depicted in Figure 2a. In our case, this represents a sudden persistent change in the state of a link. While this sudden change with an increasing slope is also possible in theory, typically it will only lead to a more reliable link, therefore they are not accounted as an anomaly.

Symptom: From the perspective of a user, services may become unavailable, offline and unreachable. From the perspective of a network, either the transmitter stops generating electromagnetic field or the receiver is unable to receive data.

Possible causes: Such sudden degradation can be induced by a transceiver failure as discussed in Section III and depicted in Figure 1a, a significant and sudden change in the position of one or both of the communicating parties leading them to remain disconnected, moving from line-of-sight to a non-line-of-sight environment with obstacles preserving electromagnetic shielding materials, and a significant hardware or software failure where built-in recovery mechanisms, such was watchdogs cannot be triggered.

B. SUDDEN DEGRADATION WITH RECOVERY (SuddenR)

The sudden degradation with recovery anomaly can be mathematically represented by a step function with decreasing slope, as depicted in Figure 2b. In this case, the state of a link suddenly changes, stays in the new state for a longer period of time and ultimately returns to the previous state. In sudden degradation with recovery, communication is interrupted for a certain period of time.

Symptom: From user's perspective, provided services may become sluggish and unavailable for a certain period of time and later resume back to their regular operations. From the perspective of the network, in the case of sudden degradation with recovery, either transmitter temporarily stops generating electromagnetic field or the receiver temporarily is unable to receive it.

Possible causes: This type of degradation can be caused by buffer congestion and software bug, as discussed in Section III and depicted in Figure 1b, where watchdog performs reboot after a certain timeout, a radio remaining in excessive active state and requiring recalibration, an obstacle blocking the communication for some time, and a signal jammer equipped on a military vehicle that is passing by.

C. INSTANTANEOUS DEGRADATION (InstaD)

The instantaneous degradation anomaly can be mathematically represented by a step function with steeply decreasing slope, forming a sudden spike, as depicted in Figure 2c. In this case, the state of the link changes suddenly, but instantaneously returns to its previous state. The instantaneous degradation anomaly may appear as an information loss.

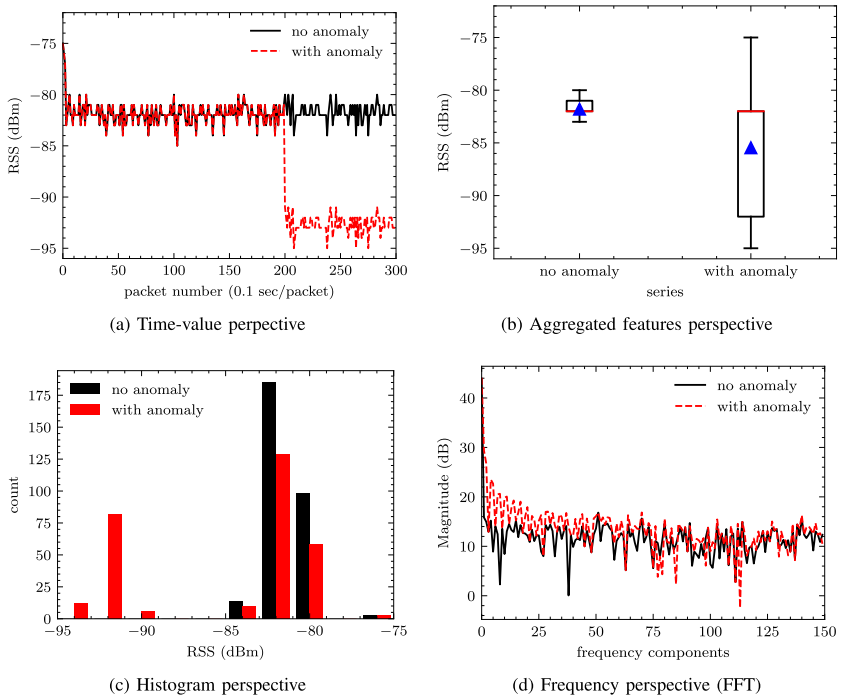
Symptoms: From user's perspective, a real-time service may experience instant lags, while other non-real-time services may work unaffected. From the perspective of the network, either transmitter experiences a deep fading instance or the receiver becomes unable to receive data due to an instant exposure to excessive noise or interference.

Possible causes: This type of degradation can be caused by an instant interference, collision, quantization errors, value reading errors or sudden saturations in the transceiver's electronic components, as discussed in Section III and depicted in Figure 1c, where anomaly can be stringently induced by the issues related to the propagation environment, such as an external device communicating on the same frequency, excessive background noise and multipath fading, just to name a few.

D. SLOW DEGRADATION (SlowD)

The slow degradation anomaly can be mathematically represented as a normalized linear function with slightly decreasing slope, as depicted in Figure 2d. In this case, the state of the link undertakes slight and unnoticeable changes for a longer period of time and it may never resume to its original state. The slow degradation anomaly may commence triggering information loss and interruptions after a certain amount of time.

Symptom: Slow degradation anomaly could go unnoticed for a very long time, where users may not even notice any difference in service quality immediately. When relevant thresholds are triggered, users commence experiencing deteriorated



(a) Time-value perpective (b) Aggregated features perspective

(c) Histogram perspective (d) Frequency perspective (FFT)

FIGURE 3. Distinct representations of the data for sudden degradation anomaly (SuddenD).

service quality. After employed compensation methods are exhausted (e. g., buffers, queues, bandwidth preservation strategies), communication may be interrupted and intended services may become unavailable. From the perspective of the network, either transmitter gradually stops generating sufficient electromagnetic field to satisfy a received signal-to-noise ratio threshold or the receiver is not able detect or collect enough electromagnetic radiation to decode the information, which can also be induced by the aging of electronic components.

Possible causes: This type of degradation may be caused by easier aging of electronic components in extreme working conditions (e. g., high moisture and heat) as it is discussed in Section III and depicted in Figure 1d, where it reflects a gradual but permanent impairment to the hardware or, slowly increasing obstacle such as a building being slowly built or vegetation growing.

V. DATA REPRESENTATION

Sections III and IV provided real-world anomaly examples and formalized wireless link anomalies, respectively. In the following, we provide five distinct ways to represent data that can be used as features while training the machine learning model.

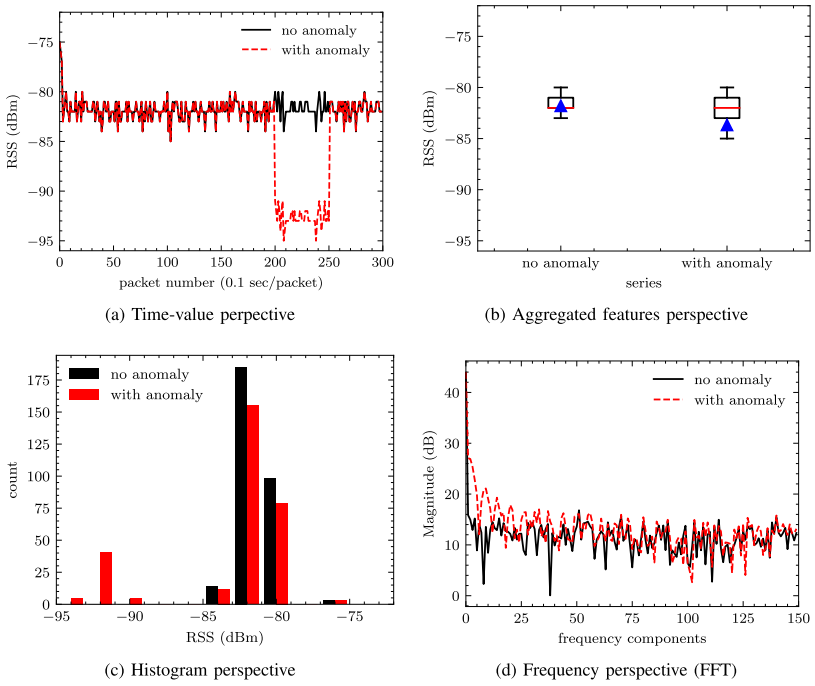
A. TIME-VALUE REPRESENTATION

The anomalies appearing in time series of RSSI values and in Figures 1 and 2 are recorded as raw time-ordered values, thus forming a time series. We refer to this time-ordered values as *time-value* representation. In Figures 3a, 4a, 5a and 6a, the time-value representation of an ordinary link is depicted with solid black lines and its anomaly injected counterpart, as per the definition from Section IV is depicted with dashed red lines.

However, through mathematical transformations, time series can be represented in other domains that, in some cases may be more suitable for the analysis of anomaly or pattern recognition. Motivated readers are referred to [35] for a comprehensive taxonomy of time series representation. In addition to the time-value representation, in this study, we also consider an aggregated representation, a histogram representation, a frequency domain representation and an automatically encoded representation.

B. AGGREGATED REPRESENTATION

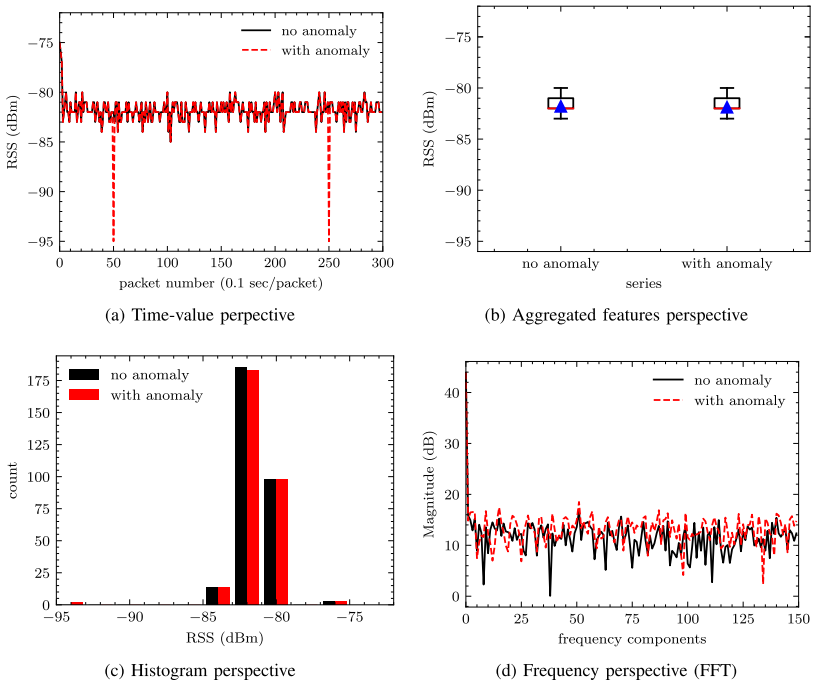
This representation contains seven statistical aggregates computed from the time-value representation, namely average, standard deviation, and all five quantile (Q) values, such as zeroth quantile (minimum), first quantile, second quantile (median), third quantile, and fourth quantile (maximum). This representation is depicted in Figures 3b, 4b, 5b and 6b for each anomaly type, where they present values belonging to middle quantiles (Q1-Q3) as a box shape, first quantile (Q0-Q1) and third quantile (Q2-Q3) are marked as separate whiskers on top and the bottom, median value (Q2) is shown as a red bar within the box shape (–), and finally, average is portrayed as a blue triangle shape (▲).



(a) Time-value perpective (b) Aggregated features perspective

(c) Histogram perspective (d) Frequency perspective (FFT)

FIGURE 4. **Distinct representations of the data for sudden degradation with recovery anomaly (SuddenR).**



(a) Time-value perpective (b) Aggregated features perspective

(c) Histogram perspective (d) Frequency perspective (FFT)

FIGURE 5. **Distinct representations of the data for spike-like instantaneous degradation anomaly (InstaD).**

C. HISTOGRAM REPRESENTATION

The histogram representation observed in Figures 3c, 4c, 5c and 6c is performed via splitting the range between (global) minimum and maximum values into ten equally-sized bins. More explicitly, this representation exhibits the percentage of values allotted in each bin.

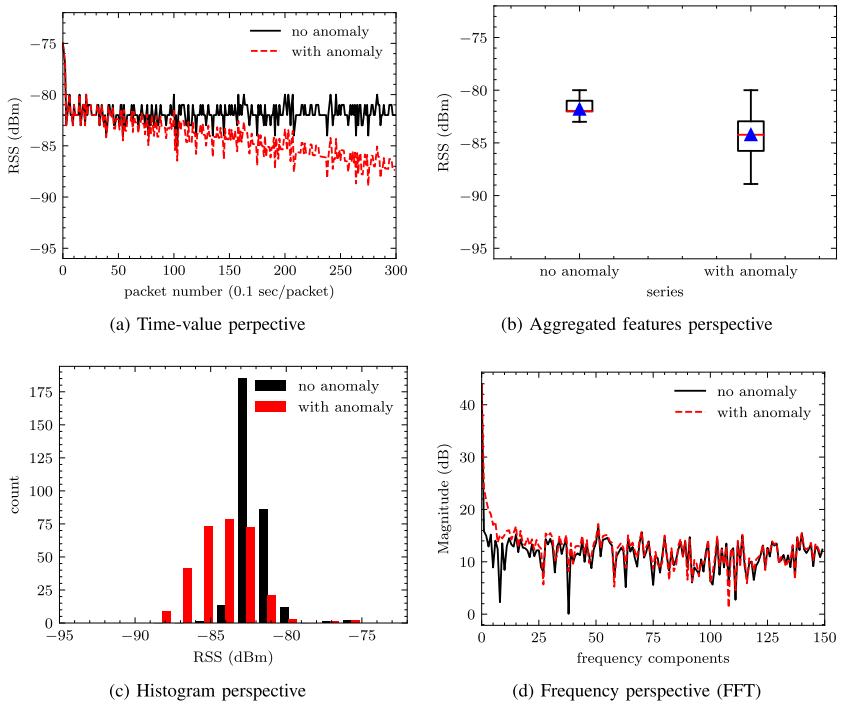


FIGURE 6. Distinct representations of the data for slow degradation anomaly (SlowD).

D. FFT REPRESENTATION

The frequency domain representation provided in Figures 3d, 4d, 5d and 6d utilizes absolute value of complex transformation, which is presented using log-scale for better contrasting "with anomaly" scenario against the "no anomaly" one.

E. ENCODED REPRESENTATION

A recent revolution of deep learning techniques, namely autoencoders, exhibits great performance returns in a diverse set of problems. To contrast against the above-mentioned traditional representations, we propose automatically generated encoded (autoencoder) representations for all anomaly types introduced in Section IV.

Autoencoders [16], [36], [37] are neural networks which are trained to generate a representation from the reduced encoding that is very similar compared its original input. The middle layer of an autoencoder is depicted with the purple circles in Figure 7 containing the reduced version of the input data and is referred to as a code $\mathbf{h}$ whose size is expected to be smaller than the size of the input data. As portrayed in Figure 7, an autoencoder is composed of two parts; i) an encoder function $\mathbf{h} = f(\mathbf{x})$, and ii) a decoder function producing a reconstruction $\hat{\mathbf{x}} = g(\mathbf{h})$. The autoencoders thus learn to include only the most useful signals from the input data, while mitigating the unnecessary signal noise.

An undercomplete autoencoder, where code size is smaller than input size, with nonlinear activation functions presents

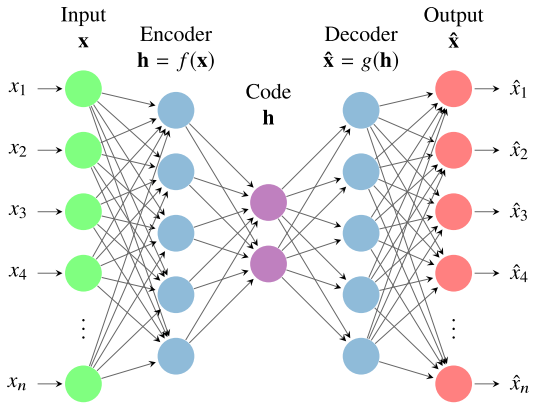
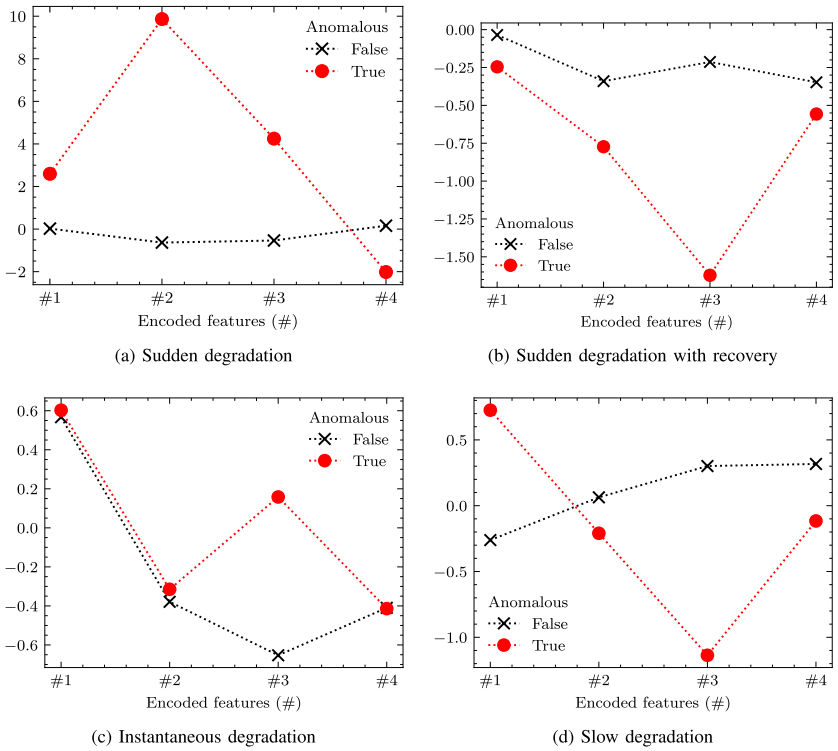


FIGURE 7. Illustration of autoencoder configuration during training process.

a generalized form of principal component analysis (PCA). Through the training process, the error between input $\mathbf{x}$ and output $\hat{\mathbf{x}}$ becomes negligible. Consequently, neural network learns a new representation of the input data, within a reduced feature-space. For example, in Figure 8a we transform time-value representation containing 300 dimensions into a newly encoded representation having only 4 dimensions. Figures 8a, 8b, 8c, and 8d present scenarios for a link with both; i) ordinary (non-anomalous) data, ii) anomaly injected (anomalous) data for SuddenD, SuddenR, InstaD



(a) Sudden degradation (b) Sudden degradation with recovery

(c) Instantaneous degradation (d) Slow degradation

FIGURE 8. Automatically generated features (code) exemplified for time-value representations.

and SlowD anomalies, respectively. Non-anomalous link is depicted with a solid black line, whereas anomalous link is marked with a dashed red line.

