UNIVERSITY OF TECHNOLOGY SYDNEY

Faculty of Engineering and Information Technology

Data Classification and Transportation in Rail Networks

by

Mahdi Saki

Principle Supervisor

A/Prof. Mehran Abolhasan

Co-Supervisor

A/Prof. Justin Lipman

THESIS SUBMITTED IN PARTIAL FULFILMENT OF
THE REQUIREMENTS FOR THE DEGREE

Doctor of Philosophy

Sydney, Australia

2020

CERTIFICATE OF ORIGINAL AUTHORSHIP

I, Mahdi Saki, declare that this thesis is submitted in fulfilment of the requirements for the award of PhD, in the School of Electrical and Data Engineering/Faculty of Engineering and IT at the University of Technology Sydney. This thesis is wholly my own work unless otherwise referenced or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

This document has not been submitted for qualifications at any other academic institution.

Signature:

Production Note:
Signature removed prior to publication.

Date: 18/01/2021

Abstract

IoT is a revolutionary technology in the digital world, with a diverse range of services being created and deployed. One of the major challenges involved in efficiently implementing IoT is the management and transportation of large volumes of data that this solution generates. Modern approaches for IoT completely rely on cellular networks. As the demand for such networks is massively growing, in this thesis, we explore other communication methods as alternatives for management and delivery of IoT data in rail networks. Particularly, the focus will be on developing strategies that utilize existing trains and the rail network as a mode of data transportation. Furthermore, the thesis will combine physical delivery of IoT data by trains to strategic collection points in rail networks with cellular infrastructure to minimize costs and increase communication scalability and efficiency. Therefore, in this thesis, we introduce a new framework into future data-driven rail networks. For this purpose, we propose an edge processing unit that includes two main parts. The first part is a data classification model that classifies IoT data into maintenance-critical data (MCD) and maintenance-non-critical data (MnCD). The second part is a data transmission unit that based on the class of data, employs appropriate communication methods to transmit data to strategic collection points. The MCD is immediately forwarded through real-time communication methods such as cellular networks. However, for the transmission of MnCD, we propose three travel pattern methods including train-to-station (T2S), train-to-train (T2T) and train-to-wayside (T2W) communications that employ trains as data carriers. We validate the

classification model and all the transmission methods through extensive experiments. The simulation results show the effectiveness of our models as follows. The data classification model was validated under different operating conditions with over 98% accuracy. For the T2S model, we showed that over 5 GB data can be offloaded through T2S communications. Additionally, our proposed mobility model for T2T communications was tested with real GPS data and showed over 98% accuracy. Furthermore, for the T2W communications, we showed that the proposed AP placement approach could improve the efficiency of data offloading up to 165%. Finally, we proved that we can offload over 250 Gigabits through T2W communications over WiFi networks.

To my wife, Atefeh,

and my sons, Kian and Ryan

Acknowledgment

This thesis would not have been fulfilled without the supervision and guidance of many individuals who contributed and extended their valuable assistance in the preparation and completion of this study.

At first, I offer my sincerest gratitude to my principal supervisor, A./Prof. Mehran Abolhasan, who has supported me throughout my thesis with his patience and knowledge. I attribute the level of my Ph.D. degree to his encouragement and effort, and without his support, this thesis would not have been completed.

I am grateful to my co-supervisor, A./Prof. Justin Lipman, for his unconditional support and valuable guidance during this thesis. It is an honour for me to work with him during my studies. He has not been only a great advisor, but also an encouraging and motivating friend.

My sincere thanks goes to Prof. Abbas Jamalipour, from the University of Sydney, for his contribution in two of my papers that both were published in the prestigus journals of IEEE Transactions. I really learned a lot from his professional comments and wish to use his constructive advice in the other works in the future.

