

UNIVERSITY OF LJUBLJANA
FACULTY OF COMPUTER AND INFORMATION SCIENCE

Timotej Gale

**Link Quality Prediction in Wireless
Networks**

UNDERGRADUATE THESIS

PROFESSIONAL STUDY PROGRAMME

FIRST CYCLE

COMPUTER AND INFORMATION SCIENCE

MENTOR: doc. dr. Tomaž Curk

CO-MENTOR: dr. Carolina Fortuna

Ljubljana, 2017

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Timotej Gale

**Napovedovanje kakovosti povezav v
brezžičnih omrežjih**

DIPLOMSKO DELO

VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM
PRVE STOPNJE
RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: doc. dr. Tomaž Curk
SOMENTOR: dr. Carolina Fortuna

Ljubljana, 2017

This work is licensed under the Creative Commons Attribution-ShareAlike 4.0 International License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/4.0/>.



This document was typeset using L^AT_EX.

Faculty of Computer and Information Science issues the following dissertation:

The topic of the thesis:

Predicting wireless link quality may alleviate many problems associated with a crowded wireless spectrum and improve overall performance. Explore available data sets on link quality and propose novel features for modeling link quality. Propose a model for link quality prediction and report on its predictive performance.

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

Uspešno napovedovanje kakovosti brezžičnih povezav lahko omili vrsto težav, povezanih z zasičenim radijskim spektrom, ki ga naprave uporabljajo za brezžično komunikacijo. Na podlagi začetne analize obstoječih zbirk podatkov o kakovosti povezav v brezžičnih omrežjih predlagajte nove značilke za napovedovanje kakovosti brezžičnih povezav. Predlagajte postopek izgradnje napovednega modela in ovrednotite klasifikacijsko točnost napovednih modelov.

Iskreno se zahvaljujem mentorjema dr. Carolini Fortuna in doc. dr. Tomažu Curku za vse nasvete in pomoč pri izdelavi diplomskega dela. Prav tako se zahvaljujem vsem, ki so na kakršenkoli način pripomogli k uspešnemu zaključku tega diplomskega dela in me podpirali v času izobraževanja.

Contents

Abstract

Povzetek

Razširjeni povzetek

1	Introduction	1
1.1	Properties of the radio spectrum and wireless networks	2
1.2	Data mining methods and tools	6
1.3	Link quality prediction problem formulation	12
1.4	Related work	13
2	Data preprocessing and exploration	17
2.1	The Rutgers dataset	17
2.2	The Sigfox dataset	23
3	Development of the predictor for link quality	29
3.1	Algorithm selection	29
3.2	Feature engineering	31
3.3	Predictor development workflow	37
4	Results	39
4.1	Wi-Fi link quality prediction	39
4.2	Sigfox link quality prediction	40
4.3	Discussion	42

5 Conclusions

45

Bibliography

47

List of acronyms and abbreviations

	English	Slovenian
CA	classification accuracy	klasifikacijska točnost
CSI	channel state information	informacija o stanju kanala
CSMA/CA	carrier-sense multiple access with collision avoidance	večkratni dostop s prepoznavanjem nosilca in izogibanjem trkom
DSSS	direct-sequence spread spectrum	neposredno razširjanje s kodnim zaporedjem
ETX	expected transmission count	pričakovano število pošiljanj
FHSS	frequency-hopping spread spectrum	razširjanje s frekvenčnim skakanjem
F-LQE	fuzzy link quality estimator	mehka cenilka kakovosti povezave
GAN	global area network	globalno omrežje
GUI	graphical user interface	grafični uporabniški vmesnik
IEEE	Institute of electrical and electronics engineers	Inštitut inženirjev elektrotehnike in elektronike
ITU	International telecommunication union	Mednarodna telekomunikacijska zveza

JSI	Jožef Stefan institute	Institut "Jožef Stefan"
KLE	Kalman-filter-based link quality estimator	cenilka povezave na osnovi Kalmanovega filtra
KNIME	Konstanz information miner	informacijski rudar Konstanz
L-ETX	learn on the fly - expected number of physical transmissions	učenje "v naglici" - pričakovano število pošiljanj
LI	link inefficiency	neučinkovitost povezave
L-NT	learn on the fly - number of physical transmissions	učenje "v naglici" - število fizičnih prenosov
LP-WAN	low-power wide area network	javno omrežje z nizko močjo
LQE	link quality estimator	cenilka kakovosti povezave
LQI	link quality indicator	indikator kakovosti povezave
MAC	media access control	nadzor dostopa do medija
MAN	metropolitan area network	mestno omrežje
NIC	network interface controller	omrežna kartica
OFDM	orthogonal frequency division multiplexing	ortogonalno frekvenčno multipleksiranje
ORBIT	open access research testbed for next-generation wireless networks	odprtodostopno raziskovalno testno okolje za brezžična omrežja naslednje generacije
PA	programmable attenuator	programabilni slabilni člen
PHY	physical layer	fizična plast
PRR	packet reception ratio	razmerje prejetih paketov
RNP	required number of packet retransmissions	potrebno število ponovitev paketov
RSSI	received signal strength indicator	indikator moči sprejetega signala

S.D.	standard deviation	standardni odklon
SNR	signal-to-noise ratio	razmerje signal/šum
USRP	universal software radio peripheral	univerzalna programska radijska periferija
WAN	wide area network	javno omrežje
WEKA	Waikato environment for knowledge analysis	okolje za analizo znanja Waikato
WLAN	wireless local area network	brezžično lokalno omrežje
WMEWMA	window mean with exponentially weighted moving average	povprečje okna z eksponentno uteženim drsečim povprečjem
WPAN	wireless personal area network	brezžično osebno omrežje
WRE	weighted regression estimator	utežena regresijska cenilka
WSN	wireless sensor network	brezžično senzorsko omrežje

Abstract

Title: Link Quality Prediction in Wireless Networks

Author: Timotej Gale

The number of wireless devices is increasing rapidly. The wireless spectrum is thus becoming crowded as various technologies co-exist and interfere with each other. One possible way to improve the performance of existing technologies is to develop accurate link quality estimators. In this thesis, we propose, implement and evaluate a novel approach to link quality prediction based on feature engineering. Following a preliminary analysis of a dataset with Wi-Fi packet traces and a dataset with Sigfox packet traces, we developed new features and built a classification model for link quality prediction. The proposed models vary in performance with respect to accuracy and completeness of predicting different types of links, mainly links of intermediate quality. The best proposed model achieved 95% classification accuracy, which is a substantial improvement compared to the 60% accuracy of the majority classifier.

Keywords: data mining, link quality, wireless network, modeling, prediction, estimator.

Povzetek

Naslov: Napovedovanje kakovosti povezav v brezžičnih omrežjih

Avtor: Timotej Gale

Število brezžičnih naprav danes hitro narašča, posledično se viša stopnja nasičenosti radijskega spektra, saj soobstoj številnih tehnologij povzroča motnje v omrežjih. Zmogljivost obstoječih tehnologij lahko povečamo z razvojem natančnejših cenilk kakovosti povezav. V diplomskem delu predlagamo, uvedemo in ovrednotimo nov pristop za razvoj sistema za napovedovanje kakovosti povezav, ki temelji na gradnji in izpeljavi značilk. Po predhodni analizi množic podatkov o paketih Wi-Fi in Sigfox tvorimo nove značilke in izgradimo klasifikacijski model za napovedovanje kakovosti povezave. Predlagani modeli se razlikujejo glede na točnost in popolnost napovedovanja posameznih vrst povezav, predvsem povezav srednje kvalitete. Najboljši model pravilno uvrsti 95% testnih primerov, kar je bistveno izboljšanje v primerjavi s 60% točnostjo večinskega klasifikatorja.

Ključne besede: podatkovno rudarjenje, kakovost povezave, brezžično omrežje, modeliranje, napovedovanje, cenilka.

Razširjeni povzetek

Zaradi razmaha omrežnih tehnologij, zmanjševanja velikosti elektronskih vezij in nižanja cen strojne opreme, lahko na trgu zasledimo vedno večjo nasičenost z brezžičnimi napravami, ki so povezane v omrežje. Večino teh naprav predstavljajo senzorji in naprave z omejenimi viri, zmogljivost omrežij s takšnimi napravami pa je v veliki meri pogojena z natančnimi ocenami kakovosti posameznih povezav med napravami.

V diplomskem delu predlagamo, uvedemo in ovrednotimo nov pristop za razvoj sistema za napovedovanje kakovosti povezav, ki temelji na inženiringu značilk. Po predhodni analizi podatkovnih množic s podatki o paketih Wi-Fi in Sigfox razvijemo nove značilke in izgradimo klasifikacijski model za napovedovanje kakovosti povezav. Model tudi ovrednotimo.

Po predstavitvi brezžičnih omrežij in podatkovnega rudarjenja ter povezanih metod in orodij formuliramo problem napovedovanja kakovosti povezave in predstavimo sorodno delo.

V diplomskem delu analiziramo dve podatkovni množici, ki ju uporabimo v razvoju sistema za napovedovanje kakovosti povezave. Prva podatkovna množica (Rutgers) je bila zajeta v testnem okolju “*Open Access Research Testbed for Next-Generation Wireless Networks (ORBIT)*” in vsebuje podatke o poslanih paketih Wi-Fi. Podatki so bili zajeti tekom petih eksperimentov, za vsak eksperiment so raziskovalci določili nivo motenj v omrežju med 0 dBm in -20 dBm, v korakih po 5 dBm.

Po čiščenju podatkov le-te statistično povzamemo s petimi števili (ang. *five number summary*). Razmerje prejetih paketov (ang. *packet reception*

ratio, PRR) se pri izbranih povezavah giba med 22% (slaba povezava) in 94.66% (zelo dobra povezava). Ugotovimo, da ima povezava z najvišjo povprečno vrednostjo RSSI (indikator moči sprejetega signala, ang. *received signal strength indicator*) in najmanjšim standardnim odklonom RSSI najvišjo vrednost PRR. Z drugimi besedami, stabilne povezave z visoko povprečno vrednostjo RSSI imajo visoko vrednost PRR, nestabilne povezave z visoko povprečno vrednostjo RSSI pa imajo nižjo vrednost PRR in se obnašajo kot prehodne povezave. Ugotovimo tudi, da so nekatere povezave med posameznimi napravami v naši podatkovni množici asimetrične. Pokažemo, da ima lahko majhna sprememba v moči motenj velik vpliv na PRR, povezava lahko postane tudi popolnoma neuporabna.

S pomočjo metod nenadzorovanega učenja preverimo, ali lahko v naši podatkovni množici zaznamo tri področja kakovosti povezave, o katerih poročajo v sorodnih člankih. Potrdimo predhodna opažanja.