VI. APPROACHES FOR THE DETECTION OF ANOMALIES

Considering the link anomalies defined in Section IV and their corresponding representations depicted in Figures 3, 4, 5 and 6, it is clear that setting predefined thresholds for the investigated data would enable the detection of abnormal measurements and aid in treating them as an outlier. However, it has been proven that since fixed threshold-based approaches do not adapt to fluctuating behaviour of the data, selecting a threshold becomes consequential and thus may lead to poor performance, especially in real-time prediction applications [38]. On the contrary, adaptive and proactive approaches, such as deep learning neural network (DNN) and recurrent neural network (RNN) [38], can learn from regular patterns of the data and accurately identify abnormal behaviours to enable more accurate anomaly detection.

A. THRESHOLD BASED DETECTION

Considering Figure 2a, detecting SuddenD requires the diagnosis of steep falling slopes that do not recover for a relatively long, possibly predefined, period of time. Detecting SuddenR amounts to the identification of a sudden drop and later a boost in signal that resumes back to the original strength level within a predefined time window. SuddenR and InstaD are somewhat similar from application perspective. However, the distinction lies in the length of the time window at which the signal recovers back to its original levels within an instant of the time for InstaD. Detecting SlowD requires the diagnosis of a slowly but rather consistently falling slope for a relatively long, possibly predefined time window.

The time-value rules are a straightforward way to approach link-level anomaly detection. These rules may either be set based on an experienced arbitrary threshold or they can be identified using a theoretical or numerical method. However, as discussed in Section V, there are various possible ways to detect anomalies. For instance, it can be seen on Figures 3b, 4b and 6b that RSS distribution of an average healthy link is significantly different than the RSS distribution of the same link when anomaly is injected, which is readily distinguishable for SuddenD, SuddenR and SlowD anomalies at a glance. More explicitly, the spread of RSS for the anomaly injected link is wider, and its mean and median values are overwritten accordingly. Similar conclusions can be made for the respective histograms in Figures 3c, 4c and 6c. However, abnormal distributions in SlowD anomaly can only be detected with long-term observations. Moreover, sudden changes in time series can also

be detected in frequency domain, which in our case, are readily observed for SuddenD and SuddenR anomalies as larger magnitudes at lower frequencies in Figures 3b and 4b, respectively. Changes due to injected anomalies are almost indistinguishable in the case of InstaD and SlowD while leveraging frequency domain.

Details of the threshold strategy are provided in Section VIII. For time-value perspective, we consider D'Agostino-Pearson's normality statistical test [39], [40]. The test assesses whether certain set of points come from normal distribution or not. If the p value is below threshold, it is likely that the measurements do not come from normal distribution. Notice that Pearson's normality test is not sufficient condition for normality claims. Although, the approach may work fine for our limited line-of-sight scenario, it will not work for mobile or non line of sight scenario. For aggregated perspective, we consider for a link to have an anomaly two separate criteria. One criterion is based on the difference between mean and median values, which (if we assume normal distribution) are fairly close. The second criterion is how much can values deviate in standard deviation. Either of them has to be true for a link to be marked to have an anomaly. For histogram perspective, we define and arbitrary threshold. Anything below that is marked as an anomaly.

B. MACHINE LEARNING-BASED DETECTION

A ML model is expected to distinguish between anomalous and ordinary behaviours of a link, thus requires to solve a binary classification problem. There are two ways to train a ML model to identify such distinctions. The first one is based on a supervised training approach where all anomaly data are labelled, although in many practical applications, producing a reliable training dataset is expensive and it can inevitably cover only the type of anomalies that are present in the training dataset, which then cannot cope with the abnormal link behaviours in a comprehensive manner. For this reason, training a ML model in an unsupervised way is more practical, where learning from patterns of the overall link operations so as to distinguish the abnormal behaviours of a link from the anticipated behaviours is provoked, which is referred to as the automated detection of an outlier [41] or an anomaly [16] using ML models.

In addition to baseline threshold-based approach discussed in Section VI-A, we also consider three supervised and three unsupervised ML techniques as elaborated in the following sections.

1) SUPERVISED APPROACHES

To evaluate the performance of selected supervised ML techniques against each other and against the threshold-based approach, we opt for a set of candidate supervised approaches leveraging one representative technique from three different classes: i) Logistic Regression from Regression Analysis [42], ii) Random Forest from tree ensemble class [43] and iii) Support Vector Machines (SVM) from kernel-method class [43].

Logistic Regression [42] is a modified linear regression able to work on classification problems. In linear regression the goal is to fit a line to data samples and minimize loss. Similarly, logistic regression aims for fitting sigmoid function with the goal to minimize loss at predicting any two classes. Logistic regression also includes a generalized form suitable for high-dimensional input data and multi-class rather than binary classification.

Random Forests [44] is an ensemble method that uses a number of decision tree classifiers followed by a voting mechanisms to perform multi-class classification. The trees are learnt by randomly splitting a relatively large feature space into smaller subspaces. Each tree provides a class in which a specific data point falls into, the class corresponds to the ''vote'' of that tree. The final outcome of the classifier then uses a mechanism, such as majority voting to provide the final result.

Support Vector Machine [45] is a learning algorithm that belongs to the family of kernel methods. Roughly speaking, SVMs attempt to learn a hyperplane that best splits a set of data into two classes. The shape of the hyperplane depends on the type of kernel function selected for the algorithm. When the kernel function is linear, so is the learnt hyperplane. When non-linear kernels are chosen, for instance RBF kernel [46], then the hyperplane is non-linear therefore better suited to approximate or discriminate non-linear random variables.

2) UNSUPERVISED APPROACHES

The cost of producing labels for supervised learning is discussed in Section VI-B. As a countermeasure, we also consider a set of candidate unsupervised approaches for developing anomaly detection models [43], where we leverage one representative technique from three different classes: i) Local Outlier Factor from Nearest Neighbour (NN) class [43], ii) Isolation Forest from tree ensemble class [43] and iii) one-class Support Vector Machines (SVM) from kernel-method class [43].

Local Outlier Factor [47] belongs to the k-Nearest Neighbour (kNN) family of algorithms, which rely on the computation of the distance between data points of the feature space. The feature vectors with smaller distance are alike and thus clustered together. One drawback for this family of algorithms is that as the dimensionality of the training data grows, the computational complexity evolves exponentially. However, there have been attempts in circumventing this exponential complexity, e. g., Ball Tree.

Isolation Forest [48] belongs to tree-based ensemble methods, and works in a roughly similar way as Random Forests as described above. Essentially, it represents a Random Forest adapted so that it optimizes outlier detection rather than multi-class classification of majority of data it sees. Based on certain metrics and distinct criteria, the algorithm decides whether particular subspaces contain any abnormal samples, namely anomalies.

Support Vector Machine, as described at the end of supervised approaches, can also be used in an unsupervised mode

for anomaly detection. In fact, most ML techniques can be used in both supervised and unsupervised mode. With this one-class approach, the model is expected to distinguish data as negative or positive instances. Then, the model can learn the boundaries of the data so as to detect the points that lie outside the boundary exposed as anomalies or outliers.

VII. METHODOLOGY AND EXPERIMENTAL DETAILS

Before we proceed with the analysis of the relative performance of the wireless link anomaly detection approaches proposed in this paper, we provide relevant methodological and experimental details.

A. TRAINING DATASET GENERATION

For our experimental evaluation, we consider a real-world measurement dataset, i.e., Rutgers [49], which contains measurements from 29 nodes at 5 different noise levels and each record has 300 measurements. Although every link is measured at five different noise levels, wee consider each recording as a different link and we assume that there is no correlation. On this existing real-world dataset we synthetically inject the four types of anomalies proposed in this paper as follows. First, we only pick the links without packet loss. This reduces our dataset from 4 060 to 2 123 ($\approx$ 52%) of independent links. Second, by means of applying one anomaly type at a time, we randomly pick 33% of these links, at which the anomaly is injected according to guidelines in Table 1, while the remaining is left intact.

TABLE 1. Artificial anomaly injections for each anomaly scenario.

Type	Links	Affected	Appearance	Persistence
SuddenD	2 123	33% (700)	once, [200th, 280th]	for ∞
SuddenR			once, [25th, 275th]	for [5, 20]
InstaD			on $\approx$1% of a link	for 1 datapoint
SlowD			once, [1st, 20th]	for [150, 180]†

† RSSI(x, start) $\leftarrow$ RSSI(x) + min(0, $-$rand(0.5, 1.5) $\cdot$ (x $-$ start))

The suddenD anomaly, observed in Figure 2a, on the affected link appears arbitrarily between 200th and 280th packet and it persists indefinitely. In case of suddenR, observed in Figure 2b, anomaly applied on the link appears only once with a random start from 25th to 275th packet, where it persists for an arbitrary duration between 5 to 20 measurements. For InstaD of Figure 2c, the anomaly can appear anywhere in the entire series with 0.01 probability, which means that each anomaly on the affected link appears three times on average. Finally, SlowD anomaly of Figure 2d appears arbitrarily between 1st and 20th measurements, where it commences with a random degrading pace of duration between 150 and 280 packets. In a nutshell, anomaly injection details are provided in Table 1.

B. COMPUTING STANDARD AND ENCODED REPRESENTATIONS

Once anomalies are injected as specified in Table 1, we compute four different data representations described in Section V. The first one, namely time-value representation of Section V-a, converts each link into a single feature vector containing 300 features. The second one, the so-called aggregated feature, summarizes each link with 7 features, which are described in Section V-b. The third one, namely histogram feature discussed in Section V-c, defines ten equally spaced bins, which are then presented to a model as a feature vector containing 10 features. The forth one, namely frequency feature elaborated in Section V-d, gives the model a large feature vector of frequency-domain representation summing up to nearly 150 features. As we compute four representations for each of the four types of anomalies, we generate 16 candidate datasets.

Next, we also consider autoencoders for each anomaly scenario and each of the four standard representations. As any other deep neural network, autoencoder also requires many iterations of training. To produce credible results with autoencoder, we build the generic model in two steps. In the first step, we split the dataset into training and test groups with a 60:40 ratio, respectively. In the second step, when the weights of the autoencoder are converged, we perform an end-to-end evaluation on the test group. Relevant autoencoder configurations are provided in Table 2, where the layers and their required parameters are outlined for the encoder and the decoder. Although recent trends in DNNs go towards the use of convolutional layers, a convolution layer would make sense only in case of time-value and frequency perspective, due to their reasonable size and correlated neighbouring vector values. Therefore, our decision is to go with fully connected (dense) layers. For the activation part, we use batch normalization (BN) followed by Leaky Rectified Linear Unit (leaky ReLU, or LReLU) with $\alpha = 0.2$ coefficient for negative values. While plain ReLU is most widely used

TABLE 2. Autoencoder configurations.

Role	Layer	Notes
Encoder	Input(*)	
	Dense(128)	
	BN + LeakyReLU(α = 0.2)	
	Dense(64)	
	BN + LeakyReLU(α = 0.2)	
	Dense(32)	
	BN + LeakyReLU(α = 0.2)	
	Dense(4)	no activation
Decoder	Input(4)	
	Dense(32)	
	BN + LeakyReLU(α = 0.2)	
	Dense(64)	
	BN + LeakyReLU(α = 0.2)	
	Dense(128)	
	BN + LeakyReLU(α = 0.2)	
	Dense(*)	no activation

* input/output size depends on feature vector
- Implementation of autoencoders in TensorFlow/Keras is available at: https://gist.github.com/gcerar/5e4e53902493632a3cfb5cc06c3317b7

non-linear activation function, its leaky version has shown several benefits and minor overall improvements [50].

To produce the encoded representations, we feed the 16 datasets corresponding to the representation provided in Sections V-(a),(b),(c),(d) into the autoencoder, resulting in additional 16 candidate datasets. Therefore, to continue with the anomaly detection, we train both supervised and unsupervised ML models on a total of 32 datasets, 16 corresponding to the four standard representations of each anomaly and the other 16 corresponding to the encoded representations.

C. PERFORMING AUTOMATIC ANOMALY DETECTION

Next, we compute the performance of the threshold, three supervised and three unsupervised ML techniques described in Section VI on the 32 generated datasets corresponding to the proposed anomalies and representations. Each approaches' output is compared to a label to identify whether the link actually contains anomalies or not.

1) THRESHOLD APPROACH

Descriptive details of leveraging certain thresholds for each anomaly can be found in Section VI-A. The utilized experimental threshold parameters are listed in Table 3. The threshold for the time-series representation that uses the D'Agostino-Pearson's normality statistical test [39], [40] is $p < 10^{-3}$. The threshold for the aggregated representation assumes the absolute difference between mean and median is higher than $3dB$ or that the double of the standard deviation is higher than $2.5dB$. The threshold for the histogram representation is set at $RSSI < -85dBm$ while threshold selection for the FFT and encoded representations were infeasible to find using our trial-and-error approach. The differences in the FFT representation are not easily visible or detectable using simple methods while the encoded representations cannot be easily interpreted, therefore also deriving an appropriate threshold is not possible.

TABLE 3. Predetermined anomaly thresholds.

Features	Anomaly thresholds
Time-series	Normality test [39], [40], when $p < 10^{-3}$
Aggregated	(|mean - median| > 3 dB) **OR** (2·stdev > 2.5 dB)
Histogram	RSSI < -85 dBm

2) MACHINE LEARNING-BASED APPROACHES

For each of the six selected ML techniques, we use standard ML cross-validation.[3] We train the models using shuffled data split into training and test sets with a 80:20 ratio, respectively. Model is trained with the training set and evaluated using the test set in order to ensure credible results. We use standard metrics for evaluating classifiers: precision, recall and F1 score. Precision measures how many of the instances detected as class A actually belong to class A, expressed as; Precision $= \frac{TP}{TP+FP}$, whereas recall measures how many of the instances belonging to class A were actually detected, expressed as; Recall $= \frac{TP}{TP+FN}$, where TP, FP and FN stand for true positives, false positives and false negatives, respectively. F1 score is quantified by the harmonic mean of the precision and the recall, where larger values indicate better classifiers with balanced and higher precision and recall performances.

For each of the ML techniques selected in Section VI, Table 4 lists the respective implementations and parameters used in the experiments. For instance, for logistic regression we use the LogisticRegression implementation available in the Python Scikit Learn toolbox.[4] As the LogisticRegression implementation enables setting 12 different parameters that influence the final model, we generally select standard values that have been proven to work on large number of cases and datasets by the ML community. However, we identify selected parameters that should be optimized, such as the regularization strength C in this case. We search for the best configuration by adapting an array of possible values $C \in [10^{-3}, 10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}]$ and ultimately select the best performing regularization factor C among them. For instance, Figure 9 presents the scenario where a model is trained using LR on time-value representation for SuddenD anomalies and based on robust scaler. For this particular scenario, the best $F1$ score of this model is attained by means of setting C to any value that is larger than 1. For the results presented in the next sections, we only account for the best $F1$ scores obtained after searching for such near-optimal regularization parameter values.

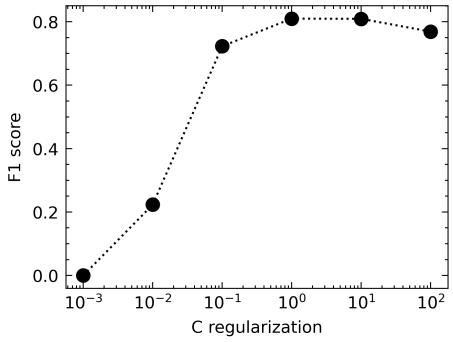


FIGURE 9. Regularization parameter (*C*) search for selecting the best performing model that is, for example, trained using LR on time-value representation for SuddenD anomalies and based on robust scaler.

The implementations chosen for the remaining algorithms also include over ten possible input parameters. For LOF, we vary the number of neighbours, algorithm and leaf size for finding the best performing model. For RForest and IForest, we vary the number of base estimators, whereas for SVM and

[3]Stratified K-Fold cross validation is implemented by using StratifiedKFold parameter in Python Scikit Learn toolbox https://scikit-learn.org/stable/

[4]https://scikit-learn.org/stable/

TABLE 4. ML techniques and their relevant parameters.