I greatly appreciate the financial support from the Rail Manufacturing Cooperative Research Centre (funded jointly by participating rail organisations and the Australian

Contents

List of Figures

List of Tables

List of Algorithms

Technical Terms and Acronyms

2-D   two-dimensional

3-D   three-dimensional

A-GPS Assisted GPS

AARF Adaptive Auto Rate Fallback

AP   access point

AP   access point

BDA   Big Data Analytics

CBM   condition-based monitoring

DFT   Discrete Fourier Transform

ECA   environment class arrangement

ECR   environment class ratio

EDP   equally distributed placement

EOP   energy optimization problem

FE   feature extraction

FSPL free space path-loss

FT   Fourier Transform

HP   hybrid placement

iDMM IoT data management module

IoT   internet of things

ISM   Industrial, Scientific, and Medical

ITS   Intelligent transportation system

LoS   line of sight

LTE-R Long Term Evolution-Railway

MBP   Measurement-Based Placement

MCD   maintenance-critical data

MCS   modulation and coding scheme

ML   machine learning

MnCD maintenance-non-critical data

OP   optimal placement

OSU   on-board storage unit

PC   principle component

PCA   Principle Component Analysis

PL   path-loss

PSD   Power Spectral Density

RBAR   Receiver Based Auto Rate

RCM   railway condition monitoring

RMSE   Root Mean Squared Error

RSS   received signal strength

RTS   Rail transportation systems

SNR   signal-to-noise ratio

SVM   Support Vector Machine

T2S   train-to-station

T2T   train-to-train

T2W   train-to-wayside

T2W   train-to-wayside

TCM   train contact model

TMM   train mobility model

UWB   ultra wide-band

WLAN wireless local area network

Nomenclature and Notation

Chapter 3

Symbol	Description
x	data sample
N	number of samples
X	matrix of data
i	bearing number
j	segment number
c	channel number
n	number of channels
m	number of sensors
F	matrix of features
$\hat{F}$	matrix of effective features
l	number of effective features
w	window length
d_{tot}	total delay
d_{col}	duration of data collection
d_{alg}	algorithm processing time

Chapter 4

Symbol	Description
t_r	WiFi resilience time
t_{en}	entering time
t_{dw}	dwelling time
t_{lv}	leaving time
stp	stopping station
ps	passing station
d	displacement
a	acceleration
v	velocity
P_{ref}	received power at reference distance
P_t	transmitter power
λ	wavelength
c	light speed
f	radio carrier frequency
n^{dBm}	noise in dBm
γ	path-loss exponent
bw	bandwidth
N_{ss}	number of spatial streams
A	offloading capacity
th	throughput
N_{stp}	number of stopping stations
N_{ps}	number of passing stations

Chapter 5

Symbol	Description
v	velocity
d	distance
a_1	traction acceleration
a_4	breaking deceleration
a_3	coasting deceleration
a_f	acceleration due to friction
T	total trip time
D	total trip distance
$\overline{x}$	position vector
$\overline{v}$	velocity vector
$\overline{X_c}$	vector of contact positions
$\overline{T_c}$	vector of contact duration
N_c	number of contacts

Chapter 6

Symbol	Description
p_{Tx}	transmitter power
p_{Rx}	receiver power
PL	path-loss
i	index of access points
j	index of train position
PL_0	path-loss at reference distance
G_a	antenna gain
$C_\pm$	clearance between round lines
C_{AP}	clearance between APs and line
l	length of track line
α	ratio of environment class
ECR	environment class ratio
ECA	environment class arrangement
eff	energy efficiency
D	data capacity
E	consumed energy
ρ	MAC efficiency factor
P_N	noise power
PL_{av}	average path-loss

1 Introduction

1.1 Background

In a data-driven based rail network, data is collected from IoT devices placed in all parts of the network [3]. Such data needs to be transmitted to the centers where the data is used for real-time monitoring (during operation) or long-term analysis (e.g. maintenance scheduling and future extension planning) [4] and [5]. As the amount of data is massively growing, finding appropriate transmission methods is essential. In the current rail system, cellular-based networks dedicated for rail applications such as GSM-R and LTE-R (Long Term Evolution-Railway) are used for data transportation [6]. However, as the public demand for cellular networks is rapidly growing, finding other communication methods as appropriate alternatives specially during peak-times can be highly beneficial even after the availability of 5G [7].

