Machine Learning for Wireless Link Quality Estimation: A Survey

Gregor Cerar[1,2], Halil Yetgin[1,3], *Member, IEEE*, Mihael Mohorčič[1,2], *Senior Member, IEEE*, and Carolina Fortuna[1]
[1]Department of Communication Systems, Jožef Stefan Institute, SI-1000 Ljubljana, Slovenia.
[2]Jožef Stefan International Postgraduate School, Jamova 39, SI-1000 Ljubljana, Slovenia.
[3]Department of Electrical and Electronics Engineering, Bitlis Eren University, 13000 Bitlis, Turkey.
{gregor.cerar | halil.yetgin | miha.mohorcic | carolina.fortuna}@ijs.si

***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 paper, 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.

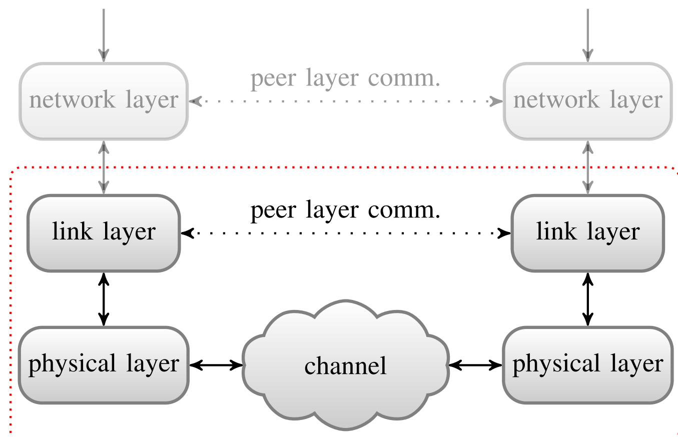


Fig. 1: The unified model of data-driven LQE comprising of physical layer (layer 1) and link layer (layer 2).

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. 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 paper, 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 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 paper 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) 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 paper can be summarized as follows.

[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.

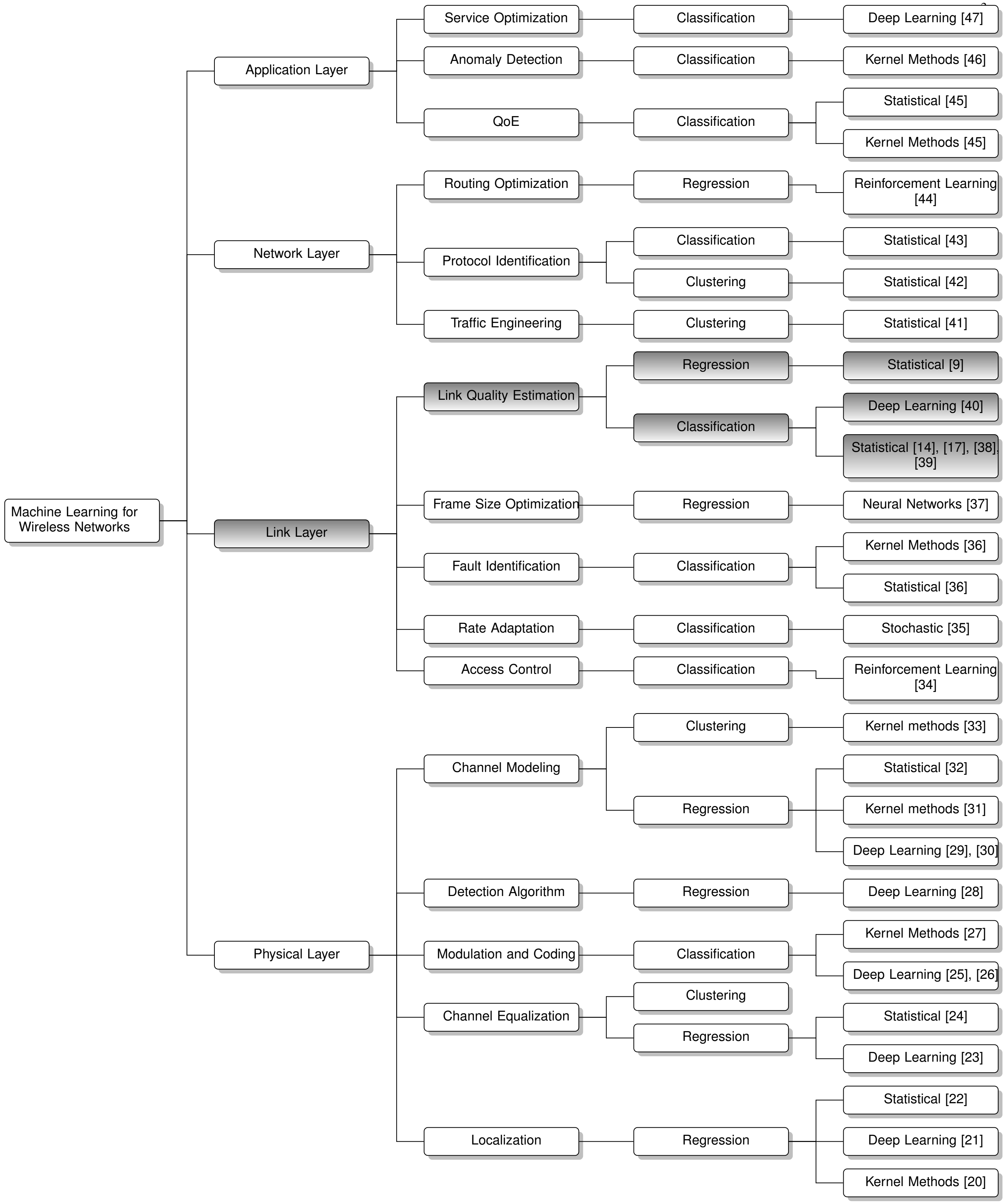


Fig. 2: Layered taxonomy of machine learning solutions for wireless communication networks.

- 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

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

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 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 paper 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

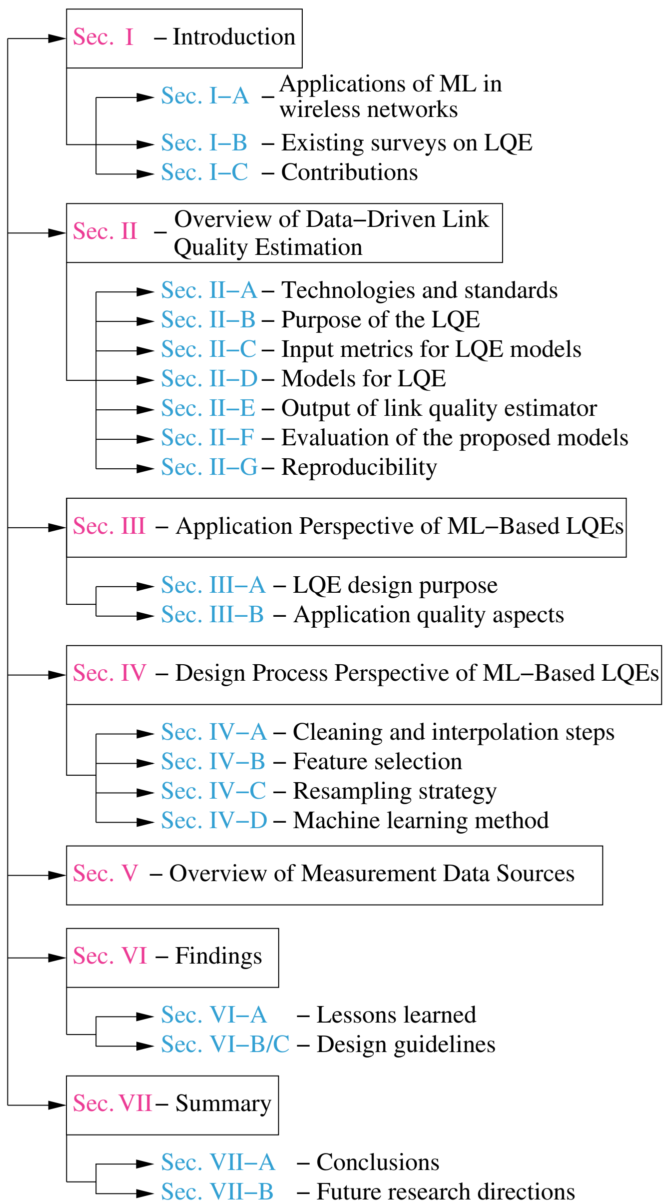


Fig. 3: Structure overview of this survey paper.