Druga podatkovna množica (Sigfox) je bila zajeta tekom eksperimentov na Institutu "Jožef Stefan" (IJS). Množica vsebuje meritve RSSI in SNR (razmerje signal/šum, ang. *signal-to-noise ratio*) poslanih paketov Sigfox. Tekom eksperimentov se je bazna postaja nahajala na strehi IJS, oddajnik pa je poslal 100 paketov na štirih lokacijah, za štiri različne jakosti pošiljanja. Protokol Sigfox je bil prilagojen tako, da je pri vsakem paketu bila poslana le prva repeticija. Po čiščenju podatkov le-te statistično opišemo s povzetkom s petimi števili. PRR se giba med 61% (slaba povezava) in 96% (dobra povezava). Ugotovimo, da so relacije med RSSI in PRR enake kot pri prvi podatkovni množici (Rutgers). Stabilne povezave z visokim RSSI imajo visok PRR, nestabilne povezave z visokim RSSI pa imajo nižji PRR.

Raziščemo povezavo med jakostjo pošiljanja in RSSI ter SNR. Opazimo, da so vrednosti RSSI in SNR višje pri višji moči pošiljanja. Pokažemo tudi, da sta RSSI in PRR korelirana.

Osrednji del diplomskega dela predstavlja razvoj sistema za napovedovanje kakovosti povezav v brezžičnih omrežjih. Naš cilj je izgradnja modela, ki napove kakovost posamezne povezave v poljubni točki v času glede na

trenutno meritev in združene meritve preteklih podatkov. Z drugimi besedami, na osnovi trenutnih vrednosti fizične plasti in kombinaciji preteklih vrednosti fizične plasti želimo napovedati PRR, ki nastopi takoj po prejemu zadnjega paketa (meritve). Pri združevanju preteklih meritev uporabimo različna okna. Okno določa število preteklih meritev, ki jih združimo.

Za modeliranje uporabimo klasifikacijsko drevo, algoritem J48. Algoritem uporablja entropijo in informacijsko teoretično metriko (ang. *information theoretic metric*) za izbor atributov, ki najbolj razlikujejo posamezne primere v podatkovni množici, na podlagi tega pa izgradi klasifikacijsko drevo. Poleg izvajanja klasifikacije je ta algoritem uporaben tudi za lažje razumevanje podatkovne množice in pomembnosti atributov. Algoritmu določimo nekatere parametre. Najpomembnejši je interval zaupanja rezanja (ang. *pruning confidence factor*) drevesa. Z ustrezno nastavitvijo preprečimo prekomerno prileganje (ang. *overfitting*) modela podatkom.

V naslednjem koraku tvorimo značilke za obe podatkovni množici. Pri prvi podatkovni množici lahko za tvorjenje značilk uporabimo RSSI in PRR, pri drugi pa RSSI, SNR in PRR. PRR izračunamo iz sekvenčnih števil paketov. Tvorimo vse možne kombinacije značilk in sestavimo ustrezne učne množice. Primerkom v učni in testni množici dodamo še ciljne razrede, te določimo glede na vrednost PRR. Vrednost PRR večja od 90% zajema ciljni razred *good*, PRR med 10% in 90% predstavlja razred *intermediate*, pri PRR, ki je manjši od 10%, pa določimo ciljni razred *bad*.

Ker so primerki v ciljne razrede razvrščeni izjemno neuravnoteženo (model je v takšnih primerih pristranski), izvedemo še nekaj dodatnih operacij nad podatki, s katerimi dobimo več različic učnih množic. Pri prvi različici uporabimo standarden postopek, ki je vgrajen v program WEKA. Naključno vzorčimo iz posameznih ciljnih razredov z vračanjem, na ta način dobimo učno množico, ki je enake velikosti kot začetna in ima enakomerno porazdelitev razredov. Pri drugi različici dodamo vrednosti za (manjkajoče) pakete v slabih in prehodnih povezavah, to storimo z interpolacijo začetnih podatkov. Tretjo učno množico pridobimo z naključnim vzorčenjem brez vračanja, ki

ga izvedemo nad drugo različico.

Sistem za napovedovanje kakovosti povezav sestavljajo skripte in programi v programskih jezikih Python ter Java. Podatke najprej očistimo in pretvorimo v enotno obliko z uporabo skripte v programskem jeziku Python. Z naslednjo skripto tvorimo značilke, ki jih nato uvozimo v program za gradnjo modelov. Le-ta je napisan v programskem jeziku Java in vključuje program WEKA.

Razvite modele ovrednotimo s prečnim preverjanjem in predstavimo kompromise med modeli. Največ informacij za razlikovanje med primeri v podatkovni množici je prispevala povprečna vrednost RSSI, najbolj točen model pa je uporabljal vse značilke. Najvišji odstotek pravilno razvrščenih primerkov dobimo z uporabo interpolirane učne množice. Model pravilno uvrsti 95% testnih primerov, kar je bistveno izboljšanje v primerjavi s 60% točnostjo večinskega klasifikatorja. Predlagani modeli se razlikujejo glede na preciznost (ang. *precision*) in senzitivnost (ang. *sensitivity*) napovedovanja posameznih vrst povezav, predvsem povezav srednje kvalitete.

Vključitev razvitih modelov v brezžična omrežja, še posebej v usmerjevalne protokole, bi lahko znatno izboljšala učinkovitost teh omrežij. Predstavimo še predloge za nadaljnje delo.

Chapter 1

Introduction

With the upswing of network technologies, minimization of circuits and decrease in hardware prices there is an increase of interconnected devices, predominantly wireless devices with limited resources, *e.g.*, wireless sensors. Some studies forecast that global market for wireless sensor devices will increase from \$2.4 billion to \$7.7 billion in 2016–2021 [28] and the number of connected devices will reach 20.4 billion - three devices per capita - by 2020 [32].

The wireless spectrum used by these devices is a scarce resource. Therefore, more efficient technology has to be developed to enable good connectivity for such large number of devices. Wireless radio signals are influenced by many factors such as obstacles and other devices in the vicinity. Because wireless sensor devices operate at a very low power, they are more susceptible to interference. Accurately estimating the wireless link quality is essential in low power networks and has a significant influence on network performance, *e.g.*, network throughput. [4]

In this thesis, we propose, implement and evaluate a novel approach to developing a link quality predictor based on feature engineering. We set out to analyze two available datasets on which we perform feature engineering and modeling. Next, we train the classifier and perform the evaluation and discuss trade-offs. Some of the results of our study were published in a peer-

reviewed paper [31].

This thesis is structured as follows. This chapter introduces the minimal background from wireless networks and data mining that is required to understand the content of the thesis. It then formulates the problem and summarizes the related work. Chapter 2 provides an analysis of the datasets used for the link quality prediction task and outlines the data preprocessing procedure. The process of development of a predictor for link quality is described in Chapter 3. Results are discussed in Chapter 4 and Chapter 5 concludes the thesis.

1.1 Properties of the radio spectrum and wireless networks

Wireless networks are computer networks in which data are transmitted over the air, therefore devices do not need to be physically connected via cables to a network. Such networks consequently enable user mobility in addition to data connectivity [26]. Devices or any arbitrary systems, which have a network address and are connected to a network, are called nodes [8]. When a node has limited resources, such as power, memory and processing capabilities, it is defined as a constrained node [7].

Individual wireless networks differ in many aspects. Depending on wireless range, performance, technology (Wi-Fi, Bluetooth, ...) and a number of devices, wireless networks can be categorized as [22]:

- **wireless personal area networks (WPAN)**, which connect devices in a very short range, usually within a person's workspace,
- **wireless local area networks (WLAN)**, which connect at least two devices in a short range, they may also provide access to the internet,
- **wireless metropolitan area networks (MAN)**, which connect multiple WLANs and cover entire cities,

- **wireless wide area networks (WAN)**, which expand over large geographical areas (multiple cities, regions),
- **low-power wireless wide area networks (LPWAN)**, which are WANs that are designed to enable low-power, long range and low data rate communication between constrained devices [17],
- **global area networks (GAN)**, which expand over an unlimited area and an indefinite number of networks,
- **space networks**, which enable transmissions between spaceships in earth's proximity.

In this thesis, the focus is on Wi-Fi, the most popular unlicensed WLAN technology and on Sigfox, an emerging unlicensed LPWAN technology. We describe Wi-Fi and Sigfox based on [30] and [31], respectively:

Wi-Fi is a short-range high-speed spread spectrum wireless technology, operating in unlicensed 2.4 GHz and 5 GHz bands, mainly used in local area networks. Wi-Fi is based on the IEEE 802.11 standards and thus uses the carrier-sense multiple access with collision avoidance (CSMA/CA) medium access method. Different amendments to the 802.11 standard (802.11b, 802.11g, ...), also called Wi-Fi types, provide miscellaneous support for maximum data transmission rate and signal range. They also employ various spread spectrum methods, namely FHSS, DSSS, and OFDM. Wi-Fi networks can have two modes of operation (see Figure 1.1): infrastructure mode and *ad hoc* mode, depending on whether the network has an access point (infrastructure) or the devices communicate directly (*ad hoc*). In order to ensure interoperability, Wi-Fi devices are tested and certified by an organization called Wi-Fi Alliance. [30]

Sigfox is an ultra-narrowband low power technology used in low-power wide area networks. While it only supports low data rate, it is energy efficient and enables long range communication. The Sigfox network consists of

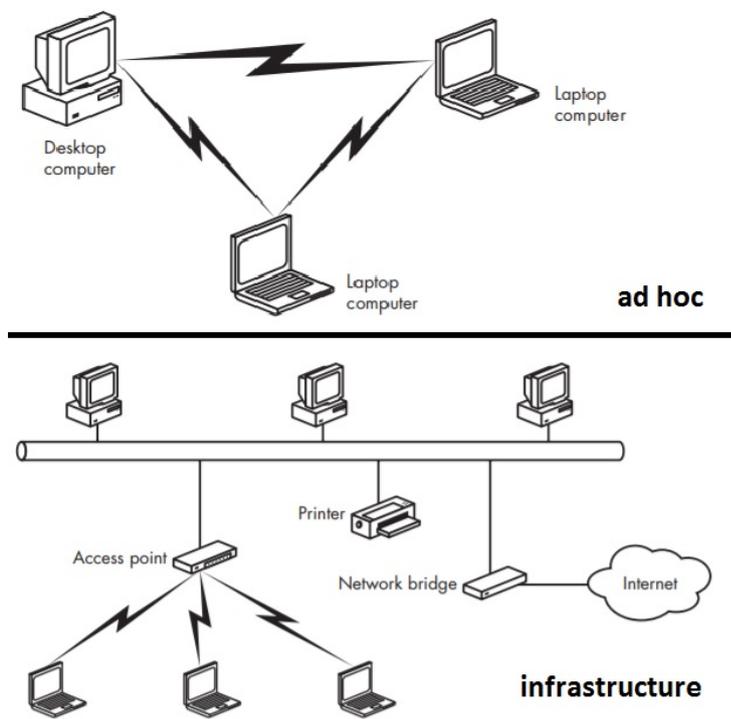


Figure 1.1: Wi-Fi modes of operation [30].



Figure 1.2: Sigfox network architecture [25].

wireless devices and base stations. It is based on a proprietary Sigfox protocol, which implements frequency hopping and frame repetition sequences. Every frame sent from a device is repeated on three out of 1500 total channels, which reside in the unlicensed 868 MHz band. Channel selection is random. Devices may only send to or receive data from a base station, device to device communication is not supported. [31]

All received data is forwarded to the Sigfox backend (see Figure 1.2), where users can see the raw data, networks statistics or forward the data to a custom web service.