As mentioned above, part of the collected data is used for real-time monitoring. Therefore, this part must be immediately sent to control centers. However, the other part that is used for long-term analysis can be transferred in a non real-time manner. This motivated me to develop a classification model that can distinguish the critical data from the rest of it. Then, the critical data, which compromises a small portion of the total data, can be directly transmitted through cellular networks. However, for the delay-tolerant

1

part of data, we can use other communication methods through wireless local area networks (WLANs) currently existing in the rail networks. This will have many benefits, some of which are discussed in the following.

1. This will provide more a reliable communication method for the situation where the cellular network service is poor or unavailable, e.g. congested urban areas during peak hours or remote rural areas.

2. Only a tiny portion of cellular bandwidth needs to be allocated for transmission of critical data and a large amount of data will be transferred through other communication methods.

3. As the data traffic in WLANs has a very low cost, the cost of transmission can be significantly reduced. Additionally, the infrastructure cost to run a small WLAN will be almost zero when comparing with the massive infrastructure costs in cellular or sattellite networks.

4. The collected data is effectively categorized as critical and non-critical data enabling multi-modal communication.

Therefore, we firstly develop a classification model that can classify the collected data into maintenance-critical data (MCD) and maintenance-non-critical data (MnCD) in an online manner [8]. Then, for transmission of MnCD, which is delay-tolerant, we develop three models using train-to-station (T2S) [9], train-to-train (T2T) [10] and train-to-wayside (T2W) [11] communication methods. Figure 1.1 shows those different kinds of communication methods in a rail network. All the proposed models were tested through extensive simulations that showed their effectiveness with high accuracy compared with real experiments.
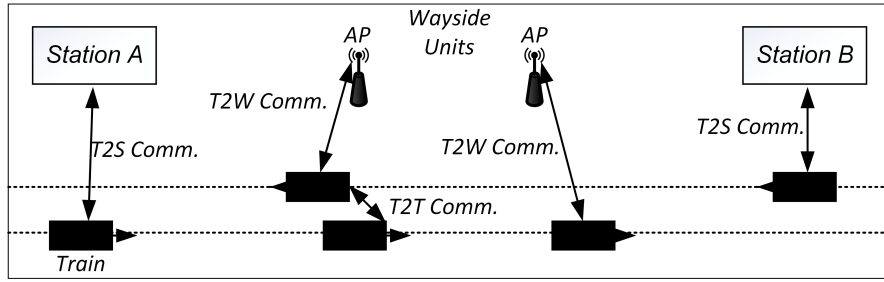
Figure 1.1: Different kinds of communications used in the thesis

1.2 Research Questions

- Is it possible to develop a model that can classifiy critial and non-critical IoT data in a rail network?

- Except cellular communcations, what are other communication methods to transmit large amount of IoT data to data centres?

1.3 Research Objectives and Contributions

- A comprehensive architecture that is appropriate for data transportation and classification in rail networks. The classification model can effectively prioritize the collected IoT data in an online manner. Additionally, as the classification model can be trained while running a train service, it is adaptable to any kind of rolling stock.

- An analytical T2S communication model that can estimate the amount of offloaded data between train and stations when trains are running within the communication zone of stations.

- A T2T communication model that determines the capacity of data transmission between two trains during contacts. As the T2T model has an in-built mobility

model, it can estimate the contact specification of trains in every rail line without requiring any GPS signal.

- A T2W communication model that can enable data transmission not only close to stations but also within the entire length of rail tracks. As we use an optimization function in this model, it can provide an energy-efficient data transmission and in a continuous manner.

1.4 Thesis Structure

For easy understanding of the thesis structure, we illustrated the area of every contribution chapter in Figure 1.2. On this basis, the outlines of all chapters are explained as follows. Introduction and Literature Review are provided in Chapters 1 and 2, respectively. Chapters 3 to 6 are devoted to the main contributions of this thesis. In Chapter 3, we present a comprehensive architecture for data classification and transportation in rail networks. The three communication methods include T2S, T2T and T2W are discussed in chapters 4-6, respectively. Finally, the conclusion and the potential future works are provided in the last chapter.

1.5 List of Publications

- **M. Saki**, M. Abolhasan and J. Lipman, "A Novel Approach for Big Data Classification and Transportation in Rail Networks," in IEEE Transactions on Intelligent Transportation Systems, vol. 21, no. 3, pp. 1239-1249, March 2020.