Approach	Technique	Implementation	Parameters and their range
Supervised	Logistic Regression (LR)	LogisticRegression from sklearn	penalty='l2', dual=False, tol=1e-4, C= (1e-3, 1e-2, 1e-1, 1.0, 10., 100.) fit_intercept=True, intercept_scaling=1, class_weight=None, solver='lbfgs', l1_ratio=None
	Random Forest (RForest)	BaggingClassifier from sklearn	base_estimator=None, n_estimators=[10, 20, 30, 40, 50, 70, 100], max_samples=1.0, max_features=1.0, oob_score=False, intercept_scaling=1,
	Support Vector Machine (SVM)	SVC from sklearn	C=(1e-3, 1e-2, 1e-1, 1.0, 10., 100.), kernel=('linear', 'rbf'), gamma=('auto', 'scale'), tol=1e-3, decision_function_shape='ovr', break_ties=False
Unsupervised	Local Outlier Factor (LOF)	LocalOutlierFactor from sklearn	n_neighbors=[5, 10, 20, 40, 50, 80], algorithm=['ball_tree', 'kd_tree', 'brute'], leaf_size=[10, 30, 50, 80], p=[1, 2] metric_params=None, contamination="auto",
	Isolation Forest (IForest)	IsolationForest from sklearn	n_estimators=[10, 20, 30, 40, 50, 70, 100], max_samples='auto', contamination='auto', max_features=1.0, bootstrap=False,
	Support Vector Machine (OC-SVM)	OneClassSVM from sklearn	nu=[0.10, 0.3, 0.5, 0.70, 0.90, 1.0], kernel=('linear', 'rbf'), gamma=('auto', 'scale'), coef0=0.0, tol=1e-3,

OC-SVM, we vary the regularization factor C, the kernel and the kernel coefficient *gamma* for the *rbf* kernel, respectively.

As some of the models are sensitive to scaling, we also consider training on data that is; i) not scaled, ii) scaled by using mean values, iii) scaled using mean and deviation, and iv) scaled using min-max. The entire procedure and parameters can be readily found and used in the existing public open source repository.[5] Six selected ML techniques with the associated parameter tuning are trained over the 32 datasets, totalling at more than 40,000 anomaly detection models.

VIII. EVALUATION

In this section, we evaluate the relative performance of various data representations discussed in Section V and of approaches discussed in Section VI for detecting four types of anomalies introduced in Section IV. The methodological and experimental details utilized for obtaining the results are elaborated in Section VII.

A. PERFORMANCE ANALYSES OF DATA REPRESENTATIONS

In this section, we first provide insight into how a model learns to classify by discussing the importance of various features resulting from the four manually generated and interpretable representations for discriminating the four types of anomalies defined in Section IV. Next, we discuss the influence of the five data representations, including those four manually generated ones and the automatically generated (autoencoder) one, as elaborated in Section V, on the performance of the learnt models. This entire subsection focuses on the influence of representations on the final models, while the influence of the ML approaches is analysed in Section VIII-B.

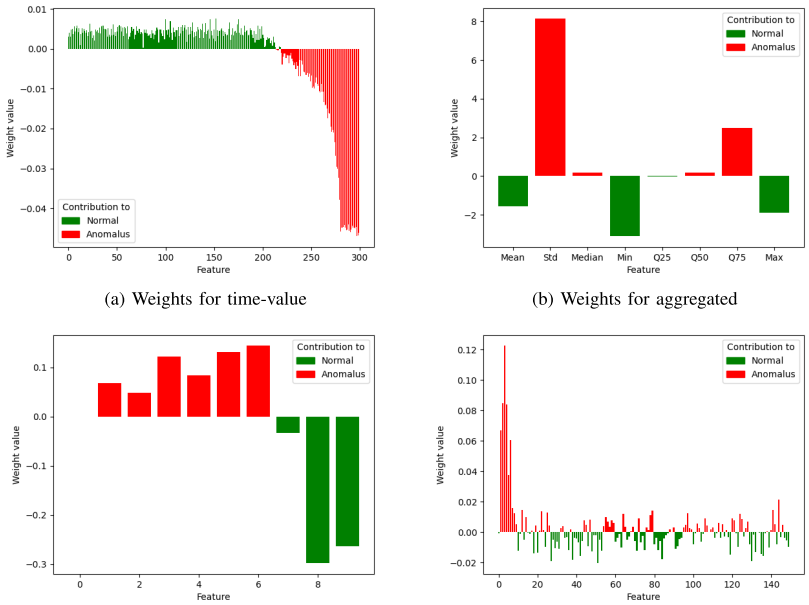
[5]Script for the design and development of anomaly detection models excluding data preprocessing is available at:
https://gist.github.com/gcerar/0b03e55f41147a7b7230f45d1f1209d6

1) ANALYSING THE DISCRIMINATIVE IMPORTANCE OF FEATURES

For analysing the discriminative power of the features in learning to classify the four anomaly types, we choose LR for its simplicity and reasonable tractability. As explaining the meaning of the automatically generated features is infeasible, we exclude them from this part of the analysis, without loss of generality.

Figures 10, 11, 12 and 13 depict the weights learnt by the LR on the representations discussed in Section V. Each set of figures corresponds to an anomaly type, namely SuddenD, SuddenR, InstaD and SlowD. In the above-referred figures, the green weights depict the features that are important for identifying normal links, whereas the red weights are important for detecting the anomalous links. Using these learnt features, it is possible to look at the LR as a linear function with as many variables as the length of the feature vector, e.g., 300 for time-values representation and 8 for the aggregated. Each point in the feature vector has its corresponding weight with which it is multiplied. When all multiplications (weight * variable(n)) are summed up, a positive or a negative value corresponding to one of the two classes are obtained, i.e., normal or anomalous links.

For the case of the *time-value representation* of the SuddenD anomaly from Figure 3a, it can be seen that the points depicted with red, mostly starting from somewhere after feature 200 play a more important role when making the decision on whether an input feature vector contains an anomaly or not. The reason why LR learns that these features are the most important ones can be explained from the way the SuddenD anomaly is injected in the training dataset. According to Table 1, SuddenD is injected randomly between packets 200 and 280. Therefore LR learns that those points are more discriminative for the anomalies. Simplistically, when multiplying the anomalous vector from Figure 3a with the weights in Figure 10a, and subsequently summing up, the result will become positive, and hence the input will be classified as



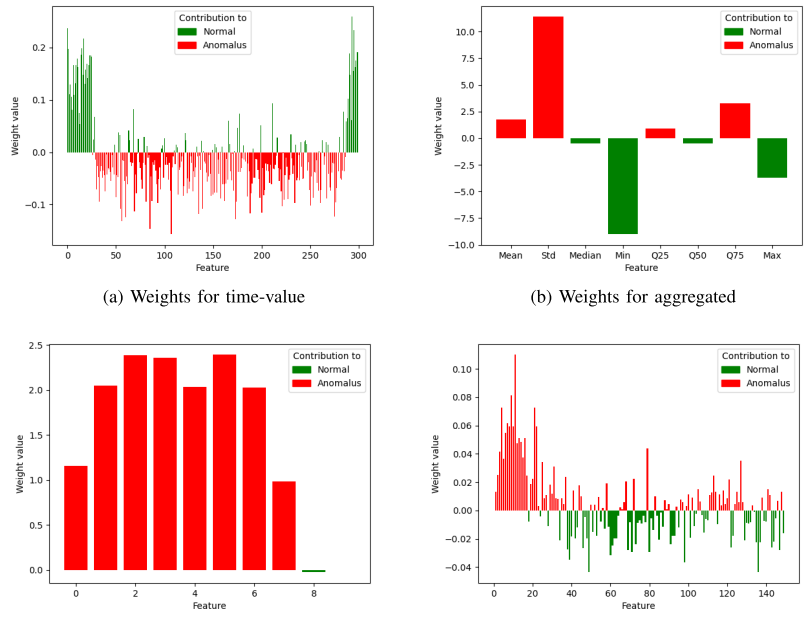
(a) Weights for time-value

(b) Weights for aggregated

(c) Weights for histogram

(d) Weights for frequency (FFT)

FIGURE 10. Learnt feature importance for distinct representations of the data for sudden degradation anomaly (SuddenD).



(a) Weights for time-value

(b) Weights for aggregated

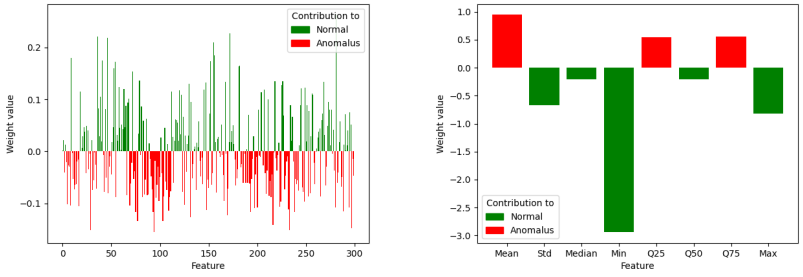
(c) Weights for histogram

(d) Weights for frequency (FFT)

FIGURE 11. Learnt feature importance for distinct representations of the data for sudden degradation with recovery anomaly (SuddenR).

anomaly. On the other hand, when the normal vector from Figure 3a is multiplied with the weights in Figure 10a, upon summing them up, the result will become negative, thus the vector will be classified as normal.

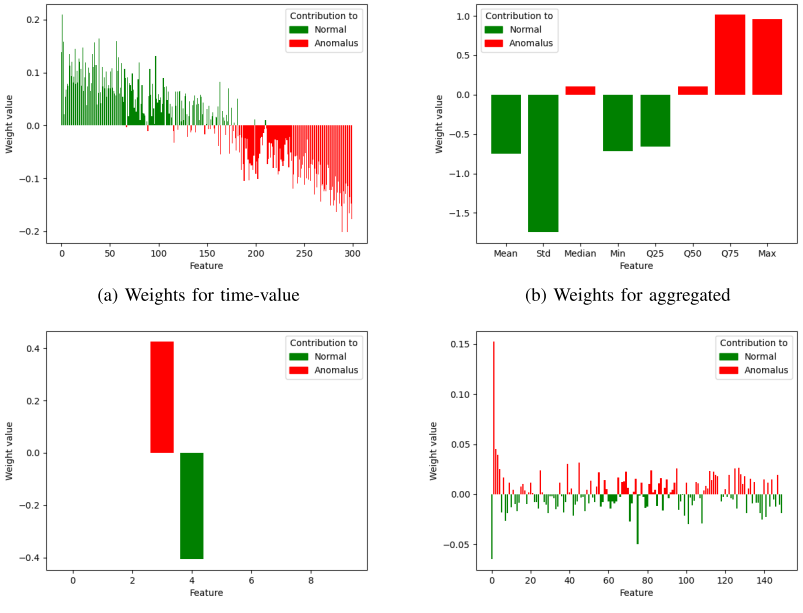
Similar discussions over time-value representations can be made for all the other anomalies. SuddenR anomaly is randomly injected between packets 25 and 275 of the time-value representation as per Table 1, and it can be seen from



(a) Weights for time-value (b) Weights for aggregated

(c) Weights for histogram (d) Weights for frequency (FFT)

FIGURE 12. Learnt feature importance for distinct representations of the data for spike-like instantaneous degradation anomaly (InstaD).



(a) Weights for time-value (b) Weights for aggregated

(c) Weights for histogram (d) Weights for frequency (FFT)

FIGURE 13. Learnt feature importance for distinct representations of the data for slow degradation anomaly (SlowD).

Figure 11a that the most important features for detecting the anomaly, represented with red, lie within this range. The importance of features for the spike anomaly that is quite random in nature and also occurs often in the data due to the nature of the wireless channel is depicted in Figure 12a. Finally, the importance of the features for detecting SlowD

TABLE 5. Performance of detecting sudden degradation (SuddenD) anomalies.

Approach	Technique	time-value features			aggregated features			histogram features			frequency domain		
		Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1
Baseline	Threshold (Tab. 3)	0.66	1.00	0.79^{1}	0.97	1.00	0.98^{1}	0.44	1.00	0.61^{1}	-	-	-
Supervised	LR	1.00	1.00	$\mathbf{1.00}^{1}$	1.00	1.00	$\mathbf{1.00}^{2}$	0.99	0.99	0.99^{1}	1.00	1.00	$\mathbf{1.00}^{1}$
	encoder + LR	1.00	1.00	$\mathbf{1.00}^{6}$	1.00	1.00	$\mathbf{1.00}^{3}$	1.00	1.00	$\mathbf{1.00}^{6}$	1.00	1.00	$\mathbf{1.00}^{1}$
	RForest	1.00	1.00	$\mathbf{1.00}^{1}$	0.99	0.99	0.99^{4}	0.99	1.00	$\mathbf{1.00}^{4}$	1.00	1.00	$\mathbf{1.00}^{1}$
	encoder + RForest	1.00	1.00	$\mathbf{1.00}^{1}$	1.00	1.00	$\mathbf{1.00}^{4}$	1.00	1.00	$\mathbf{1.00}^{4}$	1.00	1.00	$\mathbf{1.00}^{1}$
	SVM	1.00	1.00	$\mathbf{1.00}^{1,7}$	1.00	1.00	$\mathbf{1.00}^{4,7}$	0.99	1.00	$\mathbf{1.00}^{5,8}$	1.00	1.00	$\mathbf{1.00}^{1,7}$
	encoder + SVM	1.00	1.00	$\mathbf{1.00}^{1,8}$	1.00	1.00	$\mathbf{1.00}^{4,8}$	1.00	1.00	$\mathbf{1.00}^{4,7}$	1.00	1.00	$\mathbf{1.00}^{1,7}$
Unsupervised	LOF	0.36	0.53	0.43^{1}	0.51	0.38	0.43^{6}	0.88	0.67	0.76^{4}	1.00	0.20	0.33^{5}
	encoder + LOF	0.85	0.25	0.38^{3}	0.65	0.16	0.26^{4}	0.65	0.19	0.29^{1}	0.59	0.19	0.29^{2}
	IForest	0.98	0.48	0.64^{1}	0.90	0.77	0.83^{4}	0.91	0.60	0.72^{4}	0.89	0.47	0.61^{2}
	encoder + IForest	0.94	1.00	$\mathbf{0.97}^{3}$	0.89	0.86	$\mathbf{0.88}^{2}$	0.94	1.00	$\mathbf{0.97}^{3}$	0.94	0.99	$\mathbf{0.97}^{5}$
	OC-SVM	0.87	0.93	$\mathbf{0.90}^{4,8}$	0.81	0.86	$0.83^{1,8}$	0.94	0.99	$\mathbf{0.96}^{5,8}$	1.00	0.96	$\mathbf{0.98}^{2,7}$
	encoder + OC-SVM	1.00	0.84	$\mathbf{0.91}^{3,7}$	0.99	0.93	$\mathbf{0.96}^{5,7}$	1.00	0.84	$\mathbf{0.91}^{1,7}$	0.98	0.99	$\mathbf{0.99}^{3,7}$

is higher in the second half of the feature vector as depicted in Figure 13a since that's where the degradation becomes more evident.

Moving to *aggregated representations*, it can be seen from Figure 10b that standard deviation (Std) and the last quantile (Q75) are the most important features for detecting the anomaly, with minor contribution from the median and Q50. This is because standard deviation increases when SuddenD anomaly is present while the count of high RSSI values in the last quantile is smaller when this anomaly is present. Next, for SuddenR, the two main features remain the same as the shape is very similar to the SuddenD as can be seen in Figure 11b, albeit the duration differs leading to a more prominent influence of the mean for discrimination. For InstaD, that can be seen as a very narrow SuddenR randomly appearing on 1% of the link, Std looses importance while the mean and two quantiles become more predictive as depicted in Figure 12b. For SlowD, the model learns that features which inform about the slope that appears and increases, therefore Q75 counting high RSSI values and the maximum (max) are predictive. The median and Q50 that capture the intermediate values of the slowly increasing slope also add minor discriminative power, as portrayed in Figure 13b.

In the case of *histogram representation*, the first bins where *cumulated* low RSSI values corresponding to SuddenD, SuddenR and InstaD anomalies are the most important ones according to Figures 10c, 11c and 12c. For the case of SlowD presented in Figure 13c, one of the middle bins that capture intermediate values is the most discriminative while the other bins seem to not contribute to either class.

Finally, the importance of features in the case of *frequency representation* presents a similar line of reasoning as for the other representations. For SuddenD and SuddenR anomaly amplitudes at low frequencies that introduce a major shift in the mean are the most important features, as portrayed in Figures 10d and 11d. For InstaD there is no clear importance pattern as shown in Figure 12d, whereas for SlowD the feature amplitudes around 0 are the most prominent ones as illustrated in Figure 13d.

2) THE INFLUENCE OF THE REPRESENTATIONS ON THE PERFORMANCE OF THE LEARNT MODELS

The best performing results of the classification with respect to F1 score are presented in Table 5 for SuddenD, Table 6 for SuddenR, Table 7 for InstaD and Table 8 for SlowD. The first column of the tables lists the approach, the second column outlines the used ML techniques, while columns 3 to 6 list the results for time-value, aggregated, histogram and FFT representations, respectively.

The encoded representation introduced in Section V-e and employed according to the methodology in Section VII-B is inserted into the above-mentioned performance tables with the name of respective ML technique using the term "encoder". More precisely, referring to the rows corresponding to the ML technique, say IForest, the performance results are implemented for the four mentioned representations for the IForest ML technique. Additionally, at the row entitled "Encoder + IF", the numerical results refer to the IForest ML technique that is applied to the codes generated from the four representations, respectively. Finally, the superscripts identify the scaling methods utilized. The three highest F1 scores for supervised approaches and the three highest F1 scores for unsupervised approaches are delineated in bold font.

With respect to the data representations, from the results listed in Tables 5, 6, 7 and 8, two high level observations are outlined as follows.