1.1.1 Radio spectrum

Wireless networks utilize radio spectrum as their transmission medium, which represents a segment of the electromagnetic spectrum. Electromagnetic waves in the 3 kHz - 300 GHz frequency range are also called radio waves. Frequencies are partitioned into small sections, named frequency bands. Each band, based on its properties, has a different application. [26]

Communication at lower frequencies can reach longer distances than communication at higher frequencies, analogously to electromagnetic spectrum characteristics. Since data rate is closely related to frequency, communication at lower frequencies comes at the cost of having lower data rates than

communication at higher frequencies.

The usage of radio spectrum is regulated by laws and coordinated by International Telecommunication Union (ITU) [1]. Radio spectrum, depending on whether it is allocated to be reserved by organizations, may either be licensed or unlicensed. Licensed radio spectrum essentially experiences no outside interference and maintains a high signal-to-noise ratio, however it is expensive. Unlicensed spectrum, on the other hand, is free, although devices using the unlicensed spectrum must comply with requirements and regulations, such as maximum transmission power. [29]

1.1.2 Link quality estimation and prediction

A wireless link is a logical connection, formed between two devices, used to convey data packets. Link quality can be modeled by a concept called link abstraction. Using this model, it can be determined whether the link is likely to be stable and predictable at a certain time and place. In [13], authors suggest that error rates due to multipath propagation of radio waves are not unpredictable, meaning that it is possible to abstract wireless links and predict the link quality. Wi-Fi and Sigfox link abstractions, however, differ due to inherent characteristics of the two technologies. While Wi-Fi uses a predetermined static wireless channel for all packet transmissions, Sigfox is performing dynamic channel hopping at every transmission. This problem is depicted in Figure 1.3 [12]. Due to spatial differences in the radio spectrum, the link abstraction, while valid for Wi-Fi, is questionable for Sigfox.

1.2 Data mining methods and tools

According to Encyclopaedia Britannica, data mining is [9]: “(...) in computer science, the process of discovering interesting and useful patterns and relationships in large volumes of data. The field combines tools from statistics and artificial intelligence (such as neural networks and machine learning) with database management to analyze large digital collections, known as data

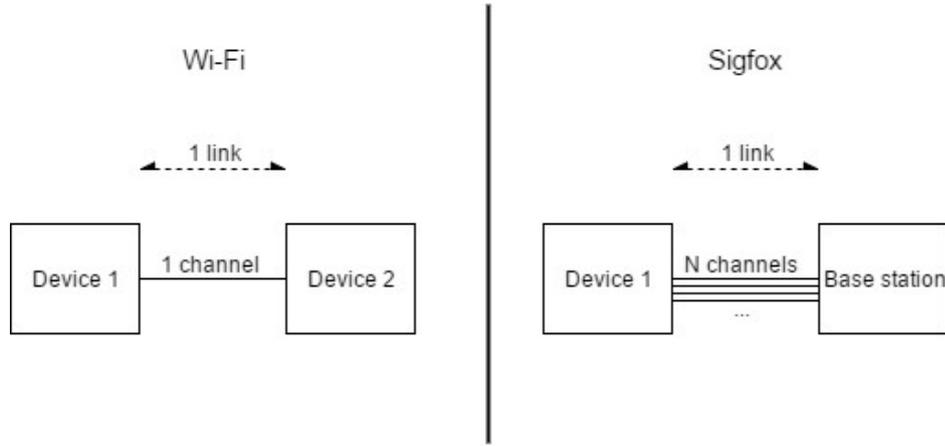


Figure 1.3: Difference in link abstractions between Sigfox and Wi-Fi [12].

sets.” In this thesis, we use the knowledge discovery process to extract useful knowledge from wireless network data [21] and develop a link quality classifier.

The main component of knowledge extraction process are algorithms called learning algorithms, which are used to learn from data. Learning, depending on the magnitude of knowledge that is provided to the learner, may be divided into two main categories: supervised and unsupervised [21]. Supervised learning refers to a class of machine learning tasks which exploit outcome variables, or labels, to direct the learning process [16]. Unsupervised learning, on the other hand, refers to a class of machine learning tasks which observe only input features and has no knowledge of outcome variables [16]. The goal of supervised learning is to predict the outcome variable based on input features while the goal of unsupervised learning is to infer the structure of input data without having any labels [16]. In a setting where both labeled and unlabeled data exist, a combination of supervised and unsupervised learning may be used. Such learning is called semi-supervised learning [15].

Based on how data is fed to the learner, algorithms can be classified as either online or offline. Offline algorithms are algorithms which receive entire

input prior to taking action [18]. In the context of machine learning, this can be thought of as a machine learning method in which the learner is trained only once on a certain amount of data. On the contrary, online algorithms are algorithms which receive a series of inputs and take action after every input [18]. In the context of machine learning, this can be thought of as a machine learning method in which the learner is iteratively trained on new data as it becomes available.

Learning algorithms attempt to solve various real-world problems by employing specific data mining tasks. In data mining, there exist several basic tasks:

1. **Classification** is a data analysis problem of deriving a model that describes data categories and annotating new data with said categories [15].
2. **Regression** is a task of modeling continuous-valued functions. The goal of regression is to predict numerical output variables. [15]
3. **Clustering** is used for partitioning data instances into clusters (groups) based on the similarity between attributes describing the said instances. Instances within a particular cluster have a high level of similarity, while instances within different clusters are highly dissimilar. [15]
4. **Outlier detection** is a problem of discovering data instances that deviate from normal, expected behavior (outliers or anomalies) [15].
5. **Association analysis** is used when the goal is to find interesting relationships in data. To put it differently, we want to know which values of attributes appear most frequently together in the data [16]. The relationships may be represented by association rules.
6. **Summarization** is a task of identifying a reduced (summarized) representation of data with the goal of optimizing compaction gain and information loss [21].

1.2.1 Methods and techniques

The process of developing a link quality predictor comprises of multiple steps, with each step coupling related activities for achieving a specific goal. At every step, we apply various methods and techniques.

Data acquisition. First of all, we acquire suitable data for developing a link quality predictor. We inspect several public datasets and select the most appropriate, based on our goals and requirements.

Data preprocessing. Next, we clean the data by discarding incorrect and anomalous values from the dataset, as well as any unnecessary data. By doing so, we may significantly improve the accuracy of the models and speed up the model training process. The final step of data preprocessing is data transformation. We couple the data that is split across multiple files and unify the data format across all datasets. This results in data, which is suitable for training (feature vectors) [21].

Data analysis. In this step, we explore the data. We statistically describe the data using five number summaries, which provide succinct information about the distribution of the data and help us better understand the data [15]. By plotting relationships between different features, such as PRR and RSSI, we can observe the correlation between various features and examine the feasibility of modeling such dependencies [21].

Feature generation. In this step, we enrich the feature vectors. By combining information of adjacent data points, we aggregate the observations and compute additional features. Additional features provide extra information to the learner and enhance the models.

Model training. In this step, we employ learning algorithms to build models using the enriched feature vectors. In order to find the best combination of input features, we build the models in bulk by utilizing all possible input vectors.

Evaluation. Finally, we assign scores to the trained models. We do so by applying cross-validation. Cross-validation is a technique for model evaluation based on data partitioning. The validation process may be described

as follows: data are first randomly split into a specified number of equal sized parts, models are then iteratively trained on all but one data part and tested on the remaining data part till all the data parts have been used for testing exactly once [15]. By using cross-validation, we can gain a better insight into how well a model would perform in practice.

1.2.2 Data mining tools

Data mining tools offer a convenient bundle of machine learning algorithm implementations and data processing tools for carrying out data mining tasks, thus simplifying and streamlining the process of knowledge extraction and link quality classifier development. In this section, we present and compare commonly used open source tools, namely: KNIME, Orange, RapidMiner, R Project, and WEKA.

Konstanz Information Miner (KNIME) is a modular data analysis environment with emphasis on workflow modeling. It is a teaching, research and collaboration platform that provides a graphical user interface for construction and interactive execution of data analysis processes. It may be integrated with other data analysis software and provides additional functionalities via extensions. [6]

Orange is an all-around software for machine learning and data mining. It includes machine learning algorithms, preprocessing tools and visualization capabilities. This tool is used in industry, science, and teaching and aims to make common machine learning and data mining techniques flexible and user-friendly. Orange may also be used as a Python library and extended with additional modules. [10]

RapidMiner is a cross-platform data science platform that aids all steps of machine learning procedures. It is written in Java and provides a graphical user interface for construction and execution of data analysis processes. RapidMiner is used in industry, science, and teaching. Additional functionalities may be added via plugins. [3]

R is an open source programming language as well as an environment

Table 1.1: Feature summary of data mining tools, an excerpt from [33].

	KNIME	Orange	RapidMiner	R	Weka
K-means clustering	Yes	Yes	Yes	Yes	Yes
Association rule mining	Yes	Yes	Yes	Yes	Yes
Linear regression	Yes	Yes	Yes	Yes	Yes
Logistic regression	Yes	Yes	Yes	Yes	Yes
Naive Bayesian classifier	Yes	Yes	Yes	Yes	Yes
Decision tree	Yes	Yes	Yes	Yes	Yes
Time series analysis	Yes	No	Some	Yes	Yes
Text analytics	Yes	Yes	Yes	Yes	Yes
Big data processing	No	No	No	Yes	Yes
Visual workflows	Yes	Yes	Yes	No	Yes

for data processing, calculations and producing graphics. It is used for statistical computing and provides many statistical and graphical techniques. Users may extend R's functionalities by writing new functions or installing packages. [2]

Waikato Environment for Knowledge Analysis (WEKA) is a modular open source workbench that provides an extensive collection of algorithms and tools for data preprocessing, machine learning and data mining. It is used in both academia and industry, and supports many machine learning tasks, including classification, regression, clustering, association rule mining, and attribute selection. The initial version of this software was written in C but was later completely rewritten in Java. WEKA may be used through one of the many graphical user interfaces (GUI) it provides or embedded in a Java application. [14]

While most of the tools offer similar functionalities, there are some minor differences. Table 1.1 shows a feature summary of all described tools. We can see that WEKA offers the most features. All tools support basic data mining techniques, such as classification and regression. R lacks support

for visual workflows, therefore it requires the knowledge of the programming language syntax in order to execute tasks. RapidMiner is simple to install, however many features from the commercial version are not present in the open source version. The installation process of KNIME is complicated and requires extensive technical knowledge. [33]

1.3 Link quality prediction problem formulation

Given a pair $\langle x_i, y_i \rangle$ of devices that communicate by sending packets over a bidirectional wireless link $L \subset \{L_{x_i}, L_{y_i}\}$, with L_{x_i} and L_{y_i} denoting unidirectional links imposed by transmissions from y_i to x_i and x_i to y_i respectively, we would like to find out the likelihood of successfully receiving the next packet

$$P(\text{reception}_{L(i+1)} \mid \text{data}_{L(i)}) \quad (1.1)$$

for every direction of the link L , based on the current packet (pk) and an aggregation of current and $W - 1$ previous packets:

$$\text{data}_{L(i)} = [pk_{L(i)}, \text{AGG}(pk_{L(i)}, pk_{L(i-1)}, \dots, pk_{L(i-(W-1)}))] . \quad (1.2)$$

To achieve this, we develop a model based on empirical input data. For every packet, we construct an input feature vector using physical layer information ($PLI_i \subseteq \{RSSI_i, SNR_i\}$) of the current packet and combined physical layer information of the current and $W - 1$ previous packets, as well as the packet reception ratio immediately succeeding the reception of the current packet:

$$F_i = [PLI_i, \text{AGG}(PLI_i, PLI_{i-1}, \dots, PLI_{i-(W-1)}), PRR_i] . \quad (1.3)$$

When combining the information about previous packets, we use different time windows. The time window specifies how many historical packets are taken into account, *e.g.*, for calculating average RSSI over time window 5,

we use the current packet measurement and 4 previous measurements. A generalization of an arbitrary aggregation function on any time window is as follows:

$$\text{AGGREGATE}(AGG, M, W, i) = AGG(M_i, M_{i-1}, \dots, M_{i-(W-1)}) , \quad (1.4)$$

where AGG is an aggregation function, W is a time window ($W > 0$), and M_i is a measurement with i specifying the measurement offset.