- **M. Saki**, M. Abolhasan and J. Lipman, "A Big Sensor Data Offloading Scheme in Rail Networks," 2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring), Kuala Lumpur, Malaysia, 2019, pp. 1-6.
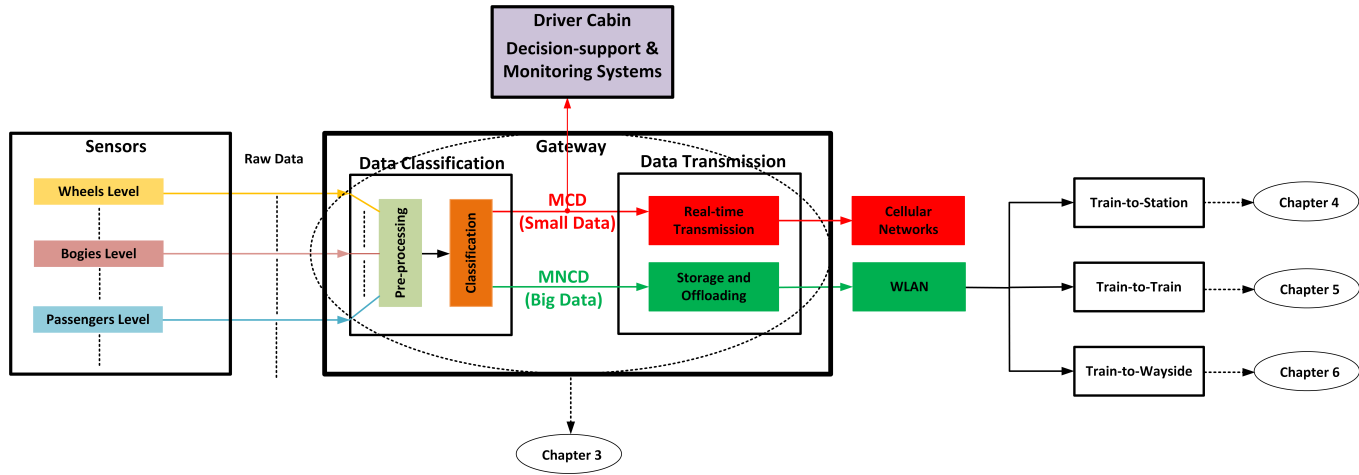
Figure 1.2: Thesis technical overview

- **M. Saki**, M. Abolhasan, J. Lipman, A. Jamalipour, "Mobility Model for Contact-Aware Data Offloading Through Train-to-Train Communications in Rail Networks", Submitted to IEEE Transactions on Intelligent Transportation Systems, pp. 1-13, 2020.

- **M. Saki**, M. Abolhasan, J. Lipman, A. Jamalipour, "A Comprehensive Access Point Placement for IoT Data Transmission Through Train-Wayside Communications in Multi-Scenario Based Rail Networks", Submitted to IEEE Transactions on Vehicular Technology, vol. 69, no. 10, pp. 11937-11949, 2020.

2 Literature Review

In this chapter, we provide a literature review in four areas based on the main contributions of the thesis, including a comprehensive architecture for data classification, and T2S, T2T and T2W communication methods in rail networks.

2.1 Data Classification and Transportation in Rail Networks

Intelligent transportation systems (ITSs) are expected to provide massive amounts of data [12]. Such a level of data is related to various areas such as: maintenance [5] and [13], operation [14] and [15], and safety [16] and [17]. Among these three areas, the application of Big Data Analytics (BDA) in data-driven CBM (condition-based monitoring) systems for railway transportation system has recently received significant attention in the literature [18]. According to the formal definition of Big Data, it is referred to the data that has 3 main features include hoigh volume, velocity and variety [19].As an example to have a picture of CBM data size, [5] provided an estimation showed that the data size for a train bearning sensor can be over 1 terabyte per hour (volume feature). Addtionally, new trains can have various kinds of sensors and IoT devices such as accelerometers, gyroscopes and magnetometers that are rapidly generating different types of data (velocity and variety features) [3]. These shows the need to use Big Data approaches in railway CBM systems.