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 input(s), model and output. The last two columns summarize statistical aspects of the data traces and their public availability

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

Traditional approach	1996	Link loss of pre-WiFi networks using Markov model [5].
	1998	Transport layer protocol that uses link loss notification [7].
	2003	Improved multi-hop routing based on link quality model [8].
	2007	Four bit cross-layer (phy, link and network) information based LQE [10].
	2010	Triangle, a PRR, LQI and SNR based LQE [12].
	2010	Fuzzy logic based LQE [13].
Machine learning	2011	Bayes, regression and neural network LQE classification [17], [38].
	2015	Topological features + SVM, k-NN, regression trees, gaussian regression LQE [68].
	2017	Reinforcement learning-based LQE [69].
	2017	LQE for a mmWave base station handover system [70].
	2019	Stacked autoencoder + SVM based link quality estimator [40].
	2019	Satellite image + network metrics LQE using SVM [6].

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, whereas others focused on routing protocol performance. This group of papers proposed a novel link quality estimators

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/experi-ment; 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/ex-periment, 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.

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.

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 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 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.

2) 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 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 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

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						✓					

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 *et al.* [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 *et al.* [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 perceptron, 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

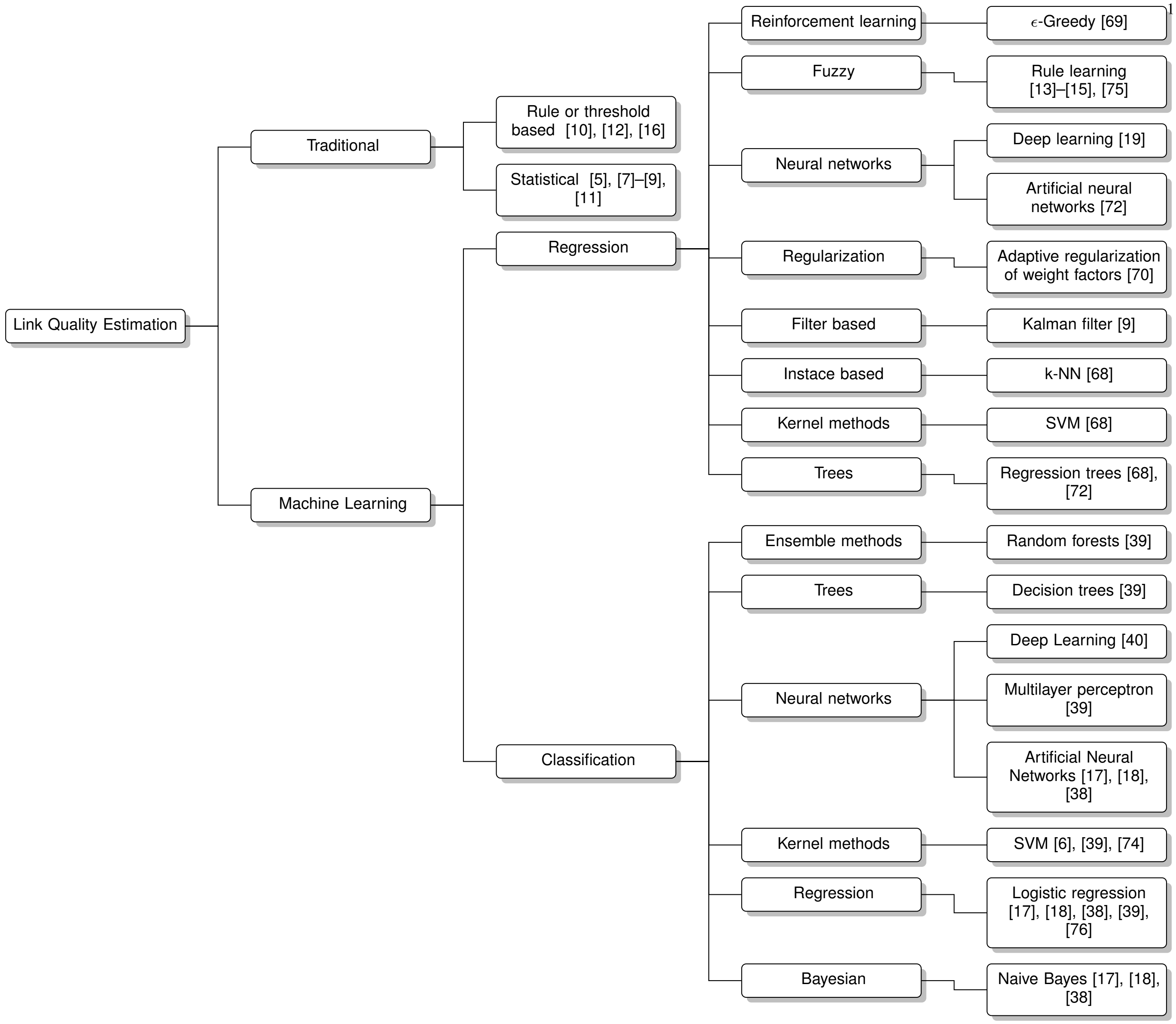


Fig. 5: Taxonomy of the LQE approaches using ML algorithms and traditional methods.

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.

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

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]	

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.

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

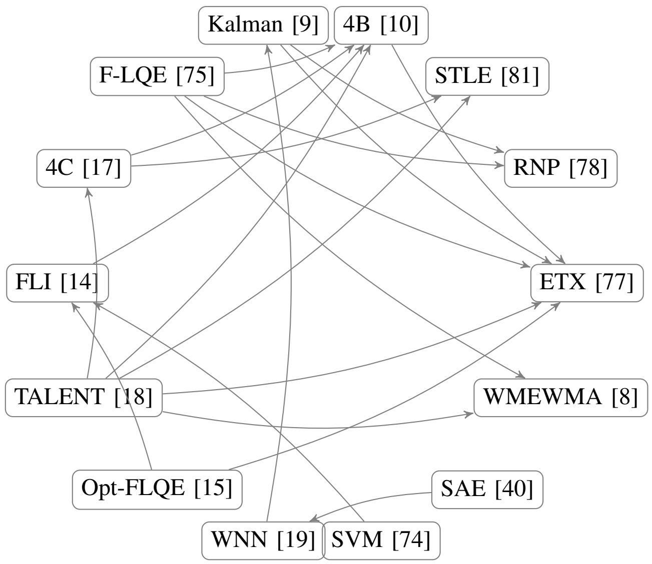


Fig. 6: Visualization of relationships for cross-comparison of the research papers with their corresponding evaluation metrics outlined in Table V.