The system outputs the predicted packet reception ratio in the range $[0, 1]$ or a label associated with a defined scope of predicted PRR.

1.4 Related work

Radio link quality estimation is a long researched topic; in the last decade, it has been explored in detail by many researchers. The importance of link quality estimation mainly lies in the essential impact it has on wireless networks, primarily wireless networks with constrained devices. While link quality estimation influences network performance, it also impacts the architecture of higher-layer protocols, *e.g.*, routing protocols [4]. Accurately and efficiently estimating the link quality remains a challenge due to the unique radio link characteristics [4]:

1. **Spatial characteristics.** The correlation between link quality and the distance between nodes is generally non-existent. Link quality can be categorized into three regions, based on stability over time, link symmetry, and overall quality: connected region, transitional region, and disconnected region. These regions are not static nor isotropic, meaning that region boundaries change over time and are of irregular shape. The transitional region, including the links of intermediate quality (PRR¹ between 10% and 90%), is known to be unreliable and to some extent unpredictable. [4]

¹Packet reception ratio is defined as the number of successfully received packets divided by the number of all transmitted packets.

2. **Temporal characteristics.** Link quality fluctuates in relation to time. Stable links often have either very low or very high average PRR. On the contrary, unstable links mostly reside in the transitional region (intermediate average PRR). Variation in environment traits may sometimes cause momentary correlation in packet reception. [4]
3. **Link asymmetry.** Link asymmetry, mostly prominent in links of intermediate quality, defines the inequality in connectivity amongst the directions of a link. [4]
4. **Interference.** Interference occurs in wireless networks because radio spectrum is shared between numerous wireless devices, which may send data at the same time and interfere with each other's transmissions. Depending on whether interference happens inside one or between different networks/technologies, interference may either be internal or external. [4]

Existing link quality estimators are either hardware or software based. Hardware based estimators (RSSI, SNR, LQI) are straightforwardly available in radio chip's registers. While such estimators have no computational overhead, their capacity of holistically capturing the link quality is debatable [4]. Software based link quality estimators are split into three groups:

1. PRR based (PRR, WMEWMA, KLE): A group of estimators that take into account the number of successfully received packets. PRR is also often used as an evaluation metric for hardware based estimators. [4]
2. RNP based (RNP, LI, ETX, Four-bit, L-NT, L-ETX): A group of estimators that are based on an average number of needed packet transmissions before successfully receiving a packet. [4]
3. Score based (WRE, MetricMap, F-LQE, CSI): A group of estimators that: "(...) provide a link estimate that does not refer to a physical phenomenon (like packet reception or packet retransmission); rather, they provide a score or a label that is defined within a certain range" [4].

One of the main challenges remains the evaluation of link quality estimators. To this day, there is no objective baseline link quality metric [4]. Only recently have link quality prediction been applied to link quality estimation, *i.e.*, including the inferences of the future link quality in a link quality estimate. State-of-the-art approaches in this field include online and offline variants of such predictors [23, 24], using Bayes classifier, logistic regression, logistic regression trained using gradient descent, and artificial neural networks on physical and link layer features: PRR, RSSI, SNR, and LQI. In [23], researchers try to predict the probability of delivering the next packet, while in [24], the output of a learning algorithm is the probability that the future reception rate on a link will be greater than a predefined threshold θ , during a short period of time t .

Chapter 2

Data preprocessing and exploration

This chapter provides an overview of the datasets used in this thesis. These datasets were summarized for the EU-funded eWINE project and reported in [5].

2.1 The Rutgers dataset

The Rutgers dataset was collected from the Open Access Research Testbed for Next-Generation Wireless Networks (ORBIT) and is publicly available at a CRAWDAD repository [20]. ORBIT is comprised of 128 IEEE 802.11a/b/g radio interfaces attached to 64 static nodes arranged on an 8 by 8 grid. In the rest of this chapter we refer to an individual node as $Node_{(x, y)}$, where x and y are coordinates in a Cartesian coordinate system imposed by the grid.

When building the testbed, the researchers aimed at constructing a physical testbed in a constrained space to create real world multi-hop network. They proposed noise injection as a more flexible option than hardware attenuation and considered methods for mapping real world wireless network topologies onto the testbed [19].

When recording this dataset, 32 nodes, each utilizing two Atheros 5212-

based IEEE 802.11a/b/g network interface controllers (NIC), were used. The traces of three nodes were missing from the publicly available dataset, thus we only use the available 29 traces (812 links in total). The interference was produced at four randomly selected locations with a signal generator and an omnidirectional antenna. The interferers were used to alter the wireless channel state and link conditions; the interference levels were configurable. The receivers were recording all received MAC frames encapsulated with a so-called Prism header that contained bitrate, received signal strength indicator (RSSI), and other physical layer information. Employing Perl scripts on the receiver side, sequence number and RSSI for each correctly received frame were extracted from the logs.

The duration of transmission period for each node was equal to 30 seconds. Since the transmitter sent one packet every 100 milliseconds, we can calculate the total number of sent packets as follows:

$$N_{pkt} = \frac{d}{100 \text{ ms}}, \quad (2.1)$$

where d is the duration of the transmission period in milliseconds. This resulted in 300 transmitted packets at every link.

The Rutgers dataset contains five experiments. The experimenters varied interferers' transmission power in-between the experiments, the transmission power was adjusted by steps of 5 dBm beginning at 0 dBm and ending at -20 dBm.

2.1.1 Data cleaning

The NICs used when recording this dataset return an RSSI value between 0 and 127 with 128 indicating an invalid value. We clean the dataset by employing a filter that removes invalid values. The filter is applied to all traces.

2.1.2 Data statistics

The dataset consists of 5 experiments; each experiment contains 812 links connecting 29 nodes which were monitored. Each node was transmitting in broadcast mode, hence there should have been 8400 measurements for every single transmission recorded by the 28 listening nodes, totaling 243600 packets per experiment. Nevertheless, some packets were lost, the number of missing packets depends on the node transmission power and distance between the interference generator and the actual link.

We statistically describe the dataset by computing five number summaries for each link's RSSI measurements. In Table 2.1, we illustrate these numbers for three representative links. The links are connecting $Node_{(1, 2)}$ (the transmitter) and $Node_{(5, 4)}$, $Node_{(7, 2)}$, and $Node_{(2, 5)}$ as listeners. Interference generator's transmission power was set to 0 dBm. From the table, it can be seen that PRR can vary from 22% for a very bad link and up to 94.66% for very good links. Additionally, we can see that the link with the highest mean RSSI of 4.0 and the lowest standard deviation (S.D.) of 0.8 results in the best packet reception ratio. Put differently, stable links with high mean RSSI are the best in terms of PRR while unstable links with high average RSSI often have lower PRR and behave as transitional links [34].

In addition to calculating the five number summary, we analyze link symmetry by plotting a directional link behavior of a representative pair of nodes, namely $Node_{(2, 5)}$ and $Node_{(1, 2)}$, as depicted in Figure 2.1 and Figure 2.2. Figure 2.1 illustrates RSSI values of all successfully received packets at $Node_{(2, 5)}$ in relation to time (sequence number). Since no packets were received at $Node_{(1, 2)}$ it was not possible to plot an analogous graph for $Node_{(1, 2)}$. We can notice that (based on the number of received packets at each node) links in our dataset are not always symmetric. This can be due to the interference generator and possibly also transceiver, antenna or other hardware particularities.

Figure 2.2 depicts the analysis of the link between the same pair of nodes, in this case, we plot PRR of directional links at different noise power levels.

Table 2.1: Five number summaries for three selected links where $Node_{(1, 2)}$ was transmitting and $Node_{(5, 4)}$, $Node_{(7, 2)}$, and $Node_{(2, 5)}$ were receiving [5].

	$Node_{(5, 4)}$	$Node_{(7, 2)}$	$Node_{(2, 5)}$
No. of missing values	234	113	18
Packet count	66	187	282
Mean RSSI	1.1	2.0	4.0
S.D. RSSI	1.1	1.1	0.8
Min. RSSI	0.0	0.0	2.0
25% RSSI	0.0	1.0	4.0
50% RSSI	1.0	2.0	4.0
75% RSSI	2.0	3.0	4.0
Max. RSSI	5.0	5.0	9.0
PRR	22.00%	62.33%	94.66%

The graph shows that a small change in noise power can cause immense variation in PRR and render a link completely useless. It also confirms that some links are asymmetric.

2.1.3 Preliminary exploration

As a part of preliminary exploration, we calculate a vector of normalized windowed average RSSI values for all links and corresponding PRR values. Figure 2.3 illustrates PRR values against windowed average RSSI values of all links when noise power was set to 0 dBm and window size to 50 packets. The plot shows that for links with RSSI of more than 2 the PRR is at least 90%, while links with RSSI between 0,5 and 2 correspond to PRR values between 20% and 90%. The third category includes links with PRR below 20% and RSSI below 0,5. This plot confirms the well-known shape of the three region link quality [34].

Next, we examine the feasibility of detecting the three zones (bad, inter-

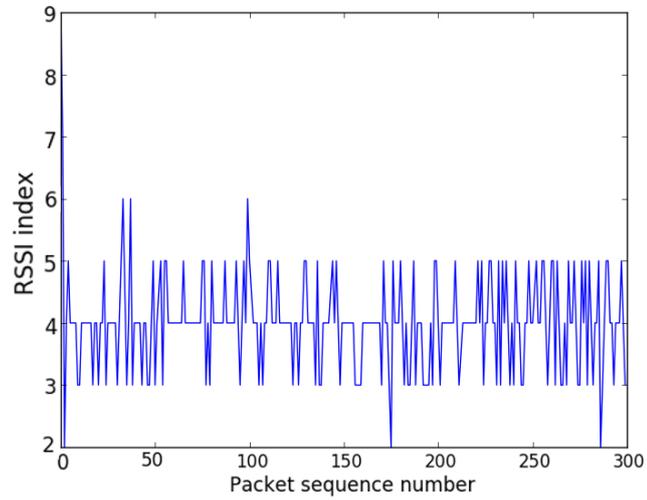


Figure 2.1: RSSI in relation to time for link between $Node_{(1, 2)}$ and $Node_{(2, 5)}$ [5].