2 Literature Review

In this section, we discuss the relevant approaches in the area of railway Big Data in two fields: data analysis and data transmission that are two main parts of our proposed data management architecture.

For the data analysis in railway applications, we have chosen the two approaches of [5] and [13] that are the most relevant works in this field. Fumeo et. al. in [5] have used a machine learning technique to estimate the useful life of train axle bearings based on the big data that comes from several on-board sensors. Hongfei et. al. in [13] have employed a combination of analytical techniques to build a prediction model using historical maintenance and failure data. These assume that a large amount of data has been collected before and is available to perform some offline analysis or build a static prediction model. However, condition monitoring data may differ from train to train. Therefore, if we use a static model for different situations, the model can give us wrong results. Hence, we need to build a dynamic model which can be matched with different rolling stocks. Therefore, we need a classification method that can be trained by small amount of data in the case of using in a different rolling stcok. Addtionally, as the proposed model is used for classification of critical data from non-critical data, such method should be reliable too. Support Vector Machine (SVM) is such classification method that requires small data size for training [20]. Furthermore, SVM provides high reliability by seperating the classes with wide margins [20].

For data transmission in railway condition monitoring (RCM) systems, there are several approaches in the literature [21]-[25]. The authors in [24], [26] and [27] have proposed new architectures for just short-term data transmission between sensor nodes in regular cabins and a sink node in the driver cabin. They employed 2.4 GHz ISM (Industrial, Scientific, and Medical) band radios for communications between IoT sensors and an on-board gateway. Tolani et al. in [25] have used a two-layered ZigBee-WiFi communication module for the mentioned internal data exchange. However, the authors in [28]-[30] have

presented more extended architecture by adding the ability of long-range transmission (as well as short-range communication). The authors in [28] and [23] have proposed an architecture using ZigBee for on-board data communication among wagons and the cabin driver and cellular communication for remote data transmission. Macucci et al. in [21] and Chiocchio et al. in [29] have developed a data acquisition and transmission system based on ISM band radios for internal (on-board) and cellular technologies for external (long-range) communications. Gruden et al. in [30] built a prototype which uses 2.45 GHz network for on-board communication and RFID technology for external data transmission. The existing literature has only proposed methods to deal with small volumes of data and is not suitable for the transmission of the large volumes of data generated.

Assuming a three-step data management scheme (that includes data collection from IoT devices to a sink node, edge processing and data transmission between the sink node and a remote center), the abovementioned approaches each worked in one step only. In this approach, we aim to cover 2 out of those 3 steps include edge processing and transmission of IoT data. Therefore, we conclude the existing gaps as follows:

- lack of an onboard data management scheme that can handle data from edge processing to transmission.

- lack of an online classification model that can be trained for any kind of rolling stock during operation.

2.2 Train-to-Station Communication Method

There are several appraoaches in the area of vehicular or railway data offloading in the literature [31]-[34]. Generally, there are two strategies for mobile data offloading via

WLANs including opportunistic and delayed offloading [35]. Opportunistic offloading is applied when a vehicle or a user passes an offloading spot in an opportunistic manner. However, the delayed strategy is for the cases that the data transmission can be delayed until it meets a WLAN access point (AP). In this section, we review the delayed offloading strategy that is used in T2S communication method.

The authors in [36] developed an analytical model for offloading of delay-tolerant data between mobile users via an available WiFi network aiming to maximize the offloading capacity. However, they assumed a fixed data rate for the required WLAN and have not considered the variations of wireless channel states due to user mobility.

Kashihara in [37] has employed a high speed short range communication such as Transfer Jet to collect data from users at bus stops and then offloaded the stored data via fiber optic at terminals. Therefore, the whole collected data must be delayed until a bus reaches a terminal and the author has not provided any solution for data offloading at bus stops. Furthermore, the paper only discussed "stopping" bus stops and has not considered "passing" bus stops as extra potential spots for data exchange.

Huang et. al. in [38] have proposed IEEE802.11p WiFi networks as an alternative communication method to cellular network for data offloading. This approach is suitable for small-scale data offloading and is not suitable for offloading of large amount of data due to low throughput of IEEE802.11p standard.