- None of the four manually generated features clearly dominates the remaining ones in terms of anomaly detection performance.

TABLE 6. Performance of detecting sudden degradation with recovery (SuddenR) anomalies.

Approach	Technique	time-value features			aggregated features			histogram features			frequency domain		
		Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1
Baseline	Threshold (Tab. 3)	0.66	1.00	0.79^{1}	0.97	0.97	0.97^{1}	0.44	1.00	0.61^{1}	-	-	-
Supervised	LR	0.92	0.86	0.89^{3}	0.99	0.99	$\mathbf{0.99}^{3}$	1.00	0.98	$\mathbf{0.99}^{2}$	1.00	1.00	$\mathbf{1.00}^{2}$
	encoder + LR	0.99	0.98	$\mathbf{0.99}^{2}$	0.99	0.99	$\mathbf{0.99}^{6}$	1.00	0.99	$\mathbf{0.99}^{2}$	1.00	1.00	$\mathbf{1.00}^{2}$
	RForest	0.96	0.96	0.96^{3}	0.99	0.99	$\mathbf{0.99}^{2}$	1.00	0.99	$\mathbf{0.99}^{5}$	0.99	0.99	0.99^{5}
	encoder + RForest	0.99	0.99	$\mathbf{0.99}^{5}$	0.99	0.98	$\mathbf{0.99}^{6}$	1.00	0.99	$\mathbf{0.99}^{6}$	1.00	1.00	$\mathbf{1.00}^{1}$
	SVM	0.98	0.96	$0.97^{2,8}$	0.99	0.99	$\mathbf{0.99}^{5,8}$	0.99	0.99	$\mathbf{0.99}^{3,8}$	1.00	1.00	$\mathbf{1.00}^{1,7}$
	encoder + SVM	0.99	0.98	$\mathbf{0.99}^{5,7}$	0.99	0.99	$\mathbf{0.99}^{6,8}$	1.00	0.99	$\mathbf{0.99}^{2,7}$	1.00	1.00	$\mathbf{1.00}^{2,8}$
Unsupervised	LOF	0.88	0.99	$\mathbf{0.93}^{5}$	0.53	0.39	0.45^{6}	0.98	0.97	$\mathbf{0.98}^{2}$	1.00	0.39	0.56^{5}
	encoder + LOF	0.95	0.61	0.74^{1}	0.67	0.16	0.26^{2}	0.79	0.27	0.40^{3}	0.80	0.29	0.43^{1}
	IForest	0.48	0.20	0.28^{1}	0.95	0.62	0.75^{1}	0.99	0.26	0.41^{1}	0.94	0.49	0.64^{6}
	encoder + IForest	0.98	0.97	$\mathbf{0.98}^{5}$	0.93	0.98	$\mathbf{0.95}^{2}$	0.95	0.96	$\mathbf{0.95}^{5}$	0.97	0.97	$\mathbf{0.97}^{3}$
	OC-SVM	0.92	0.98	$\mathbf{0.95}^{2,8}$	0.98	0.82	$\mathbf{0.89}^{2,7}$	0.93	0.95	$\mathbf{0.94}^{5,8}$	1.00	0.83	$\mathbf{0.91}^{2,7}$
	encoder + OC-SVM	0.81	0.87	$0.84^{5,8}$	0.76	0.81	$\mathbf{0.78}^{2,8}$	0.74	0.96	$0.83^{5,7}$	0.73	0.98	$\mathbf{0.84}^{6,7}$

TABLE 7. Performance of detecting spike (InstaD) anomalies.

Approach	Technique	time-value features			aggregated features			histogram features			frequency domain		
		Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1
Baseline	Threshold (Tab. 3)	0.64	0.97	0.77^{1}	0.94	0.67	0.78^{1}	0.42	1.00	0.60^{1}	-	-	-
Supervised	LR	0.92	0.87	0.89^{1}	0.96	0.99	0.97^{2}	0.95	0.95	0.95^{1}	0.97	0.91	0.94^{1}
	encoder + LR	0.95	0.92	$\mathbf{0.94}^{4}$	0.98	0.98	$\mathbf{0.98}^{5}$	0.98	0.95	$\mathbf{0.97}^{3}$	0.98	0.94	$\mathbf{0.96}^{4}$
	RForest	0.98	0.86	0.91^{3}	0.98	0.96	0.97^{6}	0.98	0.95	$\mathbf{0.96}^{2}$	0.96	0.89	0.92^{1}
	encoder + RForest	0.96	0.91	$\mathbf{0.94}^{4}$	0.97	0.97	0.97^{6}	0.98	0.95	$\mathbf{0.96}^{3}$	0.97	0.93	$\mathbf{0.95}^{4}$
	SVM	0.96	0.91	$\mathbf{0.94}^{3,8}$	0.97	0.98	$\mathbf{0.98}^{5,8}$	0.97	0.96	$\mathbf{0.96}^{6,8}$	0.97	0.91	$0.94^{1,8}$
	encoder + SVM	0.95	0.93	$\mathbf{0.94}^{4,7}$	0.98	0.98	$\mathbf{0.98}^{4,7}$	0.98	0.95	$\mathbf{0.97}^{6,8}$	0.99	0.93	$\mathbf{0.96}^{4,8}$
Unsupervised	LOF	0.79	0.86	$\mathbf{0.82}^{2}$	0.60	0.32	0.42^{3}	0.94	0.85	$\mathbf{0.89}^{5}$	0.90	0.25	0.39^{4}
	encoder + LOF	0.89	0.28	0.43^{5}	0.58	0.26	0.36^{6}	0.65	0.27	0.38^{3}	0.45	0.11	0.17^{3}
	IForest	0.29	0.16	0.21^{1}	0.67	0.73	0.70^{1}	0.98	0.34	0.50^{5}	0.97	0.42	0.59^{2}
	encoder + IForest	0.91	0.82	$\mathbf{0.86}^{2}$	0.87	0.97	$\mathbf{0.92}^{1}$	0.80	0.96	$\mathbf{0.87}^{1}$	0.82	0.97	$\mathbf{0.89}^{1}$
	OC-SVM	0.77	0.87	$\mathbf{0.82}^{5,8}$	0.99	0.82	$\mathbf{0.90}^{2,7}$	0.76	0.82	$0.79^{5,8}$	0.82	0.90	$0.86^{6,7}$
	encoder + OC-SVM	0.76	0.85	$0.80^{3,8}$	0.71	0.91	$\mathbf{0.80}^{4,7}$	0.70	0.80	$0.75^{3,8}$	0.92	0.95	$\mathbf{0.93}^{1,7}$

- In most cases, automatically generated encoded data representation improves anomaly detection performance compared to the same non-encoded counterpart.

a: SuddenD ANOMALIES

For *SuddenD* observed in Table 5, all representations produce nearly perfect F1 scores of above 0.99 with all supervised ML approaches. Moving to unsupervised approaches, it can be readily seen that the histogram representation works best with LOF, however the F1 score of 0.76 is modest. The aggregated features with $F1 = 0.83$ work best with IForest followed by the histogram features with $F1 = 0.72$. The encoded representations surpass all non-encoded ones with this approach reaching F1 scores up to 0.97. All but the manual aggregated features yield good F1 scores of above 0.9 with OC-SVM, however the frequency representation dominates with F1 score of above 0.98. The encoded representations improve the anomaly detection performance in three of the four possible cases.

b: SuddenR ANOMALIES

For *SuddenR* observed in Table 6, almost all representations produce high F1 scores of above 0.9 with all supervised ML approaches. The time-value representation is slightly inferior to the other manual and autoencoded representations, producing 0.89 F1 score with LR, 0.96 with RForest and 0.97 with SVM.

TABLE 8. Performance of detecting slow degradation (SlowD) anomalies.

Approach	Technique	time-value features			aggregated features			histogram features			frequency domain		
		Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1
Baseline	Threshold (Tab. 3)	0.17	0.10	0.13[1]	0.47	0.03	0.06[1]	0.31	0.57	0.40[1]	-	-	-
Supervised	LR	0.97	0.96	0.97[3]	0.44	0.16	0.91[4]	0.92	0.87	0.90[2]	0.92	0.89	0.90[6]
	encoder + LR	0.99	0.98	**0.99**[2]	0.99	1.00	**1.00**[3]	1.00	1.00	**1.00**[1]	0.98	0.98	**0.98**[4]
	RForest	0.98	0.98	**0.98**[5]	0.98	0.99	0.99[4]	0.99	0.99	0.99[6]	0.92	0.92	0.92[3]
	encoder + RForest	0.98	0.98	**0.98**[4]	0.99	1.00	**1.00**[3]	1.00	1.00	**1.00**[6]	0.97	0.97	**0.97**[4]
	SVM	0.97	0.97	0.97[2,7]	0.98	0.99	0.99[6,8]	1.00	1.00	**1.00**[4,8]	0.93	0.94	0.94[2,8]
	encoder + SVM	0.98	0.99	**0.99**[4,7]	1.00	1.00	**1.00**[3,7]	1.00	1.00	**1.00**[6,7]	0.98	0.98	**0.98**[4,7]
Unsupervised	LOF	0.36	0.37	0.36[3]	0.28	0.23	0.26[4]	0.32	0.26	0.29[5]	0.43	0.02	0.04[2]
	encoder + LOF	0.59	0.12	0.20[4]	0.36	0.18	0.24[4]	0.29	0.20	0.23[3]	0.59	0.11	0.18[1]
	IForest	0.29	0.20	0.24[1]	0.74	0.55	**0.63**[3]	0.33	0.13	0.18[6]	0.30	0.09	0.14[1]
	encoder + IForest	0.86	0.97	**0.91**[4]	0.40	0.41	0.40[6]	0.49	0.58	**0.53**[6]	0.64	0.61	**0.63**[1]
	OC-SVM	0.46	0.81	0.59[5,7]	0.41	0.92	**0.56**[4,7]	0.65	0.69	**0.67**[1,7]	0.69	0.73	**0.71**[6,7]
	encoder + OC-SVM	0.71	0.76	**0.73**[4,8]	0.90	1.00	**0.95**[4,7]	0.77	1.00	**0.87**[6,7]	0.63	0.91	**0.75**[5,7]

[1]No-scaling [2]Only mean scaling (by standard scaler) [3]Mean and deviation scaling (by standard scaler) [4]Only mean scaling (by robust scaler with respect to values between Q25 and Q75) [5]Mean and deviation scaling (by robust scaler with respect to values between Q25 and Q75) [6]Min-Max scaler [7]Linear kernel [8]RBF kernel

For unsupervised approaches, unlike in the case of SuddenD, the time-series and histogram representations work best with LOF, with high F1 scores of above 0.93. Similarly, the aggregated features with $F1 = 0.75$ work best with IForest followed by the frequency representation with $F1 = 0.64$ for SuddenD anomaly. The encoded representations surpass all non-encoded ones with this approach reaching F1 scores up to 0.98. The manual features yield good scores of above 0.89 with OC-SVM, however the time-value and histogram representations dominate with F1 score of above 0.94. The encoded representations do not improve the anomaly detection performance for this anomaly type using OC-SVM.

c: InstaD ANOMALIES

For *InstaD* observed in Table 7, almost all representations produce high F1 scores of above 0.9 with all supervised ML approaches. The time-value representation is slightly inferior to the other manual and autoencoded representations, producing 0.89 F1 score with LR, 0.91 with RForest and 0.94 with SVM. While for the previous SuddenD and SuddenR the remaining three representations yielded comparable F1 scores with all ML approaches, for InstaD anomaly, frequency domain representation is less suitable when compared to histogram, and histogram features are less suitable than the aggregated features in terms of the anomaly detection performance.

Considering unsupervised approaches, the more arbitrary the anomaly becomes, so the effect of the representation on the results. The time-value representation and histogram work best with LOF with F1 up to 0.89 while the encoded representation provides no additional benefit. The manual representations work poorly with RForests while the encoded ones yield F1 scores of up to 0.92. The aggregated features and encoded frequency domain representations work best with OC-SVM with $F1 = 0.9$ and $F1 = 0.93$, respectively.

d: SlowD ANOMALIES

For *SlowD* observed in Table 8, all representations produce high F1 scores of above 0.9 with all supervised ML approaches. The time-value representation performs best with LR yielding $F1 = 0.97$, while all time-value, aggregated and histogram features work well with RForest and SVM yielding an F1 score of above 0.97. This anomaly type is relatively more difficult to be detected using frequency representation when supervised approaches are considered.

For unsupervised approaches, no representation works well with LOF while all manual representations perform modestly with F1 scores of up to 0.71. However, in some specific cases, the encoded representation achieves higher detection performance. For instance, time-value encoded with IForest yields an F1 score of 0.91, while aggregated encoded yields an F1 score of 0.95 with OC-SVM. All encoded representations perform better with OC-SVM compared to their non-encoded counterparts.

B. PERFORMANCE ANALYSES OF ML APPROACHES

We now analyse the detection performance of the ML approaches described in Section VI on all the anomaly types proposed in Section IV. By using Tables 5, 6, 7 and 8 we perform an analysis across rows, unlike the cross-column analysis performed in Section VIII-A for data representations. While in Section VIII-A we already explained, as an example, how the LR approach works on our anomaly dataset, this section elaborates, as an exemplifying analysis, on what the

tree based ensemble learns. We selected the tree based ensemble as it is also easily explainable and tractable similar to LR. For the start, we remark the following major observations.

- For a given anomaly type, there is no major difference between the three selected supervised approaches.
- Among the unsupervised approaches, OC-SVM performs the best F1 scores, closely followed by IForest, whereas LOF typically performs the worst F1 scores.

1) SuddenD ANOMALIES

According to Table 5, the supervised models are able to detect SuddenD anomalies more accurately than the unsupervised models. All three supervised models have achieved near perfect F1 score of 0.99 on all data representations.

The tree based ensemble models, such as supervised RForest and unsupervised IForest, learn a set of trees and subsequently use a voting mechanism on the decision of each individual tree to determine the final class. A tree that is the fundamental part of the two ensemble models also learns which features are the most important ones. The feature with the highest discrimination power (weight) is situated at the root of the tree, then on the left and right nodes, as it can be exemplified in Figure 14, the next two important features are placed and the process follows until a certain stopping criterion is met. In our specific case, the trees learn the thresholds for particular values in the feature vector. For instance, depicted in Figure 14, it can be seen that if the value at position 290 in the time-value representation, denoted by X_{290}, is below -92.5, then the link is anomalous, otherwise it is a normal link. This simple rule is able to correctly detect $n = 596$ anomalous links and $n = 1520$ normal links while only misclassifying 7 links, thus the performance of that tree alone is $F1 = 0.99$.

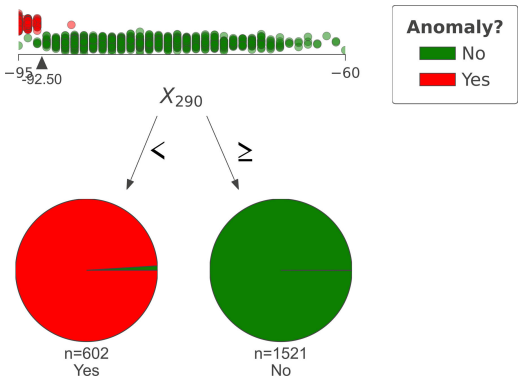


FIGURE 14. An exemplifying decision tree for the detection process of SuddenD anomaly.

The SVM models are more complex and difficult to visualize when a feature vector has more than 3 dimensions as it is the case with all manual and autoencoded representations used in this paper. SVMs essentially compute a hyperplane that attempts to separate the N-dimensional feature vector according to a criterion, such as the labels.

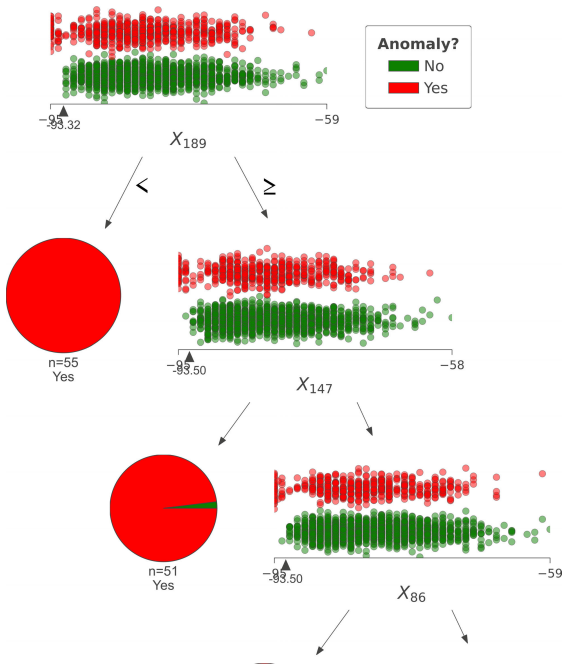


FIGURE 15. A part of the decision tree while detecting SuddenR anomaly over time-value representation.

Among the unsupervised approaches, OC-SVM is able to achieve F1 scores close to the supervised approaches, for instance 0.98, 0.96 and 0.90 on FFT, histogram and time-value representations, respectively. For OC-SVM model, with the aid of autoencoder the time-value representation is transformed to an important summary of the data by removing the noise and repetitions, leading to a performance increase from $F1 = 0.83$ to $F1 = 0.96$. Next, IForest achieved a lower performance with an F1 score between 0.61 and 0.83, the latter on the aggregated representation 0.83 while the LOF performance reached 0.76 on one occasion.