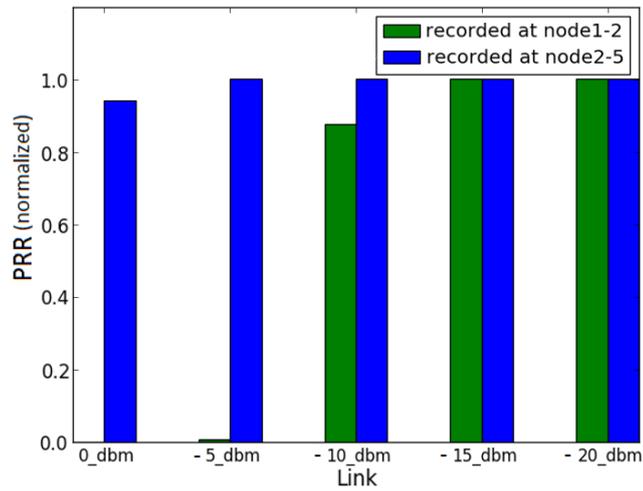


Figure 2.2: PRR of links between $Node_{(1, 2)}$ and $Node_{(2, 5)}$ at miscellaneous noise power levels [5].

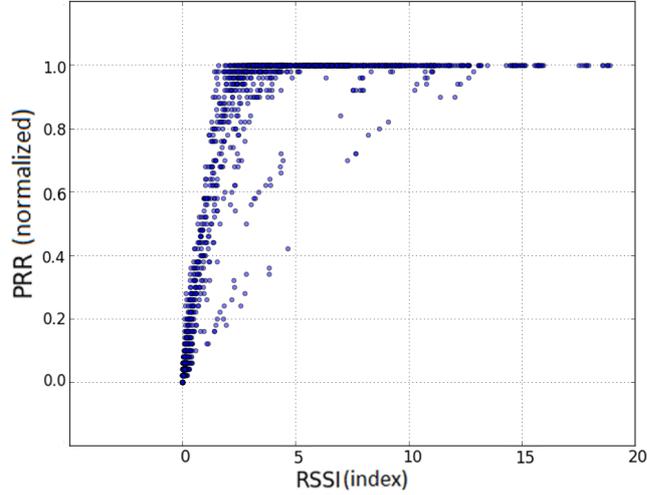


Figure 2.3: PRR versus RSSI of 812 links when the noise level was 0 dBm [5].

mediate and good link quality) [34] by employing automated methods such as unsupervised machine learning. Using the k-means clustering algorithm we partition all data into three clusters using vectors of only normalized windowed average RSSI values and their corresponding normalized RSSI histograms with 40 bins. Applying the algorithm to the data from 0 dBm noise experiment results in three clusters which are depicted in Figure 2.4. Figure 2.4 shows similar link distribution of RSSI and PRR we had in Figure 2.3, this implies that the algorithm is capable of automatically detecting the three regions.

Using the same machine learning algorithm and configuration as previously, we perform clustering on the data from the -15 dBm noise experiment. Clusters with links of good and intermediate quality become very much alike, this may be due to the decreased noise power level which improved links of intermediate quality, that is to say, it seems that previously intermediate links approach the quality of good links while previously bad links improve significantly as well.

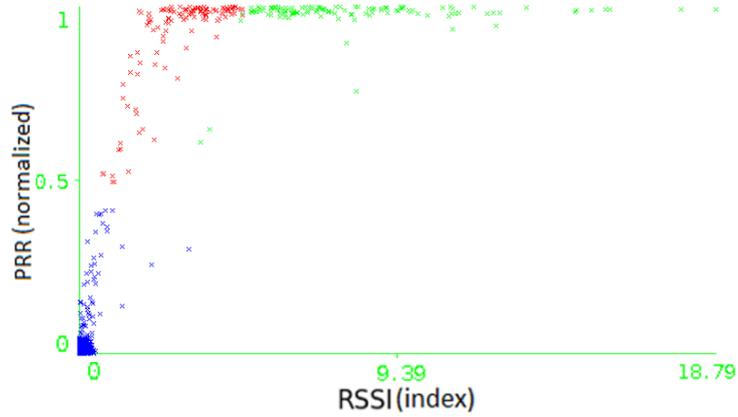


Figure 2.4: Clustering, PRR versus RSSI when the noise level was 0 dBm. Three clusters are shown in distinct colors (blue, red and green) [5].

2.2 The Sigfox dataset

This dataset contains a comprehensive set of RSSI and SNR measurements, collected from Sigfox uplink communication experiments.

In the course of the experiments, a Sigfox base station was mounted on the roof of JSI building C, coupled with a transmitter, which was moved on a trolley through four different indoor locations, as can be seen in Figure 2.5.

The transmitter used in all experiments was USRP N200 with SBX daughterboard and VERT900 antenna, which was in vertical position. Front-end PA gain was varied from 0 dB to 30 dB in steps of 10 dB. Moreover, measurements were made with 30 dB Mini-Circuits attenuator inserted amid the USRP N200 and the antenna.

At each location, 100 packets were sent for each of the four gain settings, this resulted in the total of 1600 measurements. The packet transmission frequency was defined by proprietary Sigfox library, however only the first of three packet repetitions was actually transmitted.



Figure 2.5: Sigfox dataset collection at JSI premises [5].

2.2.1 Data cleaning

After parsing the log files, which contain data for all experiments, we coupled the corresponding data and acknowledgment packets. Some transmissions were not successful, therefore not all acknowledgment packets were logged. After sorting the packets by their sequence numbers accordingly, we extracted packets for each experiment and marked the failed transmissions (missing packets). Information about missing packets will be used later in the thesis. There were no errors or particularities in the data whatsoever.

2.2.2 Data statistics

Considering that experiments were conducted with four gain setting on four different locations, the dataset effectively contains 16 links, connecting two nodes, one transmitting device and one base station. All data transmissions were uplink. The transmitting node sent 100 packets in total for every distinct location and gain configuration, totaling 1600 measurements. However,

Table 2.2: Five number summaries for three selected links [5].

	Loc. 0, gain 20	Loc. 2, gain 0	Loc. 0, gain 30
No. of missing values	6	39	17
Packet count	94	61	83
Mean RSSI	9.0	1.6	8.0
S.D. RSSI	1.6	0.4	2.4
Min. RSSI	1.1	0.0	2.9
25% RSSI	9.1	1.4	8.0
50% RSSI	9.4	1.7	9.1
75% RSSI	9.4	1.7	9.4
Max. RSSI	10.0	2.3	9.7
PRR	94%	61%	83%

some packets were lost, failures in transmission happened due to interferences in addition to changes in parameters and Sigfox protocol imposed by the researchers.

We statistically describe the data by computing five number summaries for each link's SNR and RSSI measurements. In Table 2.2, we illustrate a portion of these numbers for a few representative links. We can see that PRR varies from 61% (intermediate link) to 94% (good link). Additionally, the link with highest mean RSSI is also the best in terms of PRR, while the link with lowest mean RSSI is the worst in terms of PRR. Links with high average RSSI and low variation of RSSI perform worse than links with high average RSSI and low variation. In other words, stable links with high mean RSSI are the best in terms of PRR while unstable links with high average RSSI often have lower PRR ratio and behave as transitional links.

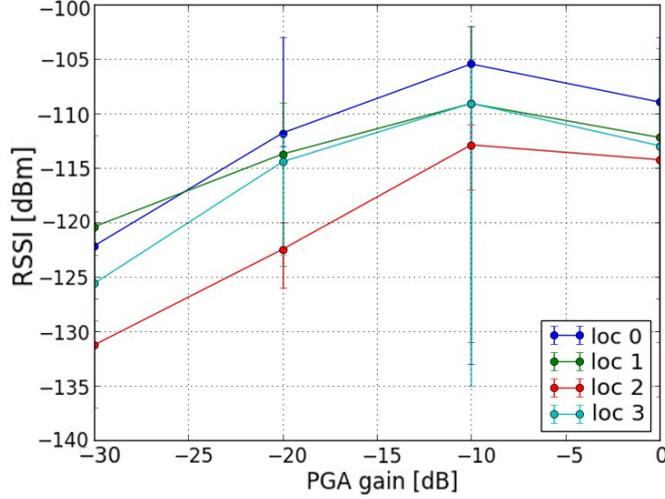


Figure 2.6: RSSI versus transmission power gain for all locations [5].

2.2.3 Preliminary exploration

We explore the interrelation of transmission power with RSSI and SNR. After plotting every pair of attributes, we notice that, as expected, the average of RSSI and SNR in general increases with the intensification of transmission power. For brevity, we only show the graph of RSSI versus transmission power gain in Figure 2.6. The unexplained drop in RSSI may be attributed to having insufficiently large dataset additionally to other hardware and software particularities.

Figure 2.7 depicts PRR against average RSSI for all links. The figure shows that link quality and physical parameters, namely RSSI, appear to be correlated. Physical layer parameters are directly linked to current characteristics of a wireless channel, so link quality and physical parameters are usually tightly coupled [23].

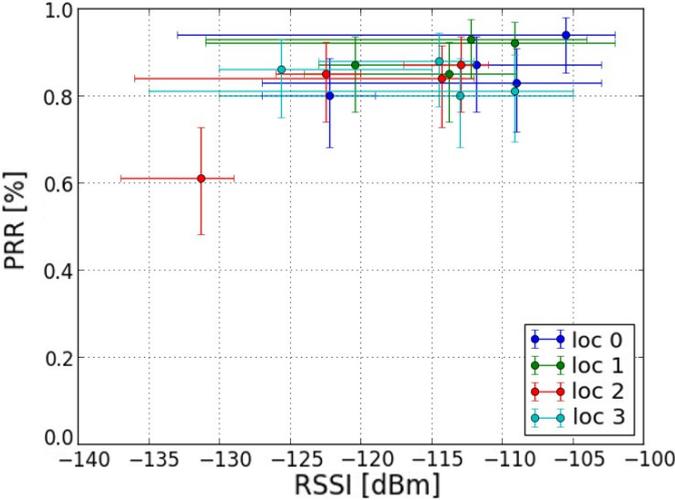


Figure 2.7: PRR versus RSSI for all links [5].

Chapter 3

Development of the predictor for link quality

This chapter illustrates the process of development of a predictor for link quality as previously reported in [5].