Due to the above issues include considering fixed-data rate WLAN, ignoring data offloading at passing stations and being non-scalable to large data offloading, we proposed an analytical model that not only employ stopping stations for data offloading in rail networks, but it will also consider the passing stations to maximize the data offloading capacity. This will also restrict the offloading delays to the short trip time between two consecutive stations rather than long-time travels between terminals. Additionally, our

proposed model will utilize a dynamic data rate scheme vs. fixed data rate for data offloading using an appropriate MAC layer rate control algorithm such as RABR and AARF that enables the offloading task to be feasible even with poor WiFi signals. Our approach will also use IEEE802.11ac-based WiFi networks that: 1) are currently available at train stations and there is no need to install any extra infrastructure, 2) can theoretically provide high throughput up to 2.34 Gbps, and 3) has the potential to be upgraded to new rapid offloading technologies such as IEEE802.11ay.

2.3 Train-to-Train Data Communication Method

There are several approaches in the area of vehicular data offloading in the literature [39] and [40]. Baron et al. in [40] proposed an software defined network (SDN)-based scheme to transfer massive delay-tolerant data between two remote data centers using private cars at parking spots for data offloading. In this scheme, the SDN controller allocates appropriate cars as data carriers based on data demand specifications including destination, volume and flows of cars and deadline. They showed that by using their proposed scheme, several petabytes of data can be offloaded by employing only 10% of vehicles equipped with a single terabyte storage device. Zhu et al. used multi-hop data transmission through vehicular opportunistic networks during vehicles contact [41]. They considered both Poisson and exponential distributions to model contact rate and duration, respectively. The authors in [42] developed an adaptive routing algorithm to offload sensor data from on-board units to RSUs through V2V, V2R and V2I communications. Their simulation results showed that 70% of generated vehicular sensor data can be offloaded via their algorithm. Zhu et al. in [1] used T2T communication to improve handoff latency when a train is moving from communication range of one base station (BS) to another one. They developed an algorithm to choose an optimum communication method between T2T and train-to-infrastructure (T2I) communications

aiming to the minimum handoff latency. To achive this, they used optimal guidance trajectory combined with cooperative relaying using T2T communications to increase network reliability and reduce handoff latency. Similarly, [43] and [44] proposed a free-space optics based method to reduce the handoff for train-to-ground communications. In the above works, they all used stochastic approaches such as Poisson distribution to model the contacts between vehicles.

For T2T communications [45]-[47], which is a kind of V2V communications, IEEE802.11p is the state-of-the-art protocol for transmission of safety-critical data among vehicles [48]-[**?**]. There is also a specific version of LTE, called LTE-V2V dedicated for vehicular communications released in 2016 [51]. Bazzi et al. in [51], through comparative simulations showed that LTE-V2V has better performance for moderate vehicle density and higher communication ranges (over 300m). However, IEEE802.11p (WAVE) shows better performance in high vehicle density and lower communication ranges (up to 300m). The main objective of data offloading through T2T communications is to reduce traffic from infrastructure-based networks e.g. cellular networks. Additionally, in Chapter 5, we aim to use T2T communications for data offloading during train contacts when the trains are running very close to each other on parallel tracks. Therefore, we choose IEEE802.11p as the transmission protocol for data offloading that is a global standard for short-range vehicular communications. Since the scope of this approach is just transmission of non-critical data, we will also try IEEE802.11g-based regular WiFi as another communication protocol with higher data rate, to show a higher offloading capacity in comparison with IEEE802.11p. This will be described with more detail in Chapter 5.