shown in rows 8-10 of Table V, whereas the authors of [68], [72] use mean absolute error (MAE) as in rows 11 and 12 of the table. Some other research papers as in [5], [13], [15], [75] use 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 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,

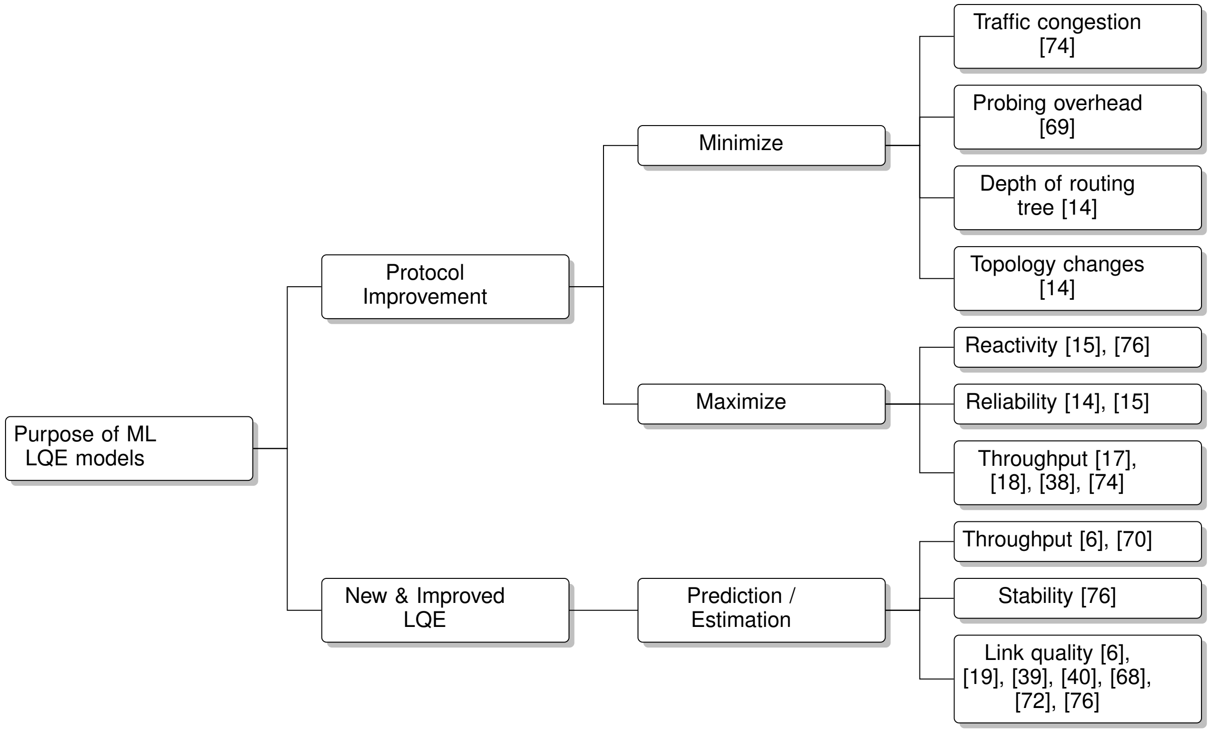


Fig. 7: Classification of the works by considering the purpose for which the ML LQE model was developed.

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 trace-sets. 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 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

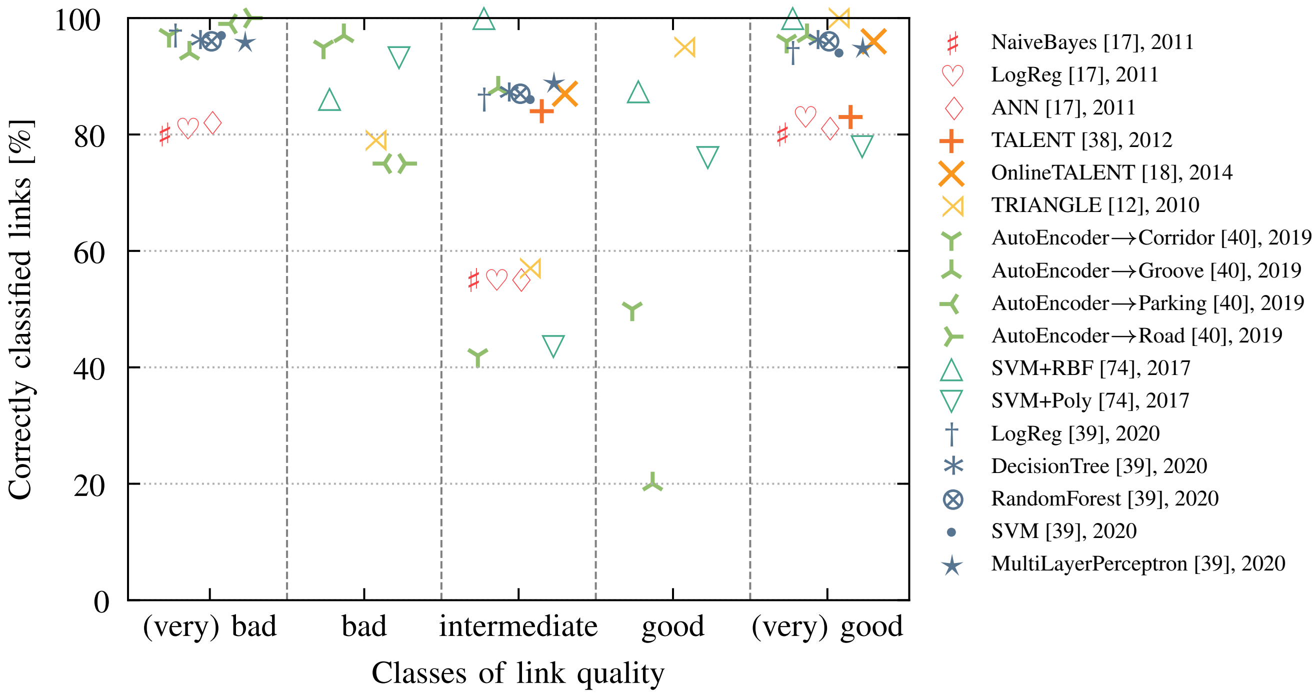


Fig. 8: Comparison of the wireless link quality classification performances throughout the surveyed papers.

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 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