3.1 Algorithm selection

We use the J48 decision trees from WEKA to build the link quality prediction models. These decision trees are based on the classical C4.5 algorithm for machine learning [27]. This algorithm uses the entropy, an information theoretic metric, to select the attribute that discriminates the most in the dataset to be higher in the tree. In other words, the splitting criterion is the normalized information gain (difference in entropy). The attribute with the highest normalized information gain is chosen to make the decision. Besides performing the classification task, this algorithm is also useful for understanding which features are more relevant than others in explaining the dataset.

There are several settings (see Figure 3.1) we can choose when running the algorithm, the most important refers to pruning the tree. By pruning, we avoid overfitting the model to the available dataset, thus creating a model that is likely to perform similarly on other, previously unseen data points.

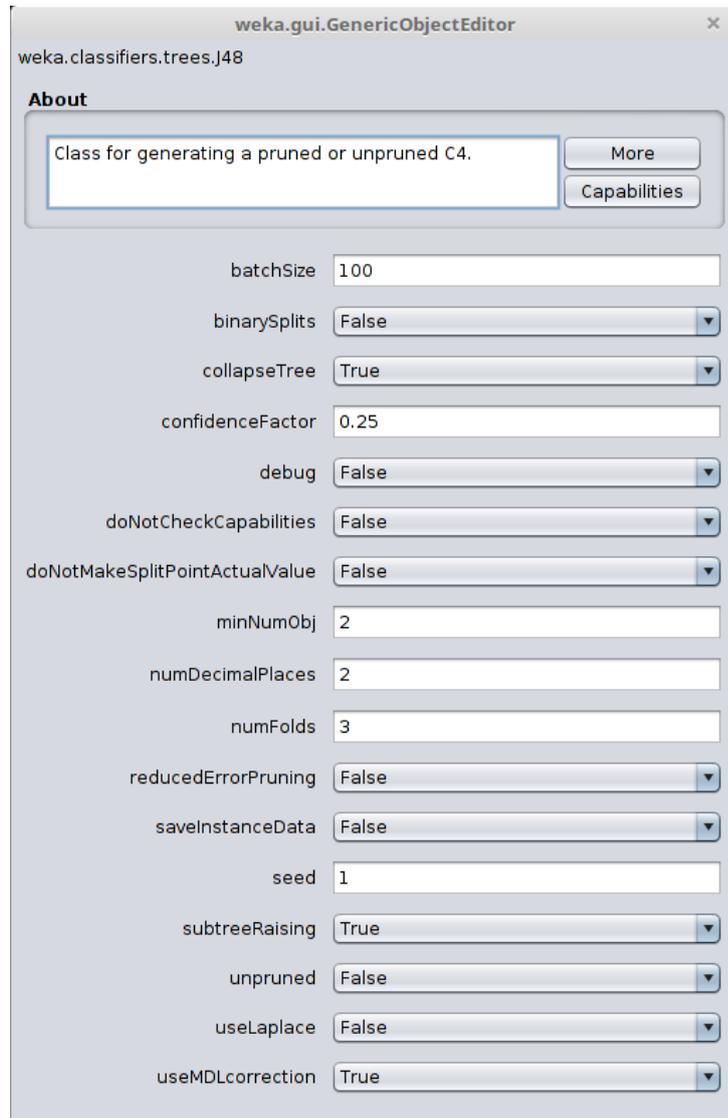


Figure 3.1: Settings for the decision tree classifier in WEKA [5].

The lower the value of the confidence factor, the more pruned the tree is. We evaluated the model with several values (0.25 and below).

3.2 Feature engineering

Existing link quality estimators that use adaptive algorithms based on machine learning to better predict the link use a subset or a linear combination of the following wireless parameters (also called features in machine learning parlance): PRR, RSSI, SNR and LQI [23, 24]. Other well-performing estimators using more traditional algorithms use PRR [34] and PRR, ETX and LQI [11].

In the remaining of the thesis, we use the two datasets described in Chapter 2: Wi-Fi from Rutgers University, and the Sigfox from JSI. For the Wi-Fi from Rutgers we are able to use RSSI and PRR for generating the feature vectors, while for Sigfox we are able to use RSSI, SNR, and PRR. PRR is calculated based on the available sequence numbers and is used only for defining the target class (output of our prediction system), while physical layer information and its aggregations represent the input to our system. PRR is computed for every data point (packet) by dividing the number of successfully received packets by a number of packets that should have been received between the current data point D_i (inclusive) and previous data point D_{i-H} (exclusive). H denotes the number of prior data points ($H \geq 0$) we take into account when calculating PRR in addition to the current data point. When $H = 0$, the PRR at every data point is either 100% (current packet received) or 0% (current packet not received). By adjusting H , we can influence the stability of our predictor.

We compute all possible combinations of feature vectors (see Equation 1.3) and generate corresponding training datasets. When combining and aggregating the physical layer information about previous packets we use different time windows, as per aggregation function Equation 1.4. The number of possible combinations is larger than we can list here (*i.e.*, 127 different

combinations for the Sigfox dataset). Examples of computed input feature vectors are listed below:

- (i) Instant RSSI.
- (ii) Instant RSSI + Avg. RSSI (window 5).
- (iii) Instant RSSI + Avg. RSSI (window 10).
- (iv) Instant RSSI + Avg. RSSI (window 20).
- (v) Instant RSSI + Avg. RSSI (window 5) + S.D. RSSI (window 5).
- (vi) Instant RSSI + Avg. RSSI (window 10) + S.D. RSSI (window 10).
- (vii) Instant RSSI + Avg. RSSI (window 20) + S.D. RSSI (window 20).
- (viii) Instant RSSI + Avg. RSSI (window 5) + S.D. RSSI (window 5) + Instant SNR.
- (ix) Instant RSSI + Avg. RSSI (window 5) + S.D. RSSI (window 5) + Instant SNR + Avg. SNR (window 5).
- (x) Instant RSSI + Avg. RSSI (window 10) + S.D. RSSI (window 10) + Instant SNR + Avg. SNR (window 10).
- (xi) Instant RSSI + Avg. RSSI (window 20) + S.D. RSSI (window 20) + Instant SNR + Avg. SNR (window 20).
- (xii) Instant RSSI + Avg. RSSI (window 5) + S.D. RSSI (window 5) + Instant SNR + Avg. SNR (window 5) + S.D. SNR (window 5).
- (xiii) Instant RSSI + Avg. RSSI (window 10) + S.D. RSSI (window 10) + Instant SNR + Avg. SNR (window 10) + S.D. SNR (window 10).
- (xiv) Instant RSSI + Avg. RSSI (window 20) + S.D. RSSI (window 20) + Instant SNR + Avg. SNR (window 20) + S.D. SNR (window 20).

After transforming all datasets to a common format and calculating new features, we generate the labels (classification categories) using the PRR, as denoted in Equation 1.1. For each instance of a vector we can tell something about the link’s state and correspondingly define its label. For instance, we can say that all vectors in which PRR is below 10% are bad links (label “bad”), all between 10%-90% are intermediate links (label “intermediate”) and all above 90% are good links (label “good”). Example configuration for

generating the feature vectors as explained here is presented in Listing 3.1 for the Sigfox dataset.

Listing 3.1: Configuring the feature generator for the Sigfox dataset.

```
Please input the number of a dataset to be imported. Available datasets:
0 ... jsi_sigfox
1 ... jsi_sigfox_test
4 ... rutgers
> 0
Please input the numbers of experiments to be imported separated by a comma.
Input * to select all experiments. Available experiments:
0 ... sfxlib
1 ... sfxlib_norep
2 ... sfxlib_norep_randfreq
3 ... sfxlib_norep_randfreq_randgain
4 ... sfxlib_norep_randfreq_randgain_30att
> 1,2,3
Import another dataset/experiment? (y/n)
> y
Please input the number of a dataset to be imported. Available datasets:
0 ... jsi_sigfox
1 ... jsi_sigfox_test
4 ... rutgers
> 1
Please input the numbers of experiments to be imported separated by a comma.
Input * to select all experiments. Available experiments:
0 ... sfxlib
1 ... sfxlib_norep
2 ... sfxlib_norep_randfreq
3 ... sfxlib_norep_randfreq_randgain
4 ... sfxlib_norep_randfreq_randgain_30att
> *
Import another dataset/experiment? (y/n)
> n
Successfully imported.
Calculate other features? (y/n)
> y
Input the PRR window. Leave blank to omit.
> 15
Input for which attributes you would like to calculate average/standard
deviation feature in the following format: attr:avg/std:window. To calculate
more features, separate the inputs by a comma. Transformations will be
applied in sequential order. New features are named in the following manner:
attr_mode. Leave blank to omit.
> snr:avg:5, rssi:std:10
Enter the number of leading samples to truncate. Leave blank to omit.
> 15
```

```

Transformations successfully applied.
Define categories? (y/n)
> y
Input the default label.
> intermediate
Input rules for labeling the samples. Rules will be applied in sequential
order. Separate the rules with a comma. Rules must be in the following
format: #label1 $rssi < -50 [and $rssi > 90[,#label2 $attr3 == 10]]
All tokens must be separated with a space. Attribute names must begin with
a $ (dollar) sign.
> #good $pr >= 0.9, #bad $pr <= 0.1
Categories defined.
Input the format of the output files (possible options:
[link|experiment|dataset|all]).
WARNING: THIS WILL DELETE ALL CONTENTS OF OUTPUT DIRECTORY!
> all
Files are successfully generated!

```

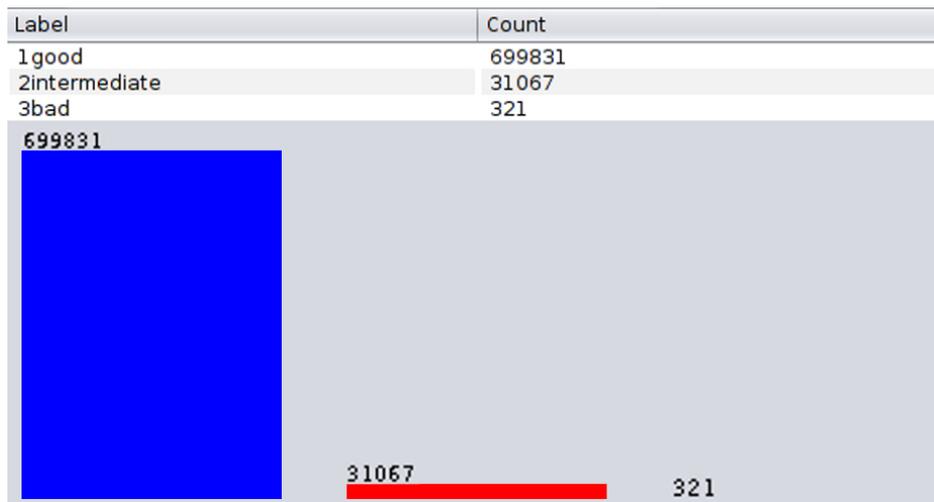


Figure 3.2: Distribution of the classes in the Rutgers dataset (class No. 1 with PRR $> 90\%$ – good links, class No. 2 with PRR between 10% and 90% - intermediate links and class No. 3 with PRR $< 10\%$ - bad links) [5].