2.4 Train-to-Wayside Communication Method

The optimal placement of APs (OPAP) in a wireless network has been addressed in several works in the literature [52]-[60]. He et al. in [52] used DIviding RECTangles (DIRECT) as optimization technique and 3D-ray tracing as propagation model for finding the best locations of transmitters in an indoor design space. They also developed functions for estimation of the system performance. Their criteria for optimization were bit error rate and coverage. They showed that their algorithm can improve the coverage of a given space at least around 10%. Jing et al. in [53] proposed a heuristic algorithm for OPAP so that the total throughput for a given WLAN is maximized. They partitioned the design space into several defined grids and used Two-Ray-Ground model to predict the signal strength of each grid. They showed that their algorithm can provide near-optimal throughput for a given indoor space. However, the algorithms proposed in [52] and [53] cannot determine the required number of APs for a network. The algorithm proposed by Ting et al. in [54] can find both the optimal number and places of APs. They used free space for propagation model and genetic algorithm for solving the optimization problem. They showed that their algorithm can achieve over 98% coverage with optimal number of APs for their six given benchmarks. Liang et al. in [55] proposed two heuristic algorithms to determine the optimal number and places of APs aimed at maximizing the coverage ratio. They also used a specific distance-based logarithmic equation for propagation model that needed to be calibrated by site measurements. Similar to [54], they also showed that their algorithms can provide over 98% coverage with minimum number of APs for 9 different case studies. The algorithms developed in [52]-[55] were proposed for indoor environments such as commercial offices and therefore, were not appropriate for outdoor applications like ITS.

On the contrary, [56]-[60] all provided solutions for OPAP in outdoor environments suitable for ITS. [56]-[58] proposed algorithms for OPAP in vehicular networks (VNs). So

et al. in [56] proposed an algorithm for optimal placement of extension points (EPs) assumed that places of APs were already fixed and defined. Their goal for optimization was to minimize the packet transmission time in a given VN. To model the communication channel, they used IEEE 802.11g as wireless standard and log-normal propagation with zero shadowing effect (i.e. $\sigma = 0$) as PL model. They showed the efficiency of their algorithm for different transmission powers, packet size and various environments. Li et al. in [57] proposed algorithms for optimal placement of gateways (OPoG) for the scenarios of single and multiple gateways in one-dimensional (1-D) and two-dimensional (2-D) VNs. Their objective of OPoG was to minimize the number of hops between APs and gateways. Then, they proposed another optimization function aimed at minimizing power consumption of APs based on OPoG. They showed that with minimizing the average number of hops between APs and gateways, the average power consumption would be also minimized. Zhang et al. in [58] proposed an analytical model for OPAP for a 1-D uni-directional vehicular ad hoc network (VANET). They divided the desired road into clusters with one AP in the middle point of every cluster. Then, they used a binary-search-based algorithm to find the maximum coverage distance of every AP in a cluster aimed at minimizing the number of APs. They showed that their algorithm was appropriate for transmission of both types of real-time and delay-tolerant data.

The authors in [59] and [60] proposed algorithms for OPAP specifically in rail networks. WEN et al. in [59] proposed an algorithm for AP deployment in small-scale rail networks. They divided the design space to several sections based on a pre-defined distance. Then, to find the optimal section for AP placement, they defined a cost function aimed at minimizing outage probability of wireless link between train and APs. To solve the optimization, they used brute force search method that was not appropriate for large-scale rail networks. They showed that the placement of three APs at second, third and fifth sections of a short track (about 300 meter length), which is divided to five sections, will result in the minimum outage probability. Therefore, they did not provided a general

and appropriate solution that can be applied to any large rail network.

Zhang et al. in [60] used vector parabolic equation (VPE) method appropriate for modeling the PL inside of tunnels. They also applied Hooke and Jeeves method for solving the optimization problem aimed at maximizing the system coverage. Their criteria for the optimization function was to minimize the average PL of the network. They also defined another optimization function with the purpose of minimizing the maximum PL at every point of tunnels. They showed that their proposed algorithm can improve the average PL for a given tunnel in a rail network. To estimate the amount of such improvement, they compared the PL between optimal and initial network. However, the initial network used for comparison had not been appropriately selected as they did not divide the rail track into equal sections. Additionally, as the authors mentioned, their VPE-based PL model is appropriate only for tunnels and could not be applied for other scenarios in rail networks. However, there are several other scenarios besides tunnels such as cuttings and viaducts in rail networks.

Therefore, the issues currently existing in the literature are as follows:

- There is no comprehensive approach for OPAPs that is appropriate for any large rail network with any length, different geometric paths and different scenarios.