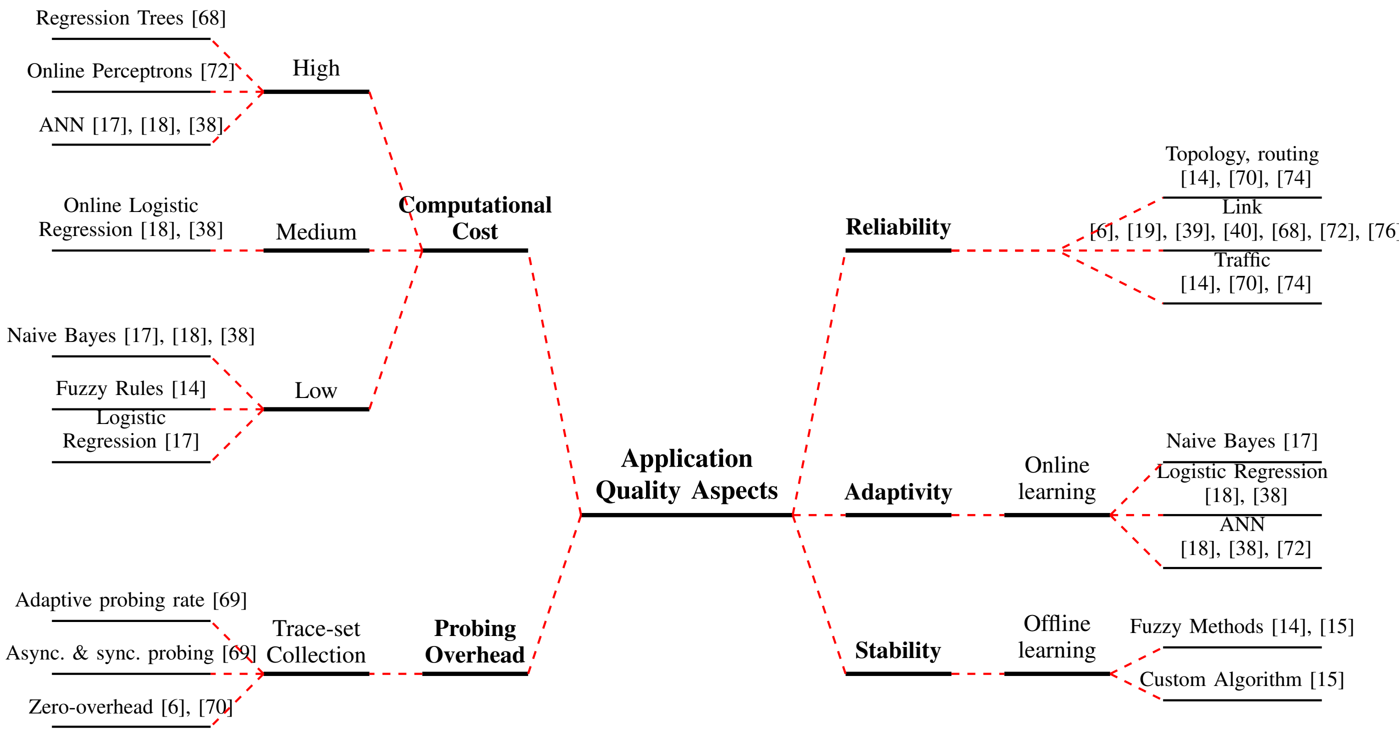


Fig. 9: Classification of the surveyed LQE papers by taking into consideration the identified application quality aspects.

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 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.

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				

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 pre-determined 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 classifies the studies based on the reported

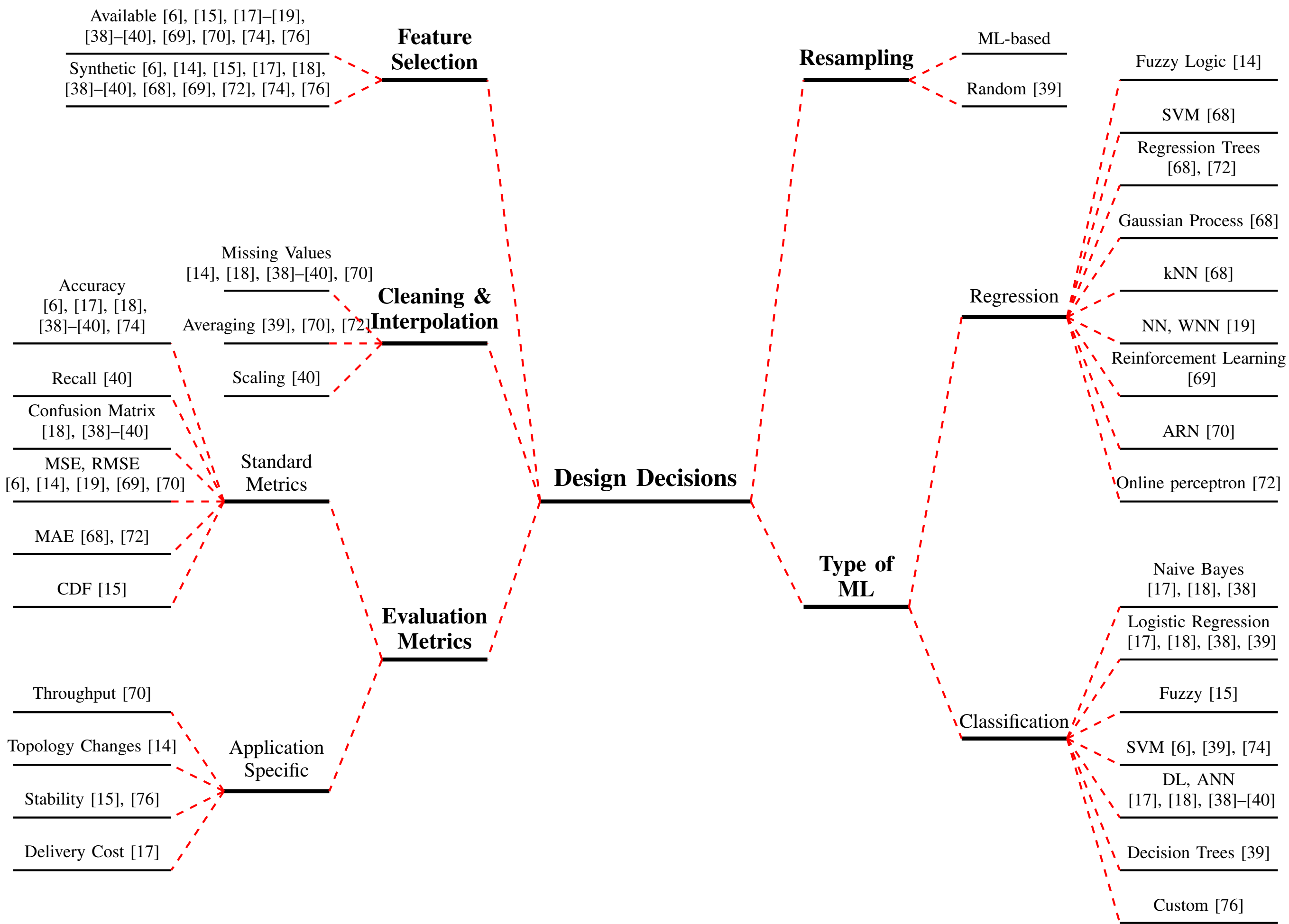


Fig. 10: Overview of the design decisions taken during the development of the ML-based LQE models for the relevant papers surveyed in Tables II and III.

design decisions taken while developing ML-based LQE models, namely cleaning and interpolation, feature selection, resampling 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.