We noticed that the resulting training data is significantly unbalanced in favor of good links. This is natural, the recorded datasets contain no information for lost packets. For example, in the Rutgers dataset, there are 699831 data points for good links (96% of the dataset), 31067 for intermediate

links (4% of the dataset) and 321 for bad links (less than 1% of the dataset) – see Figure 3.2. Training a decision model on such a dataset leads to a strong bias for good links. With a simple threshold rule, the model would correctly classify more than 90% of the dataset. In cases when such a biased distribution exists, there are two options:

1. Subsample the dominant class (use a random subset of the data points available for good links) or
2. oversample the minority classes (intermediate and bad links).

In the following we refer to Resampled and Interpolated corresponding to the two options:

Resampled: We use the standard built-in approach in WEKA for resampling the dataset in order to bias the class distribution toward a uniform distribution. We randomly sample with replacement, in this manner, we acquire a new dataset with uniform class distribution, which is the same size as original dataset.

Interpolated: To add more data for bad and intermediate links, we take the following approach. We know the number of packets n that were sent over each link and the sequence numbers of the received packets. As a result, we can identify gaps. Given received sequence numbers S_1 ($RSSI_{S_1}$) and S_m ($RSSI_{S_m}$), we compute the average RSSI as

$$RSSI_{avg} = \frac{RSSI_{S_1} + RSSI_{S_m}}{2}, \quad (3.1)$$

and standard deviation as

$$RSSI_{SD} = \sqrt{\frac{(RSSI_{S_1} - RSSI_{avg})^2 + (RSSI_{S_m} - RSSI_{avg})^2}{2}}. \quad (3.2)$$

Then, for each missing packet with a sequence number between 1 and m , we insert the sequence number and white noise with $RSSI_{avg}$ and $RSSI_{SD}$ into the dataset.

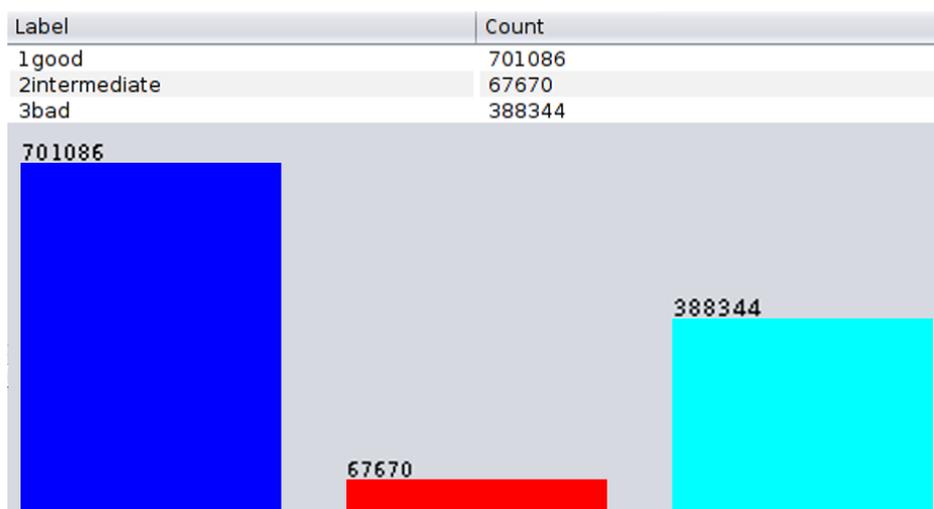


Figure 3.3: Distribution of the classes in the interpolated Rutgers dataset (class No. 1 with $\text{PRR} > 90\%$ – good links, class No. 2 with PRR between 10% and 90% - intermediate links and class No. 3 with $\text{PRR} < 10\%$ - bad links) [5].

As can be seen in Figure 3.3, this results in 701086 data points for good links (60% of data points), 67670 data points for intermediate links (5% of data points) and 388344 for bad links (33% of data points). This training set is more balanced; however, the intermediate links seem to be underrepresented so it is likely the model won't capture them too well (*i.e.*, misclassify intermediate links). To account for that, we should either record a new dataset with more intermediate links or undersample the dominating two classes to improve the accuracy. We do the latter by keeping all data points for intermediate links and randomly sample data points for other two classes. This results in a new dataset with uniform class distribution.

After interpolation, the Sigfox dataset contains only good and intermediate links with 37% of the links being good and 63% intermediate.

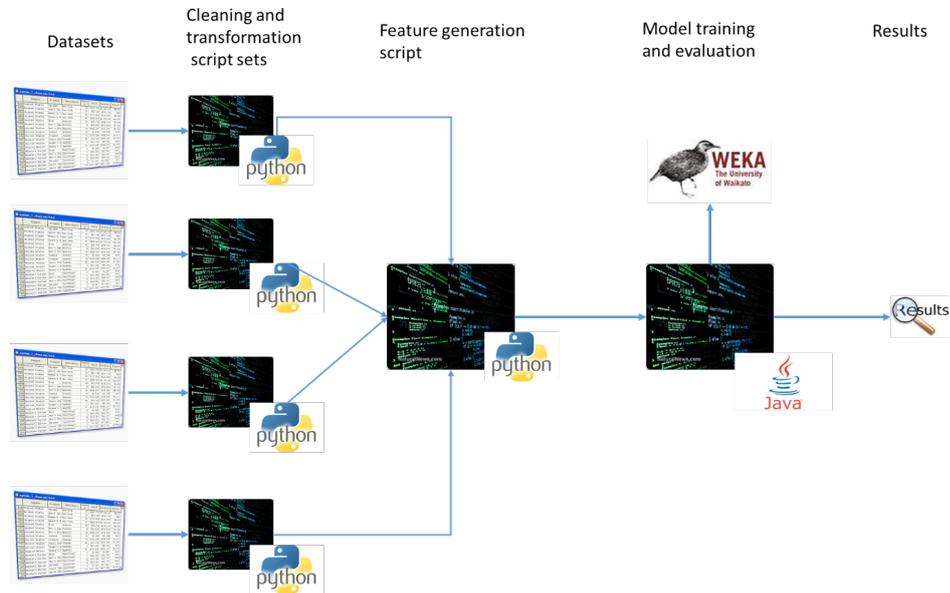


Figure 3.4: Block diagram of the predictor development workflow [5].

3.3 Predictor development workflow

The link prediction system is implemented using a set of Python data cleaning and data transformation scripts, a set of scripts specific for each dataset. Then, it uses a feature generator (see Listing 3.1) that is common for all datasets and is also written in Python. Finally, the models are trained using a Java program that incorporates WEKA. This Java program is used for building custom classification models in bulk by utilizing all possible combinations of input features. The block diagram in Figure 3.4 depicts the interaction between the modules of the system. All source code is available in a public repository¹.

¹<https://github.com/sensorlab/link-quality-estimation>.

Chapter 4

Results

We use cross-validation to evaluate the models. Average RSSI was the most reliable predictor of the link quality in all the experiments, the second was the standard deviation of the RSSI while instant RSSI was only third [31].

4.1 Wi-Fi link quality prediction

The classification results for the Wi-Fi dataset from Rutgers University are presented in Table 4.1. It can be seen that using interpolation for the missing data enables developing the most accurate classifier which correctly classifies 95.8% of the test instances. Undersampling the majority classes in an effort to balance the dataset and enable better classification for the minority class, represented by the intermediate link, decreases the performance of the classifier to 88.7%. Standard statistical resampling is less useful than interpolation, resulting in a classifier that correctly classifies only 77.2% of the test instances correctly.

In terms of feature vectors, it can be seen that the largest feature vector (RSSI, Avg. RSSI, S.D. RSSI) gives the best classification results and having smaller feature vectors (Avg. RSSI, S.D. RSSI or RSSI, S.D. RSSI) costs about 1% in performance.

Table 4.1: Classification accuracy for the Wi-Fi dataset from Rutgers [5].

		Resampled	Interpolated	Interpolated and undersampled
RSSI	Correct class	65.2%	93.5%	83.7%
	Incorrect class	34.8%	6.5%	16.3%
Avg. RSSI	Correct class	69.3%	94.0%	86.8%
	Incorrect class	30.7%	6.0%	13.2%
S.D. RSSI	Correct class	46.4%	91.0%	81.3%
	Incorrect class	53.6%	9.0%	18.7%
RSSI, Avg. RSSI	Correct class	71.6%	95.5%	88.0%
	Incorrect class	28.4%	4.5%	12.0%
RSSI, S.D. RSSI	Correct class	70.7%	95.7%	87.5%
	Incorrect class	29.3%	4.3%	12.5%
Avg. RSSI, S.D. RSSI	Correct class	72.6%	94.9%	87.7%
	Incorrect class	27.4%	5.1%	12.3%
RSSI, Avg. RSSI, S.D. RSSI	Correct class	77.2%	95.8%	88.7%
	Incorrect class	22.8%	4.2%	11.3%

4.2 Sigfox link quality prediction

All models built on the resampled dataset fail to learn from data and simply assign all instances to the majority class. The same problem occurs when modeling the undersampled dataset, most likely due to insufficient dataset size. We omit the resampled and undersampled data from further analysis and only consider the models built using the interpolated dataset.

The classification models for the interpolated Sigfox dataset perform significantly poorer than for the Wi-Fi from Rutgers dataset with the best performing model classifying only 78.8% of the test examples correctly (see Table 4.2). This can be partly explained by the fact that in the Sigfox dataset we have only good and intermediate links (37% good links and 63% intermediate links). Intermediate links tend to be quite unstable, sometimes temporarily behaving as good links, other times as bad links so they might be confused by the model. We may also attribute the mediocre results to

Table 4.2: Classification accuracy for the Sigfox dataset from JSI [5].

Feature vector	Correct class	Incorrect class
RSSI	65.2%	34.8%
Avg. RSSI	69.3%	30.7%
S.D. RSSI	46.4%	53.6%
RSSI, Avg. RSSI	71.6%	28.4%
RSSI, S.D. RSSI	70.7%	29.3%
Avg. RSSI, S.D. RSSI	72.6%	27.4%
RSSI, Avg. RSSI, S.D. RSSI	77.2%	22.8%
SNR	62.6%	37.4%
Avg. SNR	62.2%	37.8%
S.D. SNR	62.6%	37.4%
SNR, Avg. SNR	68.2%	31.8%
SNR, S.D. SNR	69.7%	30.3%
Avg. SNR, S.D. SNR	62.6%	37.4%
SNR, Avg. SNR, S.D. SNR	62.5%	37.5%
S.D. SNR, Avg. SNR, S.D. RSSI, Avg. RSSI, RSSI, <i>avgSnr</i>	78.8%	21.2%
S.D. SNR, S.D. RSSI, Avg. RSSI, <i>avgSnr</i>	78.0%	22.0%
S.D. SNR, Avg. SNR, SNR, S.D. RSSI, Avg. RSSI, <i>avgSnr</i>	77.5%	22.5%
S.D. SNR, Avg. SNR, Avg. RSSI, <i>avgSnr</i>	77.4%	22.6%

intrinsic difference between Sigfox and Wi-Fi link abstractions. Wi-Fi uses a predetermined static wireless channel for all packet transmissions while Sigfox is performing dynamic channel hopping at every transmission as discussed in Section 1.1 and further observed in [12].