2) SuddenR ANOMALIES

Compared to SuddenD, SuddenR gains a steep recovery slope, while the duration and occurrence are more random. The results in Table 6 show that supervised models are able to detect SuddenR more accurately than the unsupervised models. F1 score of supervised models ranges from 0.89 with LR on time-value representation to near perfect F1 score for remaining supervised approaches. Using encoded representation of the time-values improves the performance also in the case of LR to 0.99, which corresponds to an about 11% improvement. For the LR case, as discussed in Section VIII-A and depicted in Figure 11, the most important features are the ones that attempt to capture the random drops between packets 25 and 275.

A decision tree representing RForest and IForest ensembles is portrayed in Figure 15 for the time-value representation of the SuddenR anomaly. It can be seen that the

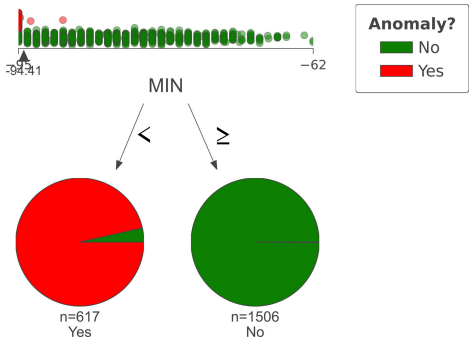


FIGURE 16. A part of the decision tree while detecting SuddenR anomaly over aggregated data representation.

most discriminative data points are X_{189}, X_{147}, X_{86} with -93.5 dBm RSSI threshold. The tree can grow very deep, eventually over-fitting the data, however, as discussed in Section VII, we undertook standard methods for avoiding that in the experimental design. Figure 16 presents an example tree learnt on aggregated feature representation. Similar to the tree in Figure 14, it is simple and effective, where it compares minimal RSSI to -94.407 dBm threshold to decide whether it is anomaly or not. Figure 17 shows a tree learnt using the histogram representation as input. While performance is similar to the previous representation, we see that using aggregated representation requires less number of decisions, i.e., depth of tree, for effective anomaly detection. Similar observations can be made for the tree learnt on fft representation for this anomaly type depicted in Figure 18.

Among the unsupervised approaches, OC-SVM, without encoded representation, is able to achieve an F1 score of around 0.90 on average through all four representations, which is almost on par with supervised approaches. IForest, on the other hand, performs much better with encoded representations, where the most significant improvement is presented on time-value representation ramping its F1 score from 0.21 to 0.86. Since SuddenR is limited in duration and thus affecting less number of features, LOF is able to pull ahead in time-value and histogram representations, where it reaches an F1 score of above 0.93.

3) InstaD ANOMALIES

In contrast to SuddenD and SuddenR, InstanD appears as an anomaly with extremely short duration (pulse). The results in Table 7 show that supervised approaches are slightly better at InstaD classification. F1 performance score of supervised approaches is slightly worse (up to 0.98) from what we have seen for SuddenD or SuddenR detection performance. Due to the arbitrary characteristics of this anomaly type, the F1 score is diminished further when the supervised approaches are

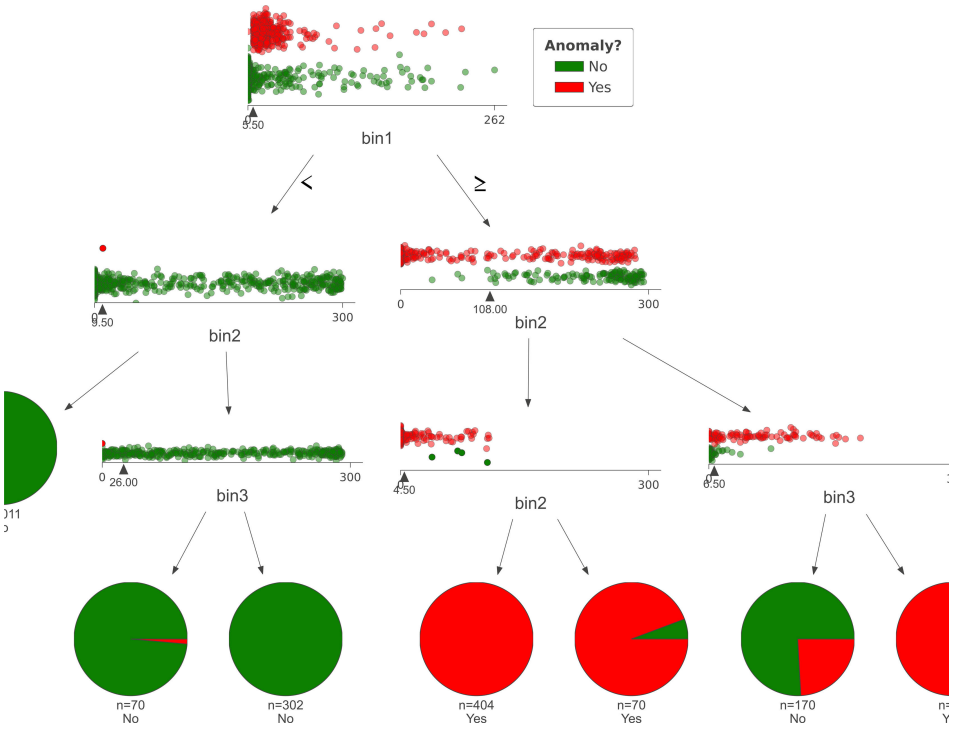


FIGURE 17. A part of the decision tree while detecting SuddenR anomaly over histogram representation.

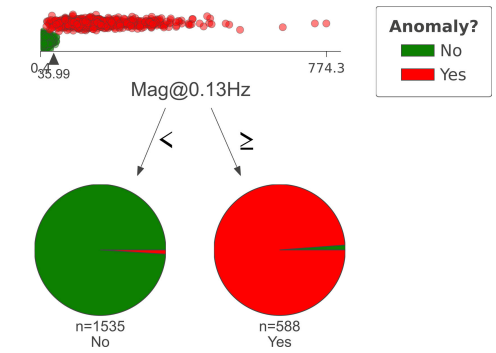


FIGURE 18. Decision tree for detecting SuddenR anomaly using FFT representation.

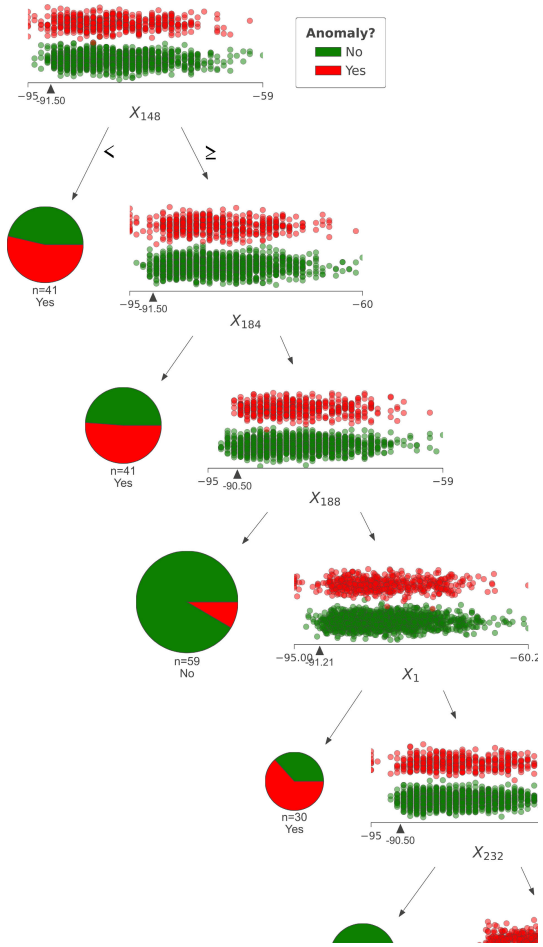


FIGURE 19. A part of the decision tree while detecting InstaD anomaly over time-value representation.

trained with the time-value and frequency domain representations as outlined in Table 7.

To better understand decision making on classifying the InstaD anomaly, we examine a decision tree representing RForest and IForest ensembles as depicted in Figure 19. Due to the random nature of this anomaly, the tree selects random points and verifies their value against a learnt threshold. For this particular tree, feature X_{148} that is compared to -91.50 dBm RSSI threshold is selected in the root. Then, it follows with the comparisons of the features in order of X_{184} and X_{188} that are compared to -91.50 dBm and -90.50 dBm, respectively and this process terminates when the final depth of the three is reached. For this anomaly type, time-series and FFT domain may not be the optimal data representations for the sake of developing a reliable and non-overfitting model.

Among the unsupervised approaches, there is no clear best approach. The top five performing models are OC-SVM using encoded FFT with 0.93 F1 score, IForest using encoded aggregated features with an F1 score of 0.92, OC-SVM using aggregated representation with an F1 score of 0.90, and LOF using histogram representation and IForest using encoded FFT, both achieving an F1 score of 0.89.

4) SlowD ANOMALIES

In contrast to SuddenD, SuddenR and InstaD, SlowD does not appear instantly, but rather gradually with random slope. The results in Table 8 show that supervised approaches are still superior to unsupervised ones. For supervised approaches, the average F1 score, ranging between 0.90 and the perfect score, is slightly better than InstaD, but slightly worse than SuddenD and SuddenR. The most notable drop in performance is observed with LR approach over aggregated, histogram and frequency representations.

To better understand the underlying reasons behind the detection performance, we visualized in Figure 20 a typical decision tree learnt on the time-value representation of this anomaly. It can be seen from the figure that the tree commences with a comparison of feature X_{282} (282nd item in time-series) to the threshold of -92 dBm. By doing so, it tries to distil anomalous samples at the end of the series, since samples with SlowD anomaly are suppose to have lower value towards the end of the time-series. However, as the first pie-chart reveals, this is not always the case, since some of the fully functioning non-anomalous (normal) links in the dataset have average RSSI close to that threshold, which leads to a high misclassification rate. In the second step of decision making, the process is repeated by comparing an earlier feature X_{64} against -89.70 dBm threshold. The tree continues to learn according to this pattern until a stopping criterion is met.

Among the unsupervised approaches, according to Table 8, the best approach is OC-SVM with best F1 scores from 0.71 to 0.95, followed by IForest with best F1 scores from 0.63 to 0.91. LOF, as an alternative unsupervised candidate, has poorly performed over all scenarios.

C. LIMITATIONS

We identify three main limitations that apply to this treatise, and to the best of our understanding also to most of the other related works in wireless network and IoT anomaly

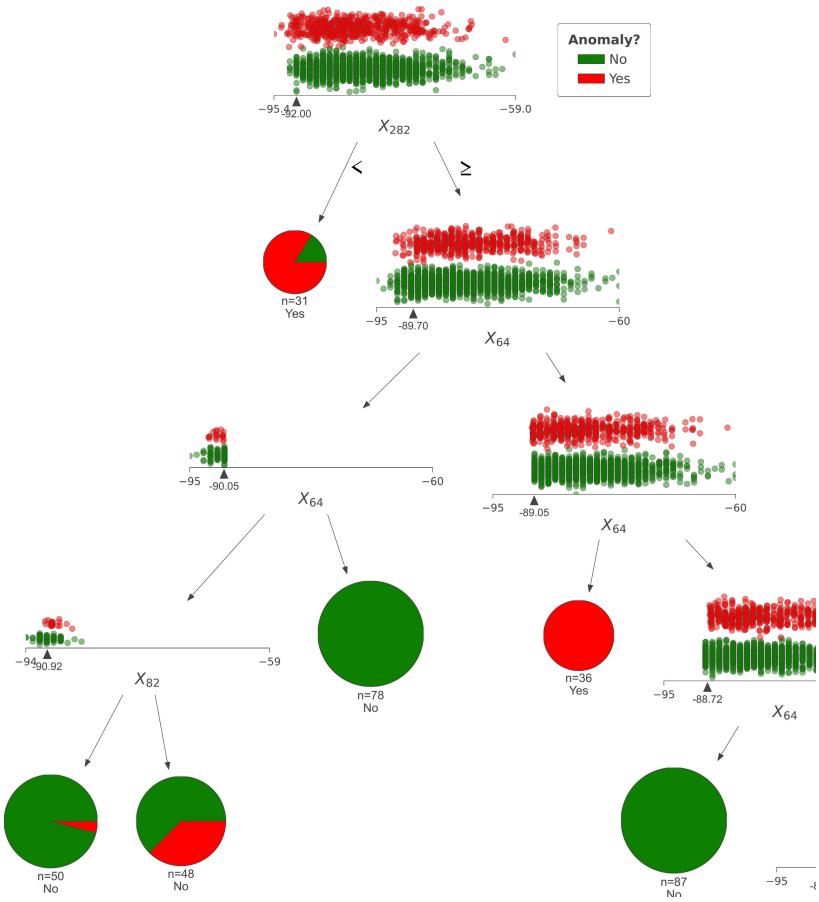


FIGURE 20. A part of the decision tree while detecting SlowD anomaly over time-value representation.

data that do not target real-world application data, such as measurements.

1) LIMITATION 1

Every ML-based tool needs sufficient data for training and evaluation. Quantifying "sufficient" is difficult but in general it means that the model needs to see enough training examples to be able to accurately approximate the underlying distribution. Intuition would say that the data that is "sufficient" to learn a normal distribution would be smaller in size than the data needed to learn an exponential distribution. While synthetic data is useful to develop a proof of concept, for anything more than that real data is required. To the best of our knowledge, only few related works consider real-world data [26] and none of them considers link layer traces.

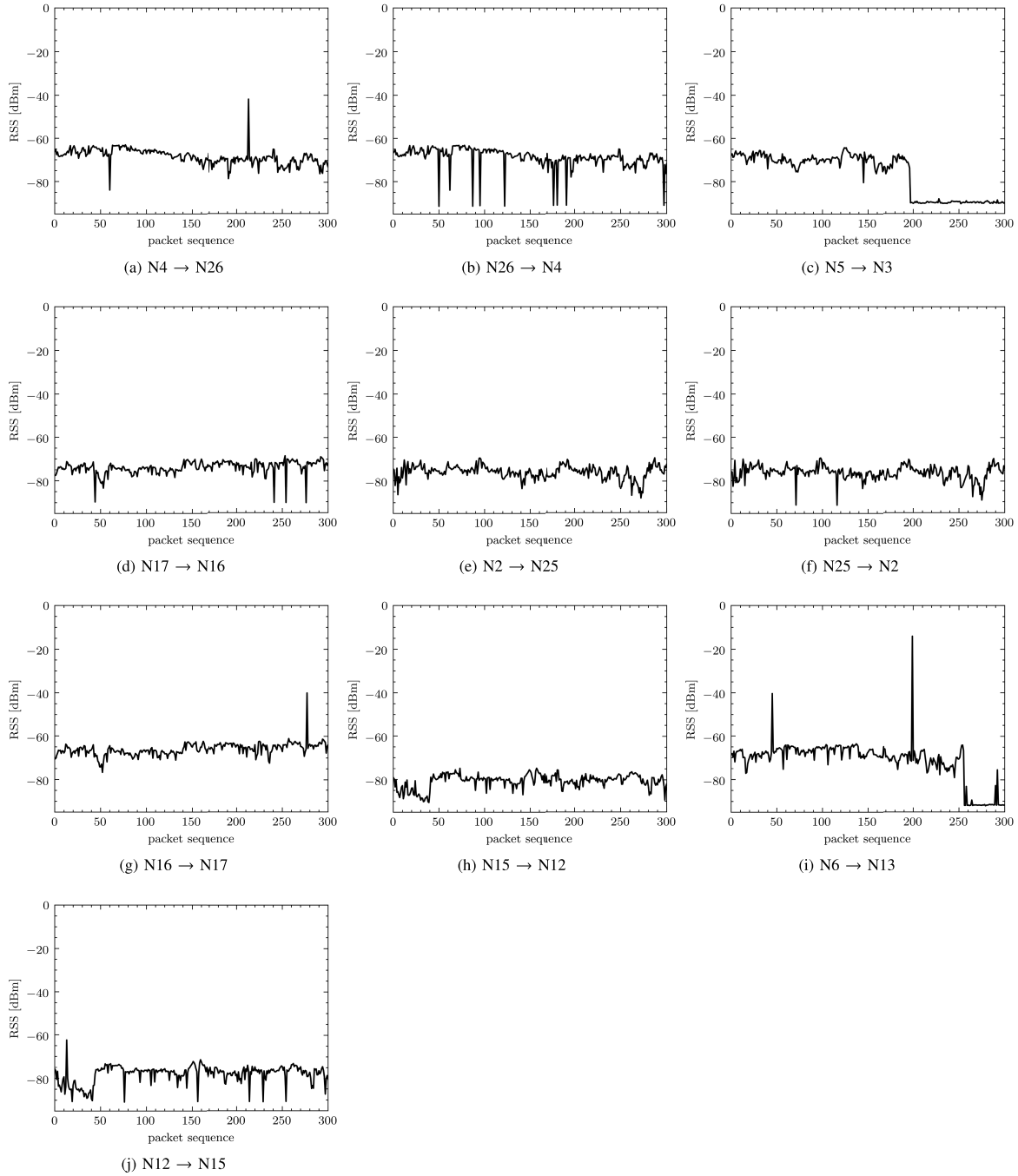
In this study, we developed the ML models using IEEE 802.11 traces available from a public dataset as the motivation data from LOG-a-TEC contains only 11 IEEE 802.15.4 traces all depicted in Figure 21. Table 9 shows how the LR model developed on IEEE 802.11 traces performs on the IEEE 802.15.4 traces. The first column of

TABLE 9. Predicted anomalies on validation data, as illustrated in Figure 21.