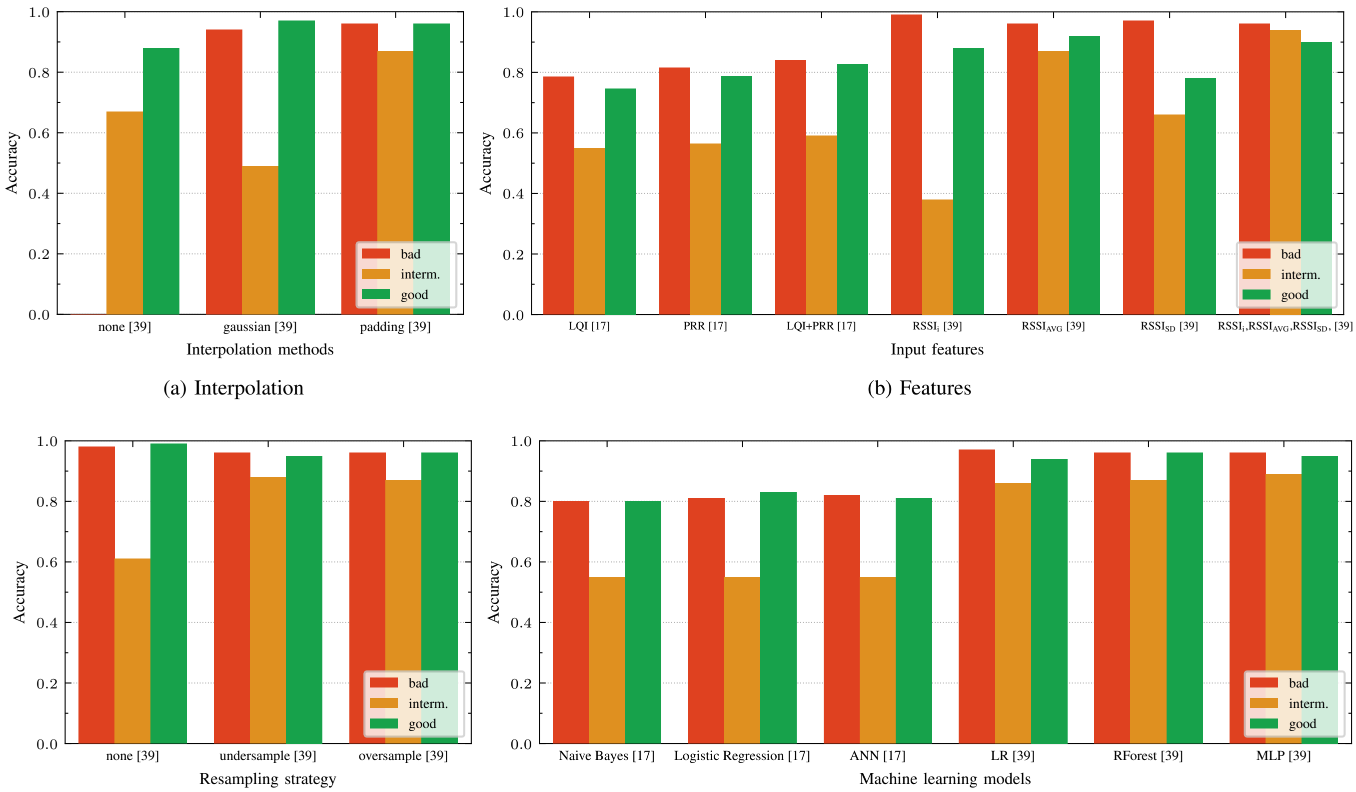
A. Cleaning & interpolation steps

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.

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

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. 11a 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



(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.

optimized for a particular scenario. More specifically, F1 score for no-interpolation is about 0.43 on the left lower part of the figure, then increases to 0.80 with Gaussian interpolation, and finally reaching 0.94 with constant interpolation (denoted by ”padding” in Fig. 11a) 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. 11b. 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 [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.

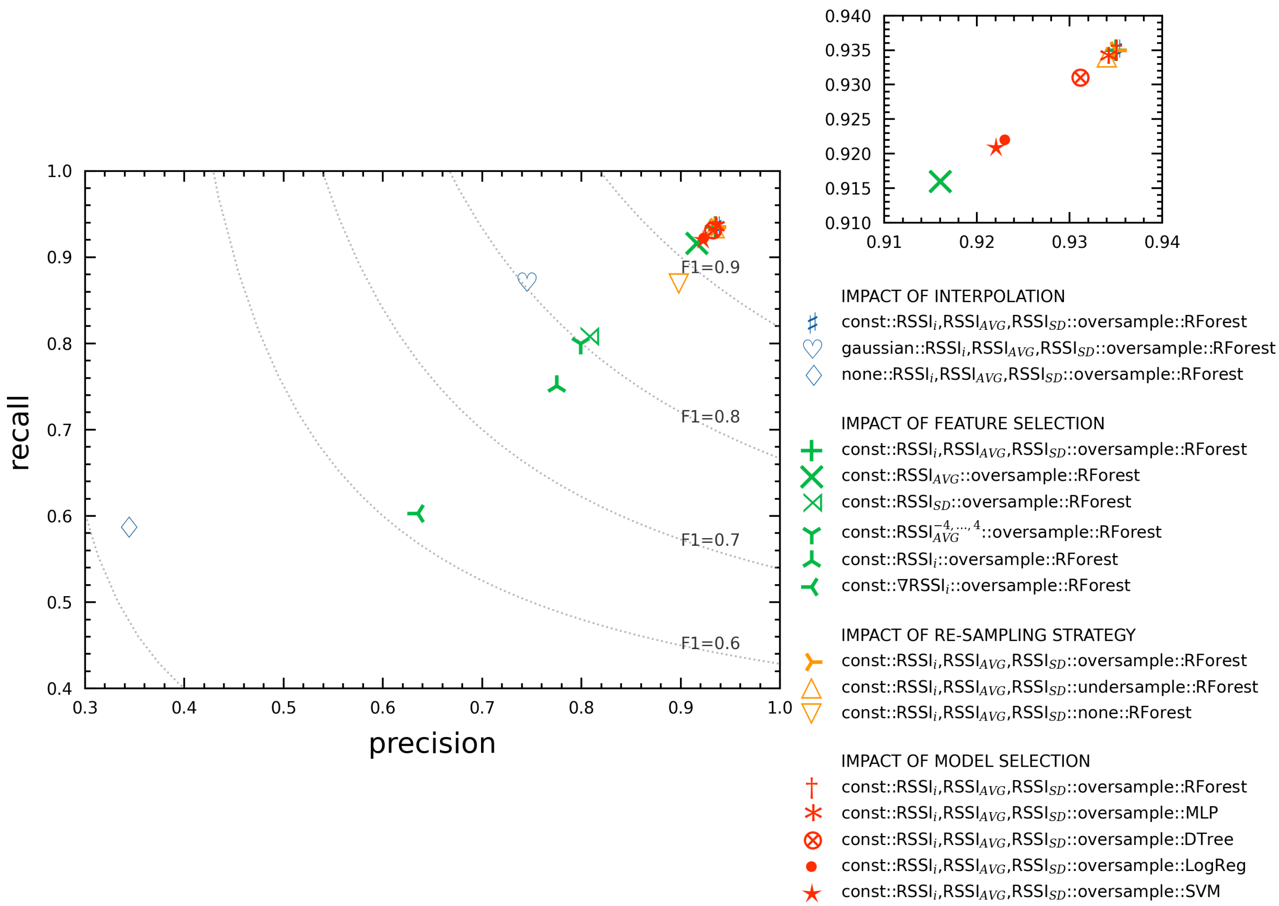


Fig. 12: Precision vs. recall performance trade-off for various design decisions including interpolation, feature selection, resampling and model selection, where the figure situated at the top-right corner is a zoomed-in portion of the closest region to F1=1 of the main figure.

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 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. 11c, 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.

Fig. 11d 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 paper, 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 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,

[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.

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.