Also in this experiment, if we just consider RSSI based feature vectors, it can be seen that the combination of RSSI, Avg. RSSI, S.D. RSSI performs the best, similar to the Wi-Fi from Rutgers case. If we also consider SNR based feature vectors, it can be seen that they perform poorly with up to 69.7% accuracy. However, RSSI and SNR based vectors such as vectors with S.D. SNR, Avg. SNR, S.D. RSSI, Avg. RSSI, RSSI, *avgSnr*¹ lead to the best results.

It is worth noticing that the model resulting from a one-dimensional fea-

¹*avgSnr* is an average SNR computed over the last 25 packets by a Sigfox base station, Avg. SNR is a feature generated by us.

Table 4.3: Normalized confusion matrices for the best performing classifiers on various datasets from the Rutgers University.

	Real/predicted	Good	Intermediate	Bad	<i>Precision</i>	<i>Recall</i>
Raw	Good	95.7	0	0	95.7%	100%
	Intermediate	4.25	0	0	0%	0%
	Bad	0.04	0	0	0%	0%
Interpolated	Good	60.1	0.4	0.09	98.3%	99.2%
	Intermediate	0.98	2.78	2.09	72.7%	47.5%
	Bad	0.03	0.64	32.89	93.8%	98%
Interpolated/undersampled	Good	31.87	1.46	0.01	92.8%	95.6%
	Intermediate	2.48	27.69	3.17	83.1%	83.1%
	Bad	0.01	4.19	29.13	90.2%	87.4%

ture vector S.D. RSSI, behaves about the same as random guessing for Sigfox, about 50% correct classification, while the same for Wi-Fi from Rutgers correctly classified 80-89% of the test instances.

4.3 Discussion

The datasets analyzed in this thesis are fairly unbalanced, some target classes are more represented than the others. For example, in the Rutgers dataset we have 96% good links, meaning that if we do not perform any machine learning and just apply the majority classifier (simple threshold rule), the overall classification accuracy would be higher than what we achieved with machine learning. Although the classification accuracy is high, we misclassify all data points associated with the underrepresented classes, since those classes are simply ignored.

In order to not overfit the majority class and decrease the classification error of underrepresented classes, mainly the intermediate class, we interpolate the datasets to add more data and subsample the overrepresented classes to achieve a uniform class distribution, as discussed in Chapter 3. While the overall model classification accuracy drops after the subsampling,

we notice an increase in precision as well as recall for intermediate links by 10% and 36% respectively (see Table 4.3). The model built on subsampled data can thus predict the intermediate links more accurately and completely, but comes at a cost of a slightly degraded performance when it comes to good and bad links.

Chapter 5

Conclusions

In this thesis we proposed, implemented and evaluated a novel approach to developing a link quality predictor based on feature engineering. Part of work was published in a peer reviewed paper [31]. We examined two datasets, performed feature engineering and built a classification model. Packet reception ratio, physical layer information, and aggregated physical layer information are used as input and the model outputs the predicted link quality. The evaluation showed that our models perform very well (95% correctly classified data instances) on Wi-Fi dataset. The model performance on Sigfox data suffers from protocol particularities explained in Section 1.1. The proposed models vary in performance with respect to accuracy and completeness of predicting different types of links, mainly links of intermediate quality. While some models tend to predict the intermediate links more accurately and completely, they suffer from a slightly degraded performance when it comes to good and bad links. The use of presented models in wireless networks, especially routing protocols, could potentially improve the performance of any wireless network significantly.

Further work includes many directions. Intermediate links are known to be hard to estimate accurately [23]. Improving the model to better predict such links would be beneficial for the general use. Other classification models should be included in the evaluation. An important unanswered question is

how different time window lengths impact classifier performance. Finally, evaluation of link quality estimators' performance is a challenging task [4]. A more extensive evaluation of the proposed models (*e.g.*, on a real world system) is needed.

Bibliography

- [1] *About International Telecommunication Union (ITU)*. ITU. <https://www.itu.int/en/about/Pages/default.aspx>. Accessed: 2 August 2017.
- [2] *R*. R Project. <https://www.r-project.org/about.html>. Accessed: 28 June 2017.
- [3] *RapidMiner*. Wikipedia. <https://en.wikipedia.org/wiki/RapidMiner>, 2017. Accessed: 28 June 2017.
- [4] Nouha Baccour, Anis Koubâa, Luca Mottola, Marco A. Zúñiga, Habib Youssef, Carlo A. Boano, and Mário Alves. Radio link quality estimation in wireless sensor networks: A survey. *ACM Transactions on Sensor Networks (TOSN)*, 8(4):34, 2012.
- [5] Adnan Bekan, Klemen Bregar, Timotej Gale, Carolina Fortuna, Tomaz Solc, Merima Kulin, Wei Liu, Vincent Sercu, Pieter Becue, Bart Jooris, Adnan Shahid, Eli De Poorter, Ingrid Moerman, Yi Zhang, Anatolij Zubow, Niels Karowski, and Filip Lemic. D5.1: Machine learning algorithms development and implementation. Technical report, eWINE project, 2016.
- [6] Michael R. Berthold, Nicolas Cebron, Fabian Dill, Thomas R. Gabriel, Tobias Kötter, Thorsten Meinl, Peter Ohl, Kilian Thiel, and Bernd Wiswedel. Knime-the konstanz information miner: version 2.0 and beyond. *AcM SIGKDD explorations Newsletter*, 11(1):26–31, 2009.

-
- [7] Carsten Bormann, Mehmet Ersue, and Ari Keranen. Rfc 7228: Terminology for constrained-node networks, 2014.
- [8] Per Christensson. *Node*. TechTerms. <https://techterms.com/definition/node>, 2006. Accessed: 11 April 2017.
- [9] Christopher Clifton. *Data mining*. Encyclopaedia Britannica. <https://www.britannica.com/technology/data-mining>, 2009. Accessed: 7 April 2017.
- [10] Janez Demšar and Blaž Zupan. Orange: Data mining fruitful and fun—a historical perspective. *Informatica*, 37(1), 2013.
- [11] Rodrigo Fonseca, Omprakash Gnawali, Kyle Jamieson, and Philip Levis. Four-bit wireless link estimation. In *HotNets*, 2007.
- [12] Timotej Gale. Industrial practice, final report. Technical report, Faculty of Computer and Information Science, University of Ljubljana, Ljubljana, Slovenia, May 2017.
- [13] Dattatraya Gokhale, Sayandeep Sen, Kameswari Chebrolu, and Bhaskaran Raman. On the feasibility of the link abstraction in (rural) mesh networks. In *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*, pages 61–65. IEEE, 2008.
- [14] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The WEKA data mining software: an update. *SIGKDD Explorations*, 11(1):10–18, 2009.
- [15] Jiawei Han, Jian Pei, and Micheline Kamber. *Data mining: concepts and techniques*. Elsevier, third edition, 2011.
- [16] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning: data mining, inference and prediction*. Springer, 2 edition, 2009.

-
- [17] Matthew Haughn. *LPWAN (low-power wide area network)*. TechTarget. <http://internetofthingsagenda.techtarget.com/definition/LPWAN-low-power-wide-area-network>, 2015. Accessed: 10 August 2017.
- [18] Richard M. Karp. On-line algorithms versus off-line algorithms: How much is it worth to know the future? In *Proceedings of the IFIP 12th World Computer Congress on Algorithms, Software, Architecture - Information Processing '92, Volume 1 - Volume I*, pages 416–429, Amsterdam, The Netherlands, 1992. North-Holland Publishing Co.
- [19] Sanjit K. Kaul, Marco Gruteser, and Ivan Seskar. Creating wireless multi-hop topologies on space-constrained indoor testbeds through noise injection. In *Testbeds and Research Infrastructures for the Development of Networks and Communities, 2006. TRIDENTCOM 2006. 2nd International Conference on*, pages 10–pp. IEEE, 2006.
- [20] Sanjit K. Kaul, Ivan Seskar, and Marco Gruteser. CRAWDAD dataset rutgers/noise (v. 2007-04-20). Downloaded from <http://crawdad.org/rutgers/noise/20070420>, April 2007.
- [21] Merima Kulin, Carolina Fortuna, Eli De Poorter, Dirk Deschrijver, and Ingrid Moerman. Data-driven design of intelligent wireless networks: An overview and tutorial. *Sensors*, 16(6):790, 2016.
- [22] Ayyappa Kys. *Types Of Wireless Networks One Should Know*. LinkedIn. <https://www.linkedin.com/pulse/types-wireless-networks-one-should-know-ayyappa-kys>, 2016. Accessed: 1 August 2017.
- [23] Tao Liu and Alberto E. Cerpa. Foresee (4c): Wireless link prediction using link features. In *Information Processing in Sensor Networks (IPSN), 2011 10th International Conference on*, pages 294–305. IEEE, 2011.

- [24] Tao Liu and Alberto E. Cerpa. Talent: Temporal adaptive link estimator with no training. In *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*, pages 253–266. ACM, 2012.
- [25] Lauren Lynn. *Lab Happenings Vol 2*. BlueFletch. <https://bluefletch.com/blog/lab-happenings-vol-2/>, 2016. Accessed: 3 August 2017.
- [26] Chebiyyam S. R. Murthy and Balakrishnan Manoj. *Ad hoc wireless networks: Architectures and protocols, portable documents*. Pearson education, 2004.
- [27] Ross Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, San Mateo, CA, 1993.
- [28] Shalini Ramamurthy. *Wireless Sensors: Technologies and Global Markets*. BCC Research, September 2016.
- [29] Marc Reed. *Licensed vs. Unlicensed Spectrum for Utility Communications*. Utility products. http://www.utilityproducts.com/articles/print/volume-7/issue-7/product-focus/amr_ami/licensed-vs__unlicensed.html, 2017. Accessed: 2 August 2017.
- [30] John Ross. *The book of wireless: A painless guide to wi-fi and broadband wireless*. No Starch Press, 2008.
- [31] Tomaz Šolc, Timotej Gale, and Carolina Fortuna. Optimization of ultranarrowband wireless communication: an experimental case study. In *4th International Workshop on Computer and Networking Experimental Research Using Testbeds (CNERT), IEEE INFOCOM 2017*, Atlanta, Georgia, USA, May 2017.
- [32] Rob van der Meulen. *Gartner Says 8.4 Billion Connected "Things" Will Be in Use in 2017, Up 31 Percent From 2016*. Gartner. <https://www.gartner.com/newsroom/id/3598917>, 2017. Accessed: 15 June 2017.

-
- [33] Hayden Wimmer and Loreen M. Powell. A comparison of open source tools for data science. *Journal of Information Systems Applied Research*, 9(2):4, 2016.
- [34] Alec Woo, Terence Tong, and David Culler. Taming the underlying challenges of reliable multihop routing in sensor networks. In *Proceedings of the 1st international conference on Embedded networked sensor systems*, pages 14–27. ACM, 2003.