Model	Predicted anomalies
LR SuddenD	Figures 21c and 21i
LR SuddenR	Figures 21c and 21f
LR InstaD	Figures 21c and 21i
LR SlowD	Figures 21b, 21c, 21d, 21e, 21f and 21i

the table lists the LR model corresponding to the anomalies defined in this paper while the second includes the subfigures with links that were classified as having the respective anomalies. It can be seen from the first row of the table that the SuddenD degradations in the IEEE 802.15.4 traces are detected correctly and they appear in the links represented in Figures 21c and 21i, while for the other degradations the models seem to generate false positives.

According to the second row of Table 9, it can be seen that the links represented in Figures 21c and 21f have been classified as SuddenR. However, when visually inspecting the links in those respective subfigures it can be seen that they are both classified as false positives. It is hard to determine the reason for misclassification since none of the classified



(a) N4 → N26 (b) N26 → N4 (c) N5 → N3

(d) N17 → N16 (e) N2 → N25 (f) N25 → N2

(g) N16 → N17 (h) N15 → N12 (i) N6 → N13

(j) N12 → N15

FIGURE 21. Anomaly detection validation test employed over real-world measurements gleaned from the LOG-a-TEC testbed, where for example, as in (g) N16→N17 indicates a communication link between nodes 16 and 17.

traces even remotely resemble SuddenR anomaly presented in Figure 4a.

As per to the third row of Table 9, we observe that the two links that are detected as having InstaD anomaly are false positives. As also discussed in Section VIII-A and Figure 12a, the weights change dynamically and arbitrarily for such anomalies, and thus no distinct pattern can be readily detected.

Finally, the last row of the table shows that a large number of 802.15.4 links are falsely classified as having SlowD anomalies. While we can see that the trace in Figure 21b contains a slightly descending slope predicted to be SlowD anomaly, this model produces false positives over the other traces in Figure 21. As discussed in Section VIII-A, the discriminative importance of the features for the detection of SlowD is sought in the last part of the signal trace. This is why Figures 21c and 21i, and to some extent Figures 21e and 21f are inevitably misclassified, since they contain lower values in the last portion of the trace.

As a conclusion, the learnt models on the relatively limited IEEE 802.11 traces are not directly and reliably transferable to the IEEE 802.15.4 traces, which indicates that the developed models cannot be readily generalized across various technologies and possibly for distinct applications.

2) LIMITATION 2

The architecture of the autoencoder that learns the encoded features has been selected for a small number of candidates as a result of the trial-and-error method. Having more data would enable training an autoencoder, which then can be better generalized for even unseen examples. Autoencoder optimization and end-to-end deep learning for the proposed anomaly types might bring further insights into developing better performing and more reliable anomaly detection models. However, as hyperparameter search in deep learning is challenging and needs a large amount of training data, we leave such optimization for the future work.

3) LIMITATION 3

In this study, we only developed offline models that would need to be periodically retrained in real-world applications in order to account for the dynamically changing environments, which are the inherent characteristics of wireless networks. This leads us to online models that can learn from continuous incoming (streaming) data. Roughly speaking, offline models outperform online counterpart models in terms of the required computational power, albeit online models are able to rapidly adapt to the changes within the application environment in an automated way thus simplify the detection system that would otherwise need to periodically re-train and update the offline models.

IX. CONCLUSION

In this paper, we introduce four types of anomalies that can be present in wireless links and are useful for being detected in real-world operational IoT deployments. We demonstrated that these anomalies were exposed on a real-world IoT deployment, namely the LOG-a-TEC testbed, and they significantly affected the expected operations of the testbed. Motivated by this, we develop detection models for each type of anomaly by considering five different data representations and six different ML techniques. We performed an extensive relative evaluation of the models from data representations and ML models perspective, and the limitations of our models are discussed. The resulting tool-set for anomaly injection, feature generation and model development are made publicly available for reproducibility.

Our study reveals that *with respect to the data representations*; i) none of the four manually generated features clearly dominates the remaining ones in terms of anomaly detection performance, and ii) in most cases, automatically generated encoded data representations improve anomaly detection performance by up to 40% compared to their non-encoded counterparts.

With respect to the selected ML approach, our results demonstrate that; i) there is no major difference among the selected supervised ML approaches, where all are capable of detecting anomalies with F1 scores of above 0.98, and ii) the unsupervised approaches are also able to detect anomalies with F1 scores of, on average, about 0.90 and OC-SVM outperforms all the other unsupervised ones reaching at F1 scores of 0.99 for SuddenD, 0.95 for SuddenR, 0.93 for InstaD and 0.95 for SlowD.

ACKNOWLEDGMENT

The authors would like to recognize Tomaž Šolc, one of the core developers of the LOG-a-TEC testbed for his contribution to the motivation of this work.

REFERENCES

[1] T. Qiu, N. Chen, K. Li, M. Atiquzzaman, and W. Zhao, "How can heterogeneous Internet of Things build our future: A survey," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 3, pp. 2011–2027, 3rd Quart., 2018.

[2] J. Davies and C. Fortuna, *The Internet of Things: From Data to Insight*. Hoboken, NJ, USA: Wiley, 2020.

[3] R. Díaz-Díaz, L. Muñoz, and D. Pérez-González, "Business model analysis of public services operating in the smart city ecosystem: The case of SmartSantander," *Future Gener. Comput. Syst.*, vol. 76, pp. 198–214, Nov. 2017.

[4] U. Wetzker, I. Splitt, M. Zimmerling, C. A. Boano, and K. Römer, "Troubleshooting wireless coexistence problems in the industrial Internet of Things," in *Proc. IEEE Int. Conf. Comput. Sci. Eng. (CSE)*, Paris, France, Aug. 2016, p. 98.

[5] J. D. C. Silva, J. J. P. Rodrigues, K. Saleem, S. A. Kozlov, and R. A. Rabêlo, "M4DN. IoT—A networks and devices management platform for Internet of Things," *IEEE Access*, vol. 7, pp. 53305–53313, Apr. 2019.

[6] A. Sheth, C. Doerr, D. Grunwald, R. Han, and D. Sicker, "MOJO: A distributed physical layer anomaly detection system for 802.11 WLANs," in *Proc. 4th Int. Conf. Mobile Syst., Appl. Services (MobiSys)*, 2006, pp. 191–204.

[7] S. Gupta, R. Zheng, and A. M. K. Cheng, "ANDES: An anomaly detection system for wireless sensor networks," in *Proc. IEEE Int. Conf. Mobile Adhoc Sensor Syst.*, Oct. 2007, pp. 1–9.

[8] H. Alipour, Y. B. Al-Nashif, P. Satam, and S. Hariri, "Wireless anomaly detection based on IEEE 802.11 behavior analysis," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 10, pp. 2158–2170, Oct. 2015.

[9] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, no. 3, p. 58, Jul. 2009.

[10] M. Vucnik, T. Solc, U. Gregorc, A. Hrovat, K. Bregar, M. Smolnikar, M. Mohorcic, and C. Fortuna, "Continuous integration in wireless technology development," *IEEE Commun. Mag.*, vol. 56, no. 12, pp. 74–81, Dec. 2018.

[11] P. Gogoi, D. K. Bhattacharyya, B. Borah, and J. K. Kalita, "A survey of outlier detection methods in network anomaly identification," *Comput. J.*, vol. 54, no. 4, pp. 570–588, Apr. 2011.

[12] A. Zimek, E. Schubert, and H.-P. Kriegel, "A survey on unsupervised outlier detection in high-dimensional numerical data," *Stat. Anal. Data Mining*, vol. 5, no. 5, pp. 363–387, Oct. 2012.

[13] C. C. Aggarwal, "Outlier ensembles: Position paper," *ACM SIGKDD Explor. Newslett.*, vol. 14, no. 2, pp. 49–58, Apr. 2013.

[14] M. Gupta, J. Gao, C. C. Aggarwal, and J. Han, "Outlier detection for temporal data: A survey," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 9, pp. 2250–2267, Sep. 2014.
[15] X. Xu, H. Liu, and M. Yao, "Recent progress of anomaly detection," *Complexity*, vol. 2019, pp. 1–11, Jan. 2019.
[16] A. A. Cook, G. Misirli, and Z. Fan, "Anomaly detection for IoT time-series data: A survey," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 6481–6494, Jul. 2020.
[17] A. Lavin and S. Ahmad, "Evaluating real-time anomaly detection algorithms–the Numenta anomaly benchmark," in *Proc. IEEE 14th Int. Conf. Mach. Learn. Appl. (ICMLA)*, Dec. 2015, pp. 38–44.
[18] R. Jurdak, X. R. Wang, O. Obst, and P. Valencia, "Wireless sensor network anomalies: Diagnosis and detection strategies," in *Intelligence-Based Systems Engineering*. Berlin, Germany: Springer, 2011, pp. 309–325, doi: 10.1007/978-3-642-17931-0_12.
[19] T. Kieu, B. Yang, C. Guo, and C. S. Jensen, "Outlier detection for time series with recurrent autoencoder ensembles," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Macao, Republic of China, Aug. 2019, pp. 2725–2732.
[20] T. J. O'Shea, J. Corgan, and T. C. Clancy, "Unsupervised representation learning of structured radio communication signals," in *Proc. 1st Int. Workshop Sens., Process. Learn. Intell. Mach. (SPLINE)*, Jul. 2016, pp. 1–5.
[21] T. J. O'Shea, T. Erpek, and T. Charles Clancy, "Deep learning based MIMO communications," 2017, *arXiv:1707.07980*. [Online]. Available: http://arxiv.org/abs/1707.07980
[22] H. Zhang, K. Liu, Q. Shang, L. Feng, C. Chen, Z. Wu, and S. Guo, "Dual-band Wi-Fi based indoor localization via stacked denosing autoencoder," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Waikoloa, HI, USA, Dec. 2019, pp. 1–6.
[23] B. Wang, F. Hu, Y. Zhao, and T. N. Guo, "Anomaly detection and array diagnosis in wireless networks with multiple antennas: Framework, challenges and tools," *IEEE Netw.*, vol. 32, no. 1, pp. 152–159, Jan. 2018.
[24] M. R. Shahid, G. Blanc, Z. Zhang, and H. Debar, "Anomalous communications detection in IoT networks using sparse autoencoders," in *Proc. IEEE 18th Int. Symp. Netw. Comput. Appl. (NCA)*, Cambridge, MA, USA, Sep. 2019, pp. 1–5.
[25] Z. Chen, C. K. Yeo, B. S. Lee, and C. T. Lau, "Autoencoder-based network anomaly detection," in *Proc. Wireless Telecommun. Symp. (WTS)*, Phoenix, AZ, USA, Apr. 2018, pp. 1–5.
[26] C. Yin, S. Zhang, J. Wang, and N. N. Xiong, "Anomaly detection based on convolutional recurrent autoencoder for IoT time series," *IEEE Trans. Syst., Man, Cybern. Syst.*, early access, Feb. 7, 2020, doi: 10.1109/TSMC.2020.2968516.
[27] V. L. L. Thing, "IEEE 802.11 network anomaly detection and attack classification: A deep learning approach," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, San Francisco, CA, USA, Mar. 2017, pp. 1–6.
[28] J. Ran, Y. Ji, and B. Tang, "A semi-supervised learning approach to IEEE 802.11 network anomaly detection," in *Proc. IEEE 89th Veh. Technol. Conf. (VTC-Spring)*, Apr. 2019, pp. 1–5.
[29] O. Salem, A. Guerassimov, A. Mehaoua, A. Marcus, and B. Furht, "Anomaly detection in medical wireless sensor networks using SVM and linear regression models," *Int. J. E-Health Med. Commun.*, vol. 5, no. 1, pp. 20–45, Jan. 2014.
[30] O. Salem, A. Guerassimov, A. Mehaoua, A. Marcus, and B. Furht, "Sensor fault and patient anomaly detection and classification in medical wireless sensor networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Budapest, Hungary, Jun. 2013, pp. 4373–4378.
[31] M. A. Alsheikh, S. Lin, D. Niyato, and H.-P. Tan, "Machine learning in wireless sensor networks: Algorithms, strategies, and applications," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 4, pp. 1996–2018, 4th Quart., 2014.
[32] T. Šolc, C. Fortuna, and M. Mohorčič, "Low-cost testbed development and its applications in cognitive radio prototyping," in *Cognitive Radio and Networking for Heterogeneous Wireless Networks*. New York, NY, USA: Springer, 2015, pp. 361–405.
[33] T. Šolc and Z. Padrah, "Network design for the LOG-a-TEC outdoor testbed," in *Proc. 2nd Int. Workshop Meas.-Based Experim. Res., Methodol. Tools*, 2013.
[34] J. K. Mann, S. Perinpanayagam, and I. Jennions, "Aging detection capability for switch-mode power converters," *IEEE Trans. Ind. Electron.*, vol. 63, no. 5, pp. 3216–3227, May 2016.
[35] J. Lin, S. Williamson, K. Borne, and D. DeBarr, "Pattern recognition in time series," *Adv. Mach. Learn. Data Mining Astron.*, vol. 1, nos. 617–645, p. 3, 2012.
[36] M. A. Kramer, "Nonlinear principal component analysis using autoassociative neural networks," *AIChE J.*, vol. 37, no. 2, pp. 233–243, Feb. 1991.
[37] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.
[38] I. Alawe, A. Ksentini, Y. Hadjadj-Aoul, and P. Bertin, "Improving traffic forecasting for 5G core network scalability: A machine learning approach," *IEEE Netw.*, vol. 32, no. 6, pp. 42–49, Nov. 2018.
[39] R. B. D'Agostino, "An omnibus test of normality for moderate and large size samples," *Biometrika*, vol. 58, no. 2, pp. 341–348, 1971.
[40] R. D'Agostino and E. S. Pearson, "Tests for departure from normality. empirical results for the distributions of b^2 and $\sqrt{b^1}$," *Biometrika*, vol. 60, no. 3, pp. 613–622, 1973.
[41] V. Hodge and J. Austin, "A survey of outlier detection methodologies," *Artif. Intell. Rev.*, vol. 22, no. 2, pp. 85–126, Oct. 2004.
[42] R. Malouf, "A comparison of algorithms for maximum entropy parameter estimation," in *Proc. 6th Conf. Natural Lang. Learn.*, vol. 20. Stroudsburg, PA, USA: Association Computational Linguistics, 2002, pp. 1–7, doi: 10.3115/1118853.1118871.
[43] C. C. Aggarwal, "Outlier analysis," in *Data Mining*. New York, NY, USA: Springer, 2015, pp. 237–263.
[44] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.
[45] V. Vapnik, *The Nature of Statistical Learning Theory*. New York, NY, USA: Springer, 2013.
[46] L. Lin and Z. Xiaolong, "Optimization of SVM with RBF Kernel," *Comput. Eng. Appl.*, vol. 42, no. 29, pp. 190–192 and 204, 2006. [Online]. Available: https://jglobal.jst.go.jp/en/detail?JGLOBAL_ID=200902220640336864
[47] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "LOF: Identifying density-based local outliers," *SIGMOD Rec.*, vol. 29, no. 2, p. 93–104, May 2000, doi: 10.1145/335191.335388.
[48] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *Proc. 8th IEEE Int. Conf. Data Mining*, Dec. 2008, pp. 413–422.
[49] S. K. Kaul, I. Seskar, and M. Gruteser. (Apr. 2007). *CRAWDAD Dataset Rutgers/Noise (v. 2007-04-20)*. [Online]. Available: https://crawdad.org/rutgers/noise/20070420/RSSI
[50] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. ICML*, 2013, vol. 30, no. 1, p. 3.

GREGOR CERAR (Graduate Student Member, IEEE) received the master's degree in telecommunications from the Faculty of Electrical Engineering, University of Ljubljana, in 2016. He is currently pursuing the Ph.D. degree with the Jožef Stefan International Postgraduate School. He is also a Research Assistant with the Department of Communication Systems, Jožef Stefan Institute. His main research interests include the IoT, wireless networking of constrained devices, and machine learning applications in IoT.

HALIL YETGIN (Member, IEEE) received the B.Eng. degree in computer engineering from Selcuk University, Turkey, in 2008, the M.Sc. degree in wireless communications from the University of Southampton, U.K., in 2010, and the Ph.D. degree in wireless communications from the Next Generation Wireless Research Group, University of Southampton, in 2015. He is currently an Assistant Professor with the Department of Electrical and Electronics Engineering, Bitlis Eren University, Turkey, and a Research Fellow with the Department of Communication Systems, Jožef Stefan Institute, Ljubljana, Slovenia. His research interests include the development of intelligent communication systems, energy efficient cross-layer design, and resource allocation of the future wireless communication networks. He was a recipient of the full scholarship granted by the Republic of Turkey, Ministry of National Education.

BLAZ BERTALANIC (Member, IEEE) received the master's degree in electrical engineering from the Faculty of Electrical Engineering, University of Ljubljana, in 2020, where he is currently pursuing the Ph.D. degree. He is also a Junior Researcher with the Department of Communication Systems, Jožef Stefan Institute. His main research interests include solving classification problems with the help of machine learning, wireless networking, electronics, and signal processing.

CAROLINA FORTUNA received the B.Sc. degree, in 2006, and the Ph.D. degree, in 2013. She was a Postdoctoral Research Associate with IBCN, Ghent University, from 2014 to 2015. She is currently a Research Fellow with the Department of Communication Systems, Jožef Stefan Institute, and an Assistant with the Jožef Stefan International Postgraduate School. Her research interests include interdisciplinary, focusing on data and knowledge driven modeling of communication and sensor systems. She has participated in H2020, FP7, and FP6 projects. In H2020 WiSHFUL, she was the Technical Leader of the project on behalf of UGhent/iMinds while in FP7 CREW she was the Technical Leader of the JSI Team. She has coauthored more than 50 peer-reviewed publications, was a TPC Member at IEEE ICC 2011, 2012, 2013, 2014, 2016, ESWC 2012, IEEE GLOBECOM 2011, 2016, VTC 2010, 2016, and IEEE WCNC 2009.