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]	✓	✓	✓	✓		✓			

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 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

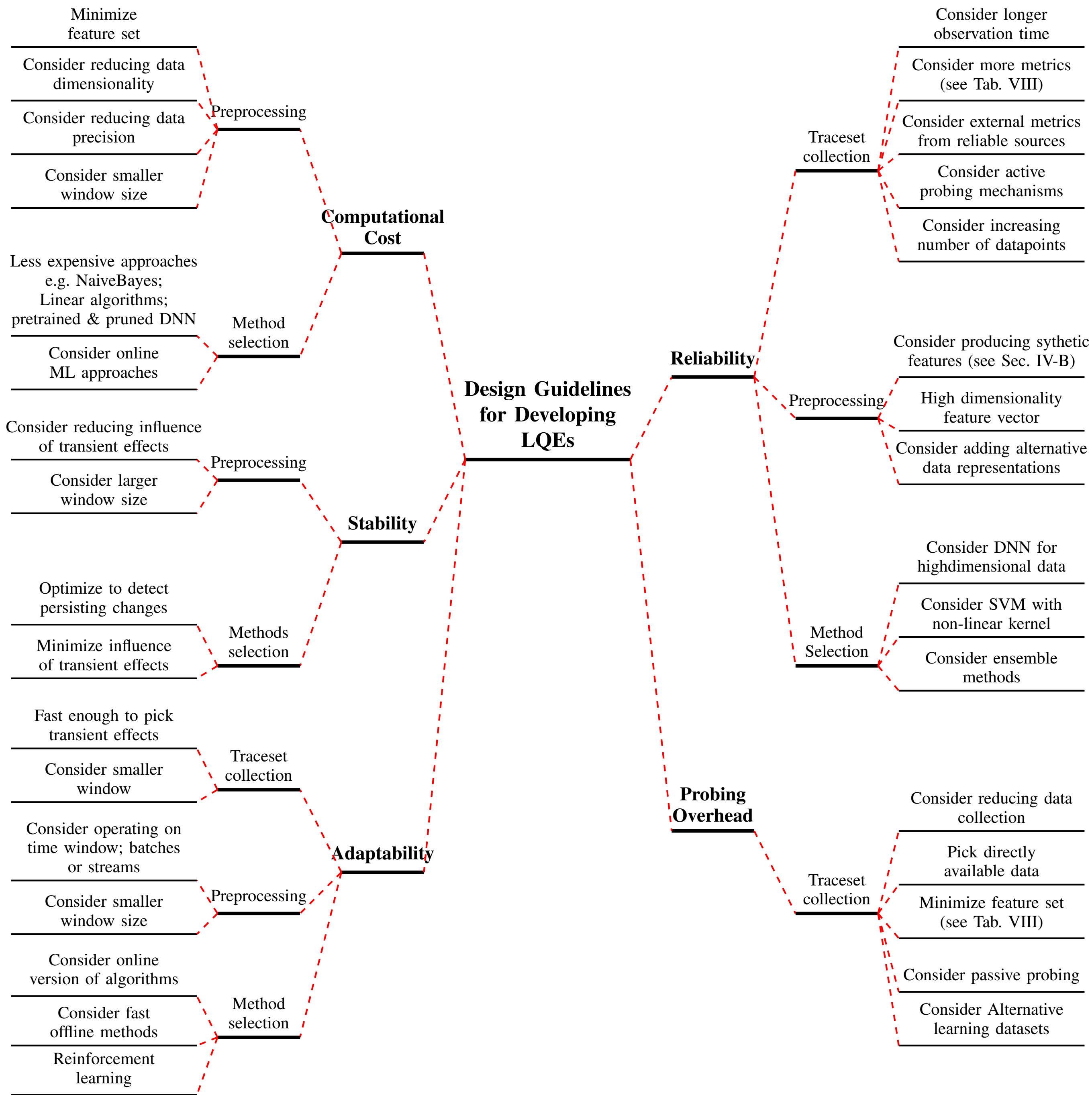


Fig. 13: Mind map representation of design guidelines for LQE model development.

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 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

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

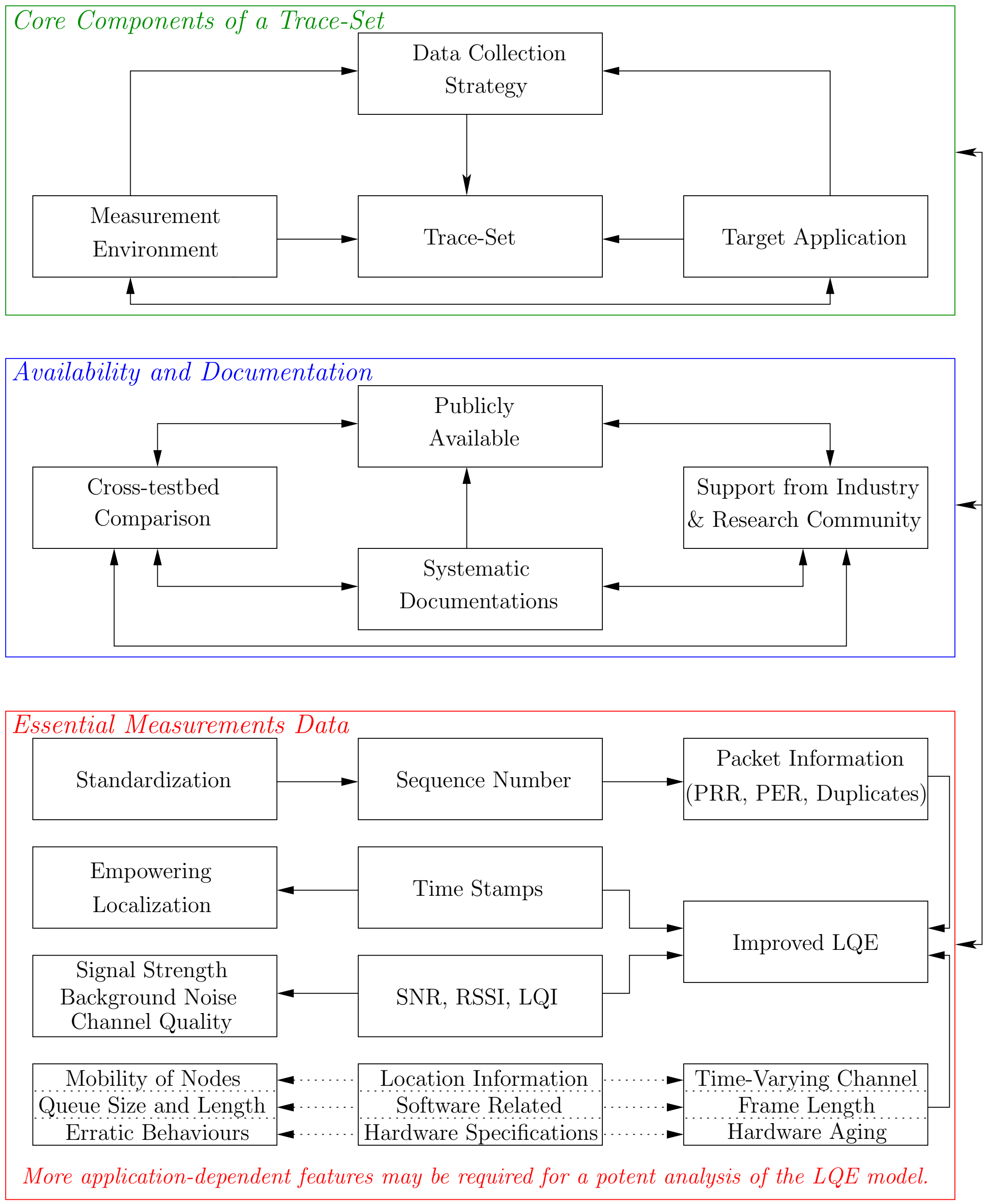


Fig. 14: Design guidelines recommended for the industry and research community to follow in order to design and collect trace-sets for the sake of LQE research.

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.

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. Conclusions

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.

Acknowledgment

This work was funded in part by the Slovenian Research Agency under Grants P2-0016 and J2-9232, and in part by the European Community H2020 NRG-5 project under Grant 762013. The authors would also like to thank Timotej Gale and Matjaž Depolli for their valuable insights.

Acronyms

4B	Four-Bit
4C	Foresee
AI	Artificial Intelligence
BER	Bit Error Rate
CDF	Cumulative Distribution Function
ETX	Expected Transmission count
F-LQE	Fuzzy-logic based LQE
FLI	Fuzzy-logic Link Indicator
KDP	Knowledge Discovery Process
LQ	Link Quality
LQE	Link Quality Estimation
LQI	Link Quality Indicator
MAE	Mean Absolute Error
ML	Machine Learning
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

REFERENCES

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

[2] 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.

[3] A. Zanella, "Best practice in rss measurements and ranging," IEEE Communications Surveys Tutorials, vol. 18, no. 4, pp. 2662–2686, 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 Communications Surveys Tutorials, vol. 19, no. 2, pp. 828–854, 2017.

[5] 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.

[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 ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN'19), Montreal, CA, 15-18 April 2019.

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

[8] 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.

[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 IEEE Global Telecommunications Conference (GLOBECOM'07), Washington, USA, 26-30 Nov. 2007, pp. 875–880.

[10] 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.

[11] K. Srinivasan, M. A. Kazandjieva, M. Jain, and P. Levis, "PRR is not enough," 2008. [Online]. Available: http://sing.stanford.edu/pubs/sing-08-01.pdf

[12] 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.

[13] N. Baccour, A. Koubâa, H. Youssef, M. B. Jamâa, D. Do Rosario, M. Alves, and L. B. Becker, "F-LQE: A fuzzy link quality estimator for wireless sensor networks," in European Conference on Wireless Sensor Networks, Coimbra, Portugal, 17-19 February 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 Journal, vol. 13, no. 10, pp. 3605–3615, October 2013.

[15] 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.

[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] 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.

[18] ——, "Temporal adaptive link quality prediction with online learning," ACM Transactions on Sensor Networks (TOSN), 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. 12 788–12 797, July 2017.

[20] V. Feng and S. Y. Chang, "Determination of wireless networks parameters through parallel hierarchical support vector machines," IEEE Transactions on Parallel and Distributed Systems, vol. 23, no. 3, pp. 505–512, March 2012.

[21] K. Bregar and M. Mohorčič, "Improving indoor localization using convolutional neural networks on computationally restricted devices," IEEE Access, vol. 6, pp. 17 429–17 441, 2018.

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

[23] A. Caciularu and D. Burshtein, "Blind channel equalization using variational autoencoders," in 2018 IEEE International Conference on Communications Workshops (ICC Workshops), 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 Transactions on Signal Processing, vol. 58, no. 3, pp. 1183–1192, March 2010.

[25] T. J. O'Shea, T. Roy, and T. C. Clancy, "Over-the-air deep learning based radio signal classification," IEEE Journal of Selected Topics in Signal Processing, 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 Journal of Selected Topics in Signal Processing, 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 Journal on Selected Areas in Communications, 2019.

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

[29] H. Ye, G. Y. Li, and B. Juang, "Power of Deep Learning for Channel Estimation and Signal Detection in OFDM Systems," IEEE Wireless Communications Letters, vol. 7, no. 1, pp. 114–117, Feb 2018.

[30] D. Neumann, T. Wiese, and W. Utschick, "Learning the MMSE Channel Estimator," IEEE Transactions on Signal Processing, vol. 66, no. 11, pp. 2905–2917, June 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 Transactions on Signal Processing, 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 Transactions on Antennas and Propagation, vol. 61, no. 12, pp. 6157–6167, Dec 2013.

[33] R. He, Q. Li, B. Ai, Y. L. Geng, A. F. Molisch, V. Kristem, Z. Zhong, and J. Yu, "A kernel-power-density-based algorithm for channel multipath components clustering," IEEE Transactions on Wireless Communications, 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 of Things Journal, 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 Transactions on Parallel and Distributed Systems, 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 of Things Journal, 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 Transactions on Vehicular Technology, 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 Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems, Toronto, Canada, November 2012, pp. 253–266.

[39] 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.

[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. 21 572–21 583, 2019.

[41] Y. Wang, Y. Xiang, J. Zhang, W. Zhou, G. Wei, and L. T. Yang, "Internet traffic classification using constrained clustering," IEEE Transactions on Parallel and Distributed Systems, 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 Transactions on Networking, 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 2017 Network Traffic Measurement and Analysis Conference (TMA), June 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 Transactions on Systems, Man, and Cybernetics - Part A: Systems and 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 Transactions on Mobile Computing, vol. 15, no. 6, pp. 1443–1456, June 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 2012 11th International Conference on Machine Learning and Applications, vol. 1, 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 Network, vol. 32, no. 1, pp. 96–101, Jan 2018.

[48] A. Azarfar, J. Frigon, and B. Sanso, "Improving the reliability of wireless networks using cognitive radios," IEEE Communications Surveys Tutorials, vol. 14, no. 2, pp. 338–354, Second 2012.

[49] Q. Dong and W. Dargie, "A Survey on Mobility and Mobility-Aware MAC Protocols in Wireless Sensor Networks," IEEE Communications Surveys Tutorials, vol. 15, no. 1, pp. 88–100, First 2013.

[50] H. Shi, R. V. Prasad, E. Onur, and I. G. M. M. Niemegeers, "Fairness in wireless networks:issues, measures and challenges," IEEE Communications Surveys Tutorials, vol. 16, no. 1, pp. 5–24, First 2014.

[51] L. Gupta, R. Jain, and G. Vaszkun, "Survey of Important Issues in UAV Communication Networks," IEEE Communications Surveys Tutorials, vol. 18, no. 2, pp. 1123–1152, Second quarter 2016.

[52] S. Jiang, "On reliable data transfer in underwater acoustic networks: A survey from networking perspective," IEEE Communications Surveys Tutorials, vol. 20, no. 2, pp. 1036–1055, Second quarter 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 Communications Surveys Tutorials, vol. 20, no. 1, pp. 708–769, First quarter 2018.

[54] Q. Mao, F. Hu, and Q. Hao, "Deep learning for intelligent wireless networks: A comprehensive survey," IEEE Communications Surveys Tutorials, vol. 20, no. 4, pp. 2595–2621, Fourth quarter 2018.

[55] C. Zhang, P. Patras, and H. Haddadi, "Deep learning in mobile and wireless networking: A survey," IEEE Communications Surveys Tutorials, vol. 21, no. 3, pp. 2224–2287, third quarter 2019.

[56] M. Amjad, L. Musavian, and M. H. Rehmani, "Effective capacity in wireless networks: A comprehensive survey," IEEE Communications Surveys Tutorials, 2019.

[57] S. K. Sharma and X. Wang, "Towards Massive Machine Type Communications in Ultra-Dense Cellular IoT Networks: Current Issues and Machine Learning-Assisted Solutions," IEEE Communications Surveys Tutorials, 2019.

[58] C. Fortuna, E. De Poorter, P. Škraba, and I. Moerman, "Data driven wireless network design: a multi-level modeling approach," Wireless Personal Communications, 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 International Conference on Communication Systems and Networks. Springer, 2017, pp. 110–126.

[60] M. Z. Zheleva, R. Chandra, A. Chowdhery, P. Garnett, A. Gupta, A. Kapoor, and M. Valerio, "Enabling a nationwide radio frequency inventory using the spectrum observatory," IEEE Transactions on Mobile Computing, vol. 17, no. 2, pp. 362–375, 2018.

[61] S. Rajendran, R. Calvo-Palomino, M. Fuchs, B. Van den Bergh, H. Cordobés, D. Giustiniano, S. Pollin, and V. Lenders, "Electrosense: Open and big spectrum data," IEEE Communications Magazine, vol. 56, no. 1, pp. 210–217, 2018.

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

[63] 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.