Chapter 6

ML-based Model Selection for Anomalous Wireless Link Detection

This chapter extends the work presented in Chapter 5 by further detailing the model selection process and hyperparameters to improve the wireless link anomaly detection performance, and provides insights into the process of developing and selecting the optimal model for anomalous wireless link classification.

Following the design steps and findings from Chapter 2, we present the design and development process of models for anomaly detection in wireless links. During the design process, we show that the development phase naturally produces an unduly large set of candidate models owing to data transformations, data scaling methods, selected ML techniques, and a variety of tunable parameters and their range of values, requiring a model ranking process to select the most appropriate models considering the application criteria.

In the design process, we consider four different data representations, four anomaly types, six scaling methods, and six different ML algorithms along with their associated parametrization attempts, resulting in over twenty thousand anomaly detection models for wireless links. We show that the transformation step, technique selection and parameter tuning significantly improve the final model, thus confirming hypothesis **H3**.

As to the contributions outlined in Chapter 1.3, this chapter represents parts of contributions **C4** and **C5**. We compared the performance of new supervised and unsupervised anomaly detection classifiers based on cross-layer data obtained from real-world wireless network testbeds (**C4**). In the process, we experimented with two data representations and hyperparametrization of the data pre-procesing steps to achieve further improved detection performance. The comparison was done using standard classification performance metrics. For contribution **C5**, we demonstrate that supervised SVM algorithm trained on encoded representations, which are marked as TRACESET #3 and #4 and correspond to encoded time-series and encoded FFT representations, achieve the highest F1-score of the evaluated models.

6.1 Problem Statement

Before the development of a well performing anomalous link classifier using ML, the learning problem has to be clearly specified and certain design decisions have to be taken during the development phase. Then, the selection phase of the most suitable model can be carried out.

6.1.1 Model development phase

As portrayed in Fig. 6.1, the development phase encompasses data transformation, data scaling methods, ML technique selection, parameter tuning and obtaining set of candidate models that represent the final outcome of this phase. The raw data collected from target wireless networks is usually pre-processed before being used to train ML techniques to obtain a model. Preprocessing [1] is a standard practice used in ML and performs operations, such as cleaning, interpolation, feature generation, resampling, window selection and other required transformation on the raw data. As our ultimate goal is to detect anomalies rather than patterns in the raw time-series, we only focus on feature engineering by subjecting the raw data to transformations through different representations targeting dimensionality reduction for more efficient and fast learning. During the design process, developers need to decide on an appropriate transformation strategy. Assuming D different strategies have been selected for transforming the raw data, this will result in D candidate tracesets to be used for training and evaluation of the ML models.

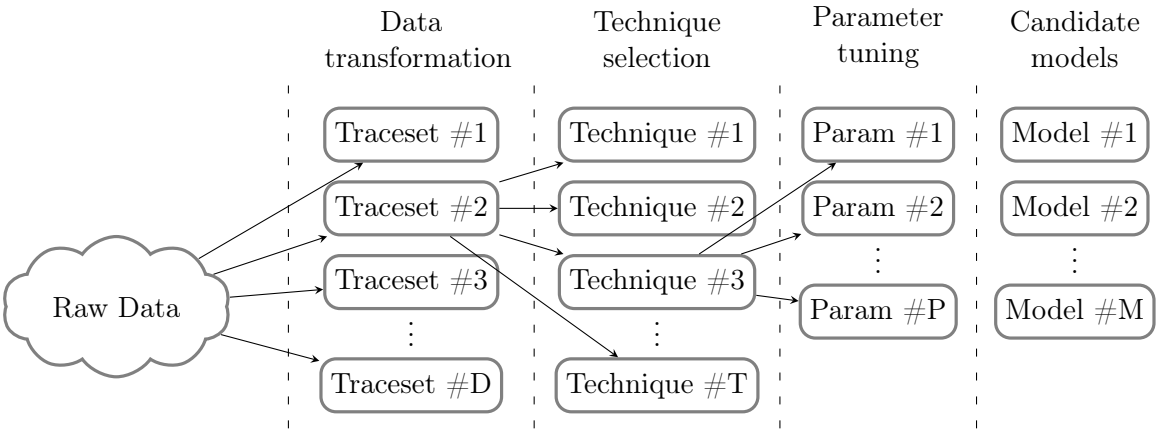


Figure 6.1: Process of the development phase for ML-based anomalous link classification model presenting that T number of Techniques are employed over D number of Tracesets with P number of Parameter tuning, consequently terminating with M number of candidate models.

Next, ML techniques to be utilized for training anomaly detection models for A number of anomaly types, have to be determined at the time of design process. Assuming that each ML technique T has to be employed over all D tracesets, also scaled with S number of scaling methods, and incorporates a predetermined number of parameters P and that each traceset is implemented with the same set of P parameters inherited from the relevant ML technique, as depicted in the technique selection and parameter tuning processes of Fig. 6.1, respectively. Then, the total number of ML model M can be derived as follows,

$$M = A \cdot S \cdot D \sum_{i=1}^{T} (P_{T_i}), \tag{6.1}$$

where P_{T_i} refers to the number of parameter tunings required by the ith ML technique represented by T_i. Again, each ML technique and its respective implementation exposes a certain number of parameters that are used for tuning and are independent of tracesets. However, notice that each T may have a set of parameter list, which requires a distinct combination of all parameters. For instance, support vector machine in Table 6.2 has 3 different set of parameters, namely C, *kernel* and *gamma*, each of which in order contains

6, 2 and 2 respective parameters resulting in 24 different models using distinct combination of parameters. Ultimately, terminating this entire development phase provides us with M number of distinct candidate ML models to be utilized as the wireless link anomaly classifier.

As a toy example, suppose that 4 different data representations are considered with 6 distinct scaling methods, on which 6 ML techniques will be implemented for 4 different anomaly types, each of which contains different set of parameters to be tuned, say $P_{T_1} = 2(3,2), P_{T_2} = 2, P_{T_3} = 1, P_{T_4} = 4, P_{T_5} = 5, P_{T_6} = 6$, then the number of candidate models produced by this phase becomes $M = 4 \cdot 6 \cdot 4 \cdot (24) = 2304$ distinct models, where $P_{T_1} = 2(3,2)$ indicates that the first ML technique has two set of parameters each of which contains three and two different parameter values, respectively.

6.1.2 Model selection phase

Given the candidate models developed during the previous phase, the most suitable ones have to be identified during the selection phase. For selecting the most suitable ML models, a set of performance criteria and their associated application requirements have to be specified, as illustrated under ranking criteria in Fig. 6.2. An application requirement may necessitate several criteria (C) for determining the most suitable model, which is, for example, determined as C1 and C2 for ranking R1 as shown by the second column of Fig. 6.2. Again, the candidate models can be ranked under any selected combination of the C criteria depending on the application requirements, which should be analysed and determined by the model designers. Finally, this phase ends with the selection of the best performing model(s) from those ranked lists. For example, a designer may opt for ranking the candidate models by using the combination of $F1$ and *Accuracy* scores, and select the model that maximizes both in the respective ranking process.

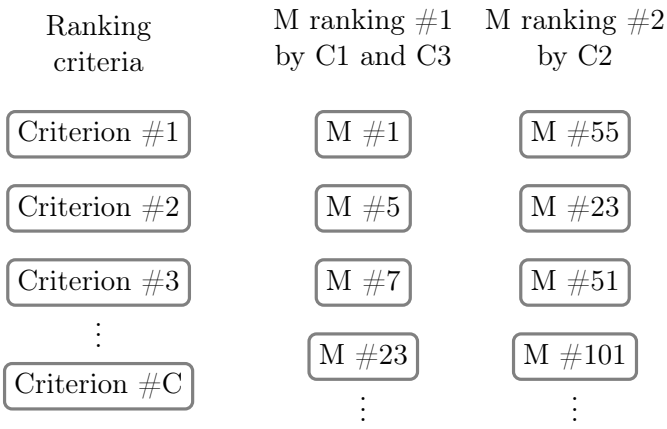


Figure 6.2: Model ranking (R) process for selecting the required model (M) based on the identified application criteria (C). For example, R1 is sorted based on the multi-objective application criteria, C1 and C3, while R2 only necessitates criterion C2 as an application requirement.

6.2 Model development phase

In this section, we discuss the design and development decisions required for anomalous link detection models.

6.2.1 Selected data representations

We consider four data representations. The first traceset, $TRACESET1$, used for training the model is represented by the raw time series that converts each link into a single feature vector containing 300 features. The second traceset, $TRACESET2$, is obtained via the FFT transformation of the raw time series, summing up to nearly 150 features. The third and fourth tracesets are computed by encoding the previous two tracesets into a reduced representation with 4 dimensional representation. Explicitly, $TRACESET3$ is generated by compressing $TRACESET1$ and $TRACESET4$ is generated by compressing $TRACESET2$ using an autoencoder.

Dataset transformation Similar to [2], we consider Rutgers [3] as our real-world measurement dataset containing records from 29 nodes, each of which contains 300 measurements. Despite the fact that each link is measured with five different noise levels, we assume that each measurement is recorded for different links, and that they do not have any correlation. Over this real-testbed dataset, the four type of anomalies proposed in [2] are synthetically injected. We only considered better links reducing our dataset from 4 060 to 2 123 ($\approx 52\%$) of independent links without packet loss. Similar to [2], we arbitrarily employed one anomaly type at a time over 33% of those links, at which the anomaly is injected according to guidelines in Table 6.1, and the other links are kept untouched.

Table 6.1: Synthetic anomaly injection method.

Type	Links	Affected	Appearance	Persistence
SuddenD	2 123	33% (700)	once, [200$^{\text{th}}$, 280$^{\text{th}}$]	for ∞
SuddenR			once, [25$^{\text{th}}$, 275$^{\text{th}}$]	for [5, 20]
InstaD			on $\approx$1% of a link	for 1 datapoint
SlowD			once, [1$^{\text{st}}$, 20$^{\text{th}}$]	for [150, 180]†

† $\text{RSSI}(x, \text{start}) \leftarrow \text{RSSI}(x) + \min(0, -\text{rand}(0.5, 1.5) \cdot (x - \text{start}))$

The details of anomaly injection method, such as at what packet range they appear and how long they persist, are provided in Table 6.1 and are kept the same as provided by the authors of [2, Table 1].

6.2.2 Selected ML techniques

A ML model is expected to distinguish between anomalous and ordinary behaviours of a link, thus requires to solve a binary classification problem. In the following, we elaborate on three supervised and three unsupervised ML techniques leveraged for our analyses.

6.2.2.1 Supervised techniques

In supervised technique, all anomaly data are labelled to train the model resulting with an inferred function, which then can be utilized for mapping unseen instances, respectively.

To evaluate the performance of the supervised models, we opt for a set of supervised ML techniques leveraging one representative algorithm from three distinct classes; i) Logistic Regression (LR) from Regression Analysis, ii) Random Forest (RForest) from tree ensemble class and iii) Support Vector Machines (SVM) from kernel-method class.

For each of three supervised ML techniques, we use 5-times Stratified K-Fold cross validation approach in order to ensure credible results, while this is not needed for unsupervised techniques.

6.2.2.2 Unsupervised techniques

In many practical applications, producing a reliable training (labelled) dataset is expensive and it can solely cover the type of anomalies that are present in the training dataset, which then cannot cope with the abnormal link behaviours in a comprehensive manner. Motivated by this, training a ML model in an unsupervised way is more practical, where learning from patterns of the overall link operations so as to distinguish the abnormal behaviours of a link from the anticipated behaviours is provoked, which is referred to as the automated detection of an anomaly [4] using ML models.

Therefore, a set of unsupervised ML techniques is also contemplated for developing anomaly detection models [5], where we select one representative algorithm from three distinct classes; i) Local Outlier Factor (LOF) from Nearest Neighbour (NN) class, ii) Isolation Forest (IForest) from tree ensemble class and iii) one-class Support Vector Machines (OC-SVM) from kernel-method class.

6.2.3 Choice of parameters for tuning ML models

In this section, we tune each ML model with the most relevant subset of available parameters exposed by their respective implementations.

For each of the ML techniques discussed in Section 6.2.2, Table 6.2 lists the respective implementations and parameters used in our experiments. For instance, for logistic regression we use the LogisticRegression implementation available in the Python Scikit Learn toolbox. As the LogisticRegression implementation enables setting 12 different parameters that influence the final model, we generally select standard values that have been proven to work on large number of cases and datasets by the ML community. Moreover, we identify selected parameters that should be optimized such as the regularization strength C in this case. For C, rather than using the standard value $C = 1.0$ we search for the best configuration by assessing an array of possible values $C \in [10^{-3}, 10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}]$ and ultimately select the best performing regularization factor.

The implementations chosen for the remaining algorithms also include over ten possible input parameters. For LOF, we vary the number of neighbours, algorithm and leaf size for finding the best performing model. Moreover, for RForest and IForest, we vary the number of base estimators, whereas having both used the RBF kernel we vary the regularization factor C for SVM, and the kernel, kernel coefficient *gamma* and *nu* (limiting bound parameter) for OC-SVM.

As some of the models are sensitive to scaling methods, we also consider training on traceset that is; Method-1) not scaled, Method-2) scaled to zero mean, Method-3) scaled to zero mean and unit variance, Method-4) scaled to zero mean where only Q1-Q3 quantiles are considered, Method-5) scaled zero mean and unit variance where only Q1-Q3 quantiles are considered, and Method-6) scaled using min-max to limit values between 0 and 1.

Having provided 6 scalers applied to 4 main tracesets along with 4 anomaly types, and implemented with 6 ML algorithms each of which is tuned with, in order of Table 6.2, $1(6) = 6, 1(7) = 7, 3(6, 2, 2) = 24, 4(6, 3, 4, 2) = 144, 1(7) = 7, 3(6, 2, 2) = 24$ different set of parameters, resulting in 212 distinct parametrized models per traceset, leveraging LR, RForest, SVM, LOF, IForest and OC-SVM, respectively. Again, note that $3(6, 2, 2)$ represents 3 different set of parameters, each of which containing 6,2,2 parameter values resulting in 24 distinct parameter combinations. Therefore, it is clear that our analyses

Table 6.2: ML algorithms and their associated parameters as provided in [2, Table 2]

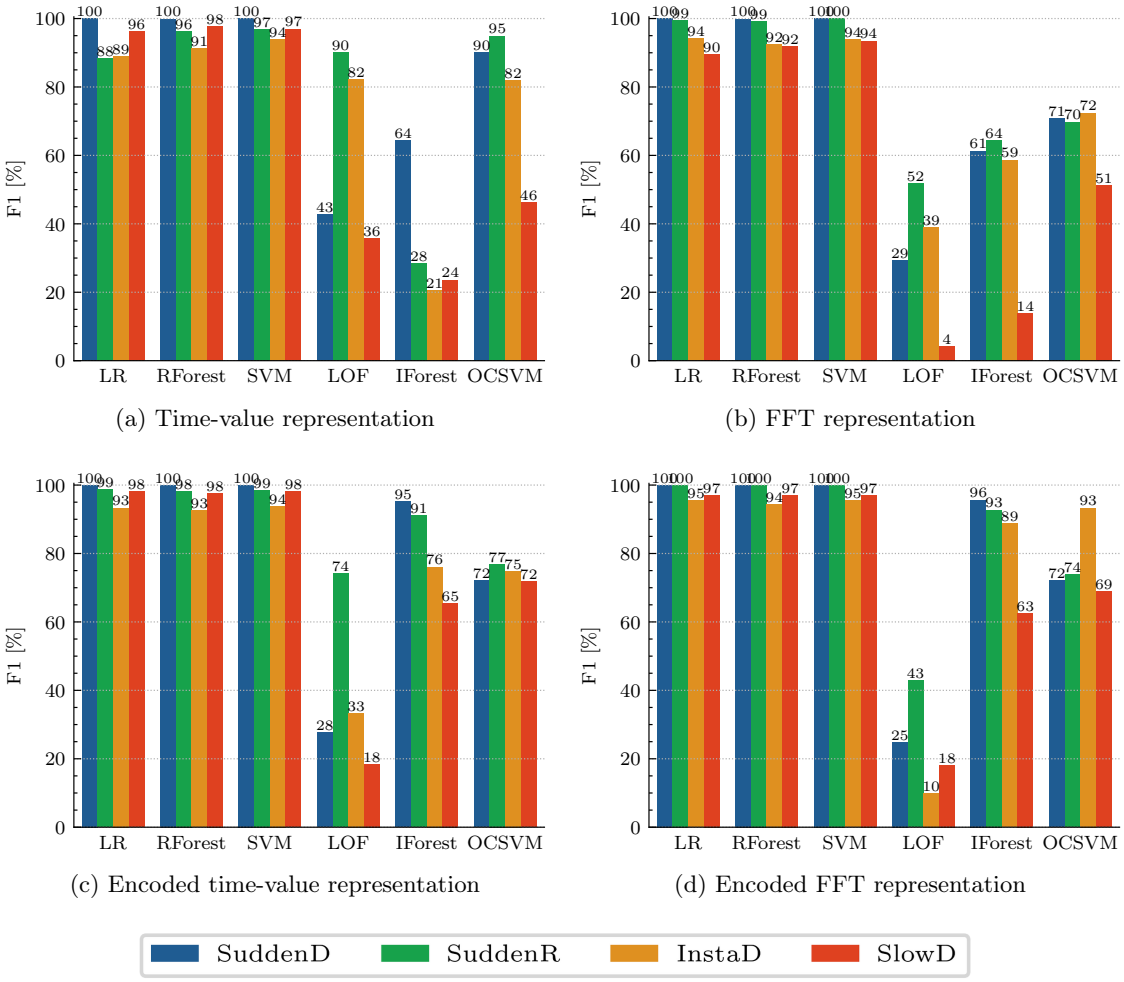
Approach	Technique	Implementation	Parameters and their range
Supervised	Logistic Regression (LR)	LogisticRegression from sklearn	penalty='l2', dual=False, tol=1e-4, C=(1e-3, 1e-2, 1e-1, 1., 10., 100.) fit_intercept=True, intercept_scaling=1, class_weight=None, solver='lbfgs', l1_ratio=None
	Random Forest (RForest)	BaggingClassifier from sklearn	base_estimator=None, n_estimators=[10, 20, 30, 40, 50, 70, 100], max_samples=1.0, max_features=1.0, oob_score=False, intercept_scaling=1,
	Support Vector Machine (SVM)	SVC from sklearn	C=(1e-3, 1e-2, 1e-1, 1.0, 10., 100.), kernel=('linear', 'rbf'), gamma=('auto', 'scale'), tol=1e-3, decision_function_shape='ovr', break_ties=False,
Unsupervised	Local Outlier Factor (LOF)	LocalOutlierFactor from sklearn	n_neighbors=[5, 10, 20, 40, 50, 80], algorithm=['ball_tree', 'kd_tree', 'brute'], leaf_size=[10, 30, 50, 80], p=[1, 2] metric_params=None, contamination="auto",
	Isolation Forest (IForest)	IsolationForest from sklearn	n_estimators=[10, 20, 30, 40, 50, 70, 100], max_samples='auto', contamination='auto', max_features=1.0, bootstrap=False,
	Support Vector Machine (OC-SVM)	OneClassSVM from sklearn	nu=[0.10, 0.3, 0.5, 0.70, 0.90, 1.0], kernel=('linear', 'rbf'), gamma=('auto', 'scale'), coef0=0.0, tol=1e-3,

are conducted with $4 \cdot 6 \cdot 4 \cdot (212) = 20,352$ possible distinct models given by Eq. (6.1) in Section 6.1.