[64] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann, 2016.

[65] Y. Gil, E. Deelman, M. Ellisman, T. Fahringer, G. Fox, D. Gannon, C. Goble, M. Livny, L. Moreau, and J. Myers, "Examining the challenges of scientific workflows," Computer, vol. 40, no. 12, pp. 24–32, 2007.

[66] J. J. Van Bavel, P. Mende-Siedlecki, W. J. Brady, and D. A. Reinero, "Contextual sensitivity in scientific reproducibility," Proceedings of the National Academy of Sciences, vol. 113, no. 23, pp. 6454–6459, 2016.

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

[68] 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.

[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," Comp. Comm., 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 2017 IEEE 85th Vehicular Technology Conference (VTC Spring). IEEE, 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 Communications Magazine, vol. 37, no. 12, pp. 82–88, 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," Computer Networks, vol. 132, pp. 68–80, February 2018.

[73] P. Levis, S. Madden, J. Polastre, R. Szewczyk, K. Whitehouse, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer et al., "TinyOS: An operating system for sensor networks," in Ambient intelligence. 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," Chinese Journal of Electronics, vol. 26, no. 2, pp. 377–384, April 2017.

[75] 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.

[76] 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.

[77] D. S. De Couto, D. Aguayo, J. Bicket, and R. Morris, "A high-throughput path metric for multi-hop wireless routing," Wireless networks, 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 Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing. ACM, 2005, pp. 414–425.

[79] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in Proceedings of COMPSTAT'2010. Springer, 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 University 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 Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems. ACM, 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," Information Technology Journal, vol. 12, no. 8, pp. 1626–1631, April 2013.

[83] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis, "Collection tree protocol," in Proceedings of the 7th ACM conference on embedded networked sensor systems. ACM, 2009, pp. 1–14.

[84] R. Pedersen and M. Schoeberl, "An embedded support vector machine," in 2006 International Workshop on Intelligent Solutions in Embedded Systems. IEEE, 2006, pp. 1–11.

[85] B. Guo, S. R. Gunn, R. I. Damper, and J. D. Nelson, "Customizing kernel functions for svm-based hyperspectral image classification," IEEE Transactions on Image Processing, vol. 17, no. 4, 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," Computers & Electrical Engineering, vol. 61, pp. 297–311, 2017.

[87] S. J. Pan and Q. Yang, "A survey on transfer learning," IEEE Transactions on knowledge and data engineering, vol. 22, no. 10, pp. 1345–1359, 2009.

[88] A. Banerjee and S. Basu, "Topic models over text streams: A study of batch and online unsupervised learning," in Proceedings of the 2007 SIAM International Conference on Data Mining. SIAM, 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 Networks, vol. 36, pp. 127–151, 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," AI Review, 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 Newsletter, vol. 6, no. 1, pp. 1–6, June 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 Proceedings of the First International Conference on Advanced Data and Information Engineering (DaEng'13) - Lecture Notes in Electrical Engineering, December 2014, pp. 13–22.

[94] 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, 2004, pp. 121–132.

[95] D. Gokhale, S. Sen, K. Chebrolu, and B. Raman, "On the feasibility of the link abstraction in (rural) mesh networks," in INFOCOM 2008. The 27th Conference on Computer Communications. IEEE. IEEE, 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 TRIDENTCOM, Barcelona, Spain, 1-3 March 2006.

[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 2015 IEEE 35th International Conference on Distributed Computing Systems (ICDCS). IEEE, 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 International Symposium on Privacy Enhancing Technologies Symposium. Springer, 2009, pp. 108–127.

[99] 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.

[100] 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.

[101] 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.

[102] E. Anderson, C. Phillips, D. Sicker, and D. Grunwald, "Modeling environmental effects on directionality in wireless networks," Mathematical and Computer Modelling, vol. 53, no. 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," Knowledge Discovery from Sensor Data, 2007.

[104] F. Fenicia, L. Pfister, D. Kavetski, P. Matgen, J.-F. Iffly, L. Hoffmann, and R. Uijlenhoet, "Microwave links for rainfall estimation in an urban environment: Insights from an experimental setup in Luxembourg-City," Journal of Hydrology, vol. 464–465, pp. 69–78, September 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0022169412005483

[105] A. Kelmendi, G. Kandus, A. Hrovat, C. I. Kourogiorgas, A. D. Panagopoulos, M. Schoenhuber, M. Mohorčič, and A. Vilhar, "Rain attenuation prediction model based on hyperbolic cosecant copula for multiple site diversity systems in satellite communications," IEEE Transactions on Antennas and Propagation, vol. 65, no. 9, pp. 4768–4779, 2017.

[106] M. Spuhler, V. Lenders, and D. Giustiniano, "BLITZ: wireless link quality estimation in the dark," in Wireless Sensor Networks, Berlin, Heidelberg, 13-15 February 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 2013 Humaine association conference on affective computing and intelligent interaction. IEEE, 2013, pp. 245–251.

[108] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in Advances in neural information processing systems, 2014, pp. 2672–2680.

[109] H. Ye, G. Y. Li, B. F. Juang, and K. Sivanesan, "Channel agnostic end-to-end learning based communication systems with conditional gan," in 2018 IEEE Globecom Workshops (GC Wkshps), Dec 2018, pp. 1–5.

[110] Y. Yang, Y. Li, W. Zhang, F. Qin, P. Zhu, and C. Wang, "Generative-adversarial-network-based wireless channel modeling: Challenges and opportunities," IEEE Communications Magazine, vol. 57, no. 3, pp. 22–27, March 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. 69 464–69 480, 2019.

[112] T. Gruber, S. Cammerer, J. Hoydis, and S. Ten Brink, "On deep learning-based channel decoding," in 2017 51st Annual Conference on Information Sciences and Systems (CISS). IEEE, 2017, pp. 1–6.

[113] F. Hu and S. Kumar, "Deep Q -Learning-Based Node Positioning for Throughput-Optimal Communications in Dynamic UAV Swarm Network," IEEE Transactions on Cognitive Communications and Networking, 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 2017 IEEE International Conference on Communications (ICC), May 2017, pp. 1–6.

Gregor Cerar received his 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 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 with the Next Generation Wireless Research Group, University of Southampton in 2015. He is an Assistant Professor with the Department of Electrical and Electronics Engineering, Bitlis Eren University, Turkey and a research fellow at the Department of Communication Systems of 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 TPC member for IEEE VTC-2018, VTC-2019, VTC-2020 and IEEE Globecom-2020, and is an associate editor with IEEE Access. He was a recipient of the full scholarship granted by the Republic of Turkey, Ministry of National Education.

Mihael Mohorčič 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 authored or co-authored over 200 refereed journal and conference papers, co-authored 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. He participated in many EC co-funded research projects since 1996 and is currently involved in H2020 projects Fed4FIRE+, SAAM, RESILOC and BD4OPEM.

Carolina Fortuna received her B.Sc. in 2006, Ph.D. in 2013 and was a postdoctoral research associate at IBCN, Ghent University, 2014-2015 and visited Stanford Infolab in 2017. Currently, she is a research fellow at the Department of Communication Systems, Jožef Stefan Institute and an assistant at the Jožef Stefan International Postgraduate School. Her research is interdisciplinary, focusing on data and knowledge driven modelling of communication and sensor systems. She has participated in H2020, FP7 and FP6 projects including in various leadership roles. She coauthored over 50 peer-reviewed publications, edited a book, was a TPC member at IEEE ICC 2011-2021, ESWC 2012, IEEE Globecom 2011 - 2021, VTC 2010, 2016, IEEE WCNC 2009.