6.3 Model selection phase

As shown at the end of Section 6.2.3, the development phase of the anomaly detection models yields 20,352 distinct models. Depending on the application requirement and its associated criteria, a model selection followed by a ranking based on F1 score would be plausible. In our analyses, we rely on F1 score.

Fig. 6.3 provides the best performing models using F1 score as the ranking criterion. Each subfigure corresponds to a data representation, each group of bars depicts the algorithm listed on the x-axis, whereas each bar corresponds to a particular anomaly type and the height of the bar represents F1 scores, as seen on the y-axis. It can be readily seen from Figs. 6.3(c) and (d) corresponding to the encoded representations that usually models perform a higher F1 score, when compared to Figs. 6.3(a) and (b) revealing the superiority of the encoded representations. This is especially clear for the unsupervised models where F1 score reaches up to 95% for the encoded time-value representation and up to 96% for the encoded FFT representation with high deviations in the considered ML techniques of both representations.



(a) Time-value representation

(b) FFT representation

(c) Encoded time-value representation

(d) Encoded FFT representation

Figure 6.3: Performance comparison of the best performing models based on F1 scores considering all anomaly types and data representations.

Through the analysis of the trained models, we notice that scaling methods aid in achieving better detection performance. However, scaling has to be performed per-link rather than on the entire dataset. In particular, using standard scaler or robust scaler consistently outperforms the models with no scaling.

For detecting *SuddenD* anomalies, all three supervised models are suitable as they achieved near perfect F1 score for all four datasets considered. On the other hand, the most suitable unsupervised approach is IForest reaching an F1 score of around 95% when it is employed over TRACESET3 and TRACSET4.

For detecting *SuddenR* anomalies, the most suitable supervised models are RForest and SVM whose performances are higher on TRACESET2 with an F1 score of about 99%, when compared to around 97% on TRACESET1. Utilizing encoded representations, namely TRACESET3 and TRACESET4, yield only slight improvement for SVM and RForest, albeit contribute 10 percentage points performance improvement for LR, which is just below RForest and SVM models. Furthermore, the most suitable unsupervised algorithms for detecting SuddenR are IForest and OC-SVM, where IForest performs best on encoded TRACESET3 with an F1 score of 91% and OC-SVM performs best on non-encoded TRACESET1 with an F1 score of 95%.

For detecting *InstaD* anomalies, all three supervised algorithms are suitable attaining an F1 score of around 93% on all four datasets. For unsupervised approaches, the highest F1 score, 93%, is achieved by OC-SVM, followed by the IForest model with an F1 score of 89% both applied to TRACESET4. It is demonstrated that a combination of encoding and FFT representation yields the best F1 score for detecting InstaD anomalies.

For *SlowD* anomalies, all the considered supervised models, regardless of being trained with a particular traceset, are suitable for efficiently detecting SlowD anomalies with an F1 score around 97%, while the models trained with TRACESET3 are the most suitable ones, especially when a well-balanced anomaly detection performance is needed. On the contrary, unsupervised approaches are less suitable for detecting SlowD anomalies, as OC-SVM on TRACESET3 reaches at only an F1 score of 72%, while IForest attains the highest score of 65% on the same traceset.

6.4 Summary

In this chapter we show that ML techniques can be effectively leveraged for automatically identifying abnormal behaviours of wireless links. We first elaborated on the development phase of ML models for efficient detection of anomalous wireless links, including the details of data transformation, data scaling and parameter tuning. Then, we discussed model ranking and selection phase using F1 score based on the application requirements and their associated criteria. We showed that the model development, tuning and selection are sophisticated processes, especially when various data representations, different anomaly types, a diverse set of scaling methods, a variety of ML models and their associated distinct parameters with a large set, are considered.

We demonstrated that the best performing models based on F1 score are, in most cases, trained on data representations with autoencoders, i.e., TRACESET3 and TRACESET4. With a goal to achieve high accuracy, a choice of supervised models over unsupervised would prevail. On average, amongst 20,352 available models directly selecting the supervised SVM model for detecting any anomaly type, that is trained over encoded representation of time series and FFT, would be the most plausible decision.

References

[1] M. Kulin, C. Fortuna, E. De Poorter, D. Deschrijver, and I. Moerman, "Data-driven design of intelligent wireless networks: An overview and tutorial," *Sensors*, vol. 16, no. 6, p. 790, 2016.

[2] G. Cerar, H. Yetgin, B. Bertalanic, and C. Fortuna, "Learning to detect anomalous wireless links in iot networks," *IEEE Access*, vol. 8, pp. 212 130–212 155, 2020. DOI: 10.1109/ACCESS.2020.3039333.

[3] S. K. Kaul, I. Seskar, and M. Gruteser, *CRAWDAD dataset rutgers/noise (v. 2007-04-20)*, Downloaded from https://crawdad.org/rutgers/noise/20070420/RSSI, traceset: RSSI, Apr. 2007. DOI: 10.15783/C7B59W.

[4] A. Cook, G. Mısırlı, and Z. Fan, "Anomaly detection for IoT time-series data: A survey," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6481–6494, Jul. 2020.

[5] C. C. Aggarwal, "Outlier analysis," in *Data mining*, Springer, 2015, pp. 237–263.

Chapter 7

Conclusions and Future Work

In this dissertation, we have studied two applications of link state information. One is concerned with link quality estimation, which accurately predicts the future state of the wireless link based on current and past link conditions learned from representative datasets. The other focuses on link anomaly detection, which attempts to quickly, proactively, and accurately identify unexpected link behavior regardless of its cause. Using a systematic quantitative approach, we have shown that the ML approach excels in both link quality estimation and anomaly detection.

In a comprehensive review of the existing data-driven LQEs, we thoroughly examined their approaches and showed how ML-based LQE research has gained momentum over time. Overall, the research community has shown remarkable improvements toward better estimators, but only a small number of papers provide all the implementation details. This fact, together with the use of non-standard metrics, makes reproducibility and comparability of different solutions difficult, and a fully fair comparison of LQE models impossible. For this reason, we advocate the use of standardized metrics and evaluation of the proposed data-driven LQE estimators using open data traces to facilitate the comparison of new link quality models.

To gain a deeper insight into how each step of the KDP affects the final performance, we conducted a systematic quantitative study on the impact of design steps on the ML-based LQE performance. To make our study reproducible, we used openly available wireless link traces suitable for LQE research. We showed that data cleaning, interpolation, resampling, and synthetic feature creation can have a greater impact on the overall classification performance than the selection of the ML algorithm. For instance, the classification performance metrics have shown that adding synthetic features yields up to 6% overall and up to 49% for minority class detection improvement, while resampling strategies have shown over 20% improvement for minority class with almost zero penalty to other classes.

In studying anomalies in wireless links, we identified four types of anomalies. For each anomaly type, we determined the symptoms from the user's perspective and the possible causes. In a systematic quantitative study, we used six reference ML algorithms along with four different representations of a time series data to accurately detect anomalies. Our study has shown that in terms of data, none of the four time series representations clearly dominates the others in terms of anomaly detection performance. However, we have shown that automatically generated encoded representations using autoencoders improve anomaly detection by up to 40% compared to their non-encoded counterparts.

7.1 Summary of Contributions

As part of this dissertation, we carried out an in-depth and comprehensive survey on research of data-driven LQEs, which has not been available before and represents our first contribution **C1**. With the main focus on ML-based approaches, we investigated how authors proceed in data collection, data preparation, and model building to meet the LQE requirements for a certain application. On a subset of proposed LQEs, which used sufficiently common metrics and similar goals, we made per-class performance comparison. Finally, we summarized lessons learned from the comprehensive overview of related literature and examination of datasets suitable for LQE research, and we provided guidelines to the industry and research community for developing ML-based LQE models along application quality aspects and collection of generic trace-sets.

This dissertation provides a systematic investigation of the impact of data representation, feature space, and pre-processing on data-driven approaches to wireless link quality estimation and wireless link anomaly detection, which represents our second contribution **C2**. Our quantitative study, conducted on a selected openly available dataset, shows how approaches to raw data cleaning and interpolation, new feature generation, feature selection, resampling, and window size selection have a significant impact on the overall performance of the ML-based LQE model. Furthermore, we show how synthetic features, feature selection and resampling strategy significantly improve minority class recognition when using an imbalanced dataset.

As our third contribution **C3**, we developed a new supervised classifier for link quality estimation. The novel tree-based classification model was trained and evaluated on cross-layer data from a representative real-world wireless network. We used the previously obtained lessons learned on pre-processing steps and we proposed modifications to classification fairness of minority classes, which were previously classified poorly due to an imbalanced dataset, while providing minimal impact on other classes. The novel classifier achieves state-of-the-art performance measured by standard classification evaluation metrics, and exhibits low training time.

As our fourth contribution **C4**, we developed new supervised and unsupervised ML-based anomaly detection classifiers for wireless links and evaluated their performance with respect to two threshold-based approaches using four different time series representations. As a minor additional contribution, we performed an in-depth analysis of the selected algorithms with explainable AI approaches, explaining their decision process and the contributions of certain data features.

As our fifth contribution **C5**, we proposed the use of autoencoder based on unsupervised deep learning neural networks for encoding input features. The feature space reduction obtained by this approach along with raw input data de-noising capabilities improved the anomaly detection performance by up to 40% for the selected reference ML-based algorithms compared to using raw input data directly.

7.2 Future Work

We conclude the thesis with a discussion of open challenges and possible directions for future research.

In this dissertation, we studied existing data-driven LQEs and presented our own novel ML-based models for link quality estimation and anomaly detection. The main and most important next step of the studied and proposed ML-based models is their generalization. That is, their training data and application are in most cases limited to a single frequency band, to only one technology and homogeneous networks. Here, we propose the exploration

of more generalized and versatile ML-based LQEs models, which however requires collection of generic trace-sets from representative wireless networks.

Next, our research of anomaly detection in wireless links was limited to detecting one type of anomaly at a time, making the investigation more tractable. As an important future extension, we propose the exploration of an all-round link anomaly detection model that is capable of detecting all four types of link anomalies and even classifying them.

As another enhancement in anomaly detection approaches we propose investigation in replacing the currently used offline training approach, which tends to be superior in performance, with online approaches which adapt faster and do not require intensive re-training of the model for each sample batch. Resulting online anomaly detection models may be better suited for stream and batch processing on wireless devices with sufficient computational power.

The use of LQEs and link anomaly detection approaches built on deep neural networks have shown promising results in this study. However, their potential is limited by how well data scientists can design individual parts of neural networks. To overcome this limitation, a promising new research area called Automated Machine Learning (AutoML) has emerged, with the idea to automate the entire process of knowledge discovery, designing neural networks, and satisfying higher-level constraints such as ensuring classification fairness. It would be interesting to see such solutions applied to LQEs and link anomaly detection problems.

Bibliography

Publications Related to the Thesis

Journal Articles

G. Cerar, H. Yetgin, M. Mohorčič, and C. Fortuna, "Machine learning for wireless link quality estimation: A survey," *IEEE Communications Surveys Tutorials*, pp. 1–1, 2021. DOI: 10.1109/COMST.2021.3053615.

G. Cerar, H. Yetgin, B. Bertalanič, and C. Fortuna, "Learning to detect anomalous wireless links in iot networks," *IEEE Access*, vol. 8, pp. 212 130–212 155, 2020. DOI: 10.1109/ACCESS.2020.3039333.

Conference Papers

G. Cerar, H. Yetgin, M. Mohorčič, and C. Fortuna, "Learning to fairly classify the quality of wireless links," in *16th Conference on Wireless On-demand Network Systems and Services (WONS 2021)*, 2021.

——, "On designing a machine learning based wireless link quality classifier," in *2020 IEEE 31st Annual International Symposium on Personal, Indoor and Mobile Radio Communications*, 2020, pp. 1–7. DOI: 10.1109/PIMRC48278.2020.9217171.

G. Cerar, M. Mohorčič, and C. Fortuna, "Data driven link quality estimation," in *Knjiga povzetkov : science of the future how to stay up-tod-date with your research! = Book of abstracts*, M. Topole, T. Turk Dermastia, M. Dežman, B. Škrlj, A. Jurov, K. Bačnik, J. Masten, Ž. Marinko, P. Jovičević Klug, R. Pahič, I. Rybkin, J. Černilogar, and A. Kikaj, Eds., Ljubljana: Jožef Stefan International Postgraduate School, Jožef Stefan Institute, 2019, p. 40.

G. Cerar and C. Fortuna, "The impact of feature selection on the performance of link quality estimation," in *Zbornik = Proceedings*, M. Dežman, A. Pecman, K. Bačnik, J. Masten, M. Topole, M. Bergant, M. Cevzar, A. Jurov, B. Škrlj, and T. Turk Dermastia, Eds., Ljubljana: Mednarodna podiplomska šola Jožefa Stefana, = Jožef Stefan International Postgraduate School, Inštitut Jožef Stefan, = Jožef Stefan Institute, 2018, p. 37.

Other Publications

Journal Articles

C. Fortuna, A. Bekan, T. Javornik, G. Cerar, and M. Mohorčič, "Software interfaces for control, optimization and update of 5g machine type communication networks," *Computer Networks*, vol. 129, pp. 373–383, 2017, Special Issue on 5G Wireless Networks for IoT and Body Sensors, ISSN: 1389-1286. DOI: https://doi.org/10.1016/j.comnet.2017.06.015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128617302578.

Conference Papers

A. Hrovat, G. Cerar, G. Gardašević, D. Vasiljević, and T. Javornik, "Testing the Interoperability of the Joint Scheduling and Routing Algorithm for IIoT Applications," in *Proceedings of the Twenty-eighth International Electrotechnical and Computer Science Conference (ERK 2019)*, Društvo Slovenska sekcija IEEE, 2019, pp. 40–43. [Online]. Available: https://erk.fe.uni-lj.si/2019/ERK19.pdf.

G. Cerar, A. Švigelj, M. Mohorčič, C. Fortuna, and T. Javornik, "Improving CSI-based Massive MIMO Indoor Positioning using Convolutional Neural Network," in *European Conference on Networks and Communications (EuCNC 2021)*, 2021.

Biography

Gregor Cerar received his Bachelor's (2013) and Master's (2016) degrees from the Faculty of Electrical Engineering of the University of Ljubljana, where he completed the Telecommunications study programme. In 2016, he received a Young Researcher Grant from the Slovenian Research Agency and enrolled in doctoral studies at the Jožef Stefan International Postgraduate School in Information and Communication Technologies' study programme. At the same time, he joined the Department of Communication Systems (E6) at the Jožef Stefan Institute as a young researcher.

Under the supervision of Prof. Dr. Mihael Mohorčič and Dr. Carolina Fortuna, his work began with embedded systems, where he got a grip on recent technologies used in wireless sensor networks. From there on, his research converged toward wireless communications, especially wireless link quality estimation and wireless link anomaly detection, where he uses statistical and machine learning approaches.

He presented some of his research milestones toward PhD at national (e.g. ERK, IPSSC) and international conferences (e.g. PIMRC, WONS, EuCNC). He attended several international summer/training schools. His most recognizable achievement was winning the first place for the best solution for solving a task with deep neural networks.

During his research work, he also participated in the H2020 project NRG5. He is also one of the core maintainers and contributors to the LOG-a-TEC 3.0 experimental wireless network testbed.