- While trains may experience a combination of several environments even during one trip, up to our knowledge there is no approach that can work for a complex environment including multiple environments. We will show that this will have a significant impact on the number and placement of APs.

- There is no approach that provides solutions for different cases of AP placement in a rail network. We consider three examples of such cases areas follows:

Case 1: Assuming a simple equally distributed placement (EDP) method for placement

of APs, we need to know the minimum required number of APs so that the average PL of the network does not exceed a defined threshold.

Case 2: A limited number of APs is available and we seek the optimal places to install those APs in the network.

Case 3: There is no preliminary constrain for the number and placement of APs and we need an algorithm to provide a total solution.

We address the above issues in Chapter 6.

2.5 Conclusion

In this chapter, we reviewed different areas of data transportation in rail networks included data classification and various train communications i.e. T2S, T2T and T2W methods. For the data classification, we realized that there is no appropriate approach to train the model in an online manner. This can cause high errors when the model is used in different trains. Regarding T2S communications, the existing works only used models that can only offload data during dwelling at stopping-station. This means that there is no approach for data offloading during movement e.g. at passing-stations.

For T2T communications, we found that there is a lack of a realistic model to determine the contact specifications between trains. This can cause unrealistic results that cannot actually work. Finally, for T2W communications, to the best of our knowledge, there is no single model for communication channel that condsiders all different environments existing in rail networks. This means, with the current models in the literature, we cannot find one model that can be used for all parts of a rail network and every part needs a different model.

We address the above gaps and propose appropriate solutions for every part through chapters 3-6 corresponding to data classification, T2S, T2T and T2W communication methods, respectively.

3 A Comprehensive Scheme for Data Classification and Transportation in Rail Networks

3.1 INTRODUCTION

The development of condition-based monitoring (CBM) systems in the Australian railway industry with over 33 thousand kilometers operational rail track has received the highest investment priority till 2040 [61], [62]. As will be described in the following, railway condition monitoring (RCM) systems will deal with of Big Data problem in the future because these systems meet the three aspects of variety, velocity and volume [63]. Future data-driven RCM systems will be strongly reliant on the huge amount of data received from heterogeneous IoT devices (variety) that are widely distributed throughout the rail network [4]-[65]. In order to have a sense of the volume of available data that needs to be processed, the authors in [5] have presented an example showing that the volume of collected data from only one vibration sensor of a train bearing with sampling rate of 25.6 KHz (velocity) will be as big as 10 Terabytes only for one train per eight hours of its operation. Consequently, the amount of gathered raw data from all sensors distributed throughout a long moving train can be as huge as several hundreds of Ter-

abytes (volume). Therefore, finding appropriate solutions for storage or transmission of such massive data will be necessary.

Currently, there are no cost-effective communication network to handle such massive data traffic [66]. Irrespective of future high capacity data transport methods such as 5G, we will still have the challenge of network coverage as railway networks are widely distributed all around the country and providing full network coverage for all areas especially for remote areas will be very costly and difficult [67]. Furthermore, assuming that a high capacity data transmission network is available to send all data to data centers, real-time processing of such huge unclassified data sent from all trains will not be efficient [68]. On the other hand, long-term storage of large scale data on-board with current storage technologies will be costly and affected by limitation in storage space [69].

In this chapter, we will propose a new framework for classification of Big IoT Data in rail networks. The proposed framework, which can be implemented in an on-board IoT gateway, will consist of two main parts: data analysis and data transmission. In data analysis part, raw data collected from heterogeneous on-board sensors will be analyzed and classified. Then, based on the classification of data, appropriate transmission solutions including real-time or delayed protocols will be applied for the transportation of collected data to control centers. In fact, the reality of having different transmission methods is one of the advantages of data classification prior to its transportation (edge processing). Performing data classification before its transmission will allow data centers to receive data in a classified manner instead of having a huge amount of unclassified raw data which will be extremely computationally expensive to process. This also causes a significant reduction in the capacity and bandwidth of real-time communication networks with railway centers as only critical data needs to be transmitted in real-time.

The main contributions of this chapter are as follows:

- We propose an on-board edge processing scheme that will handle the whole data management process including pre-processing, classification and multi-mode transmission of data in one integrated unit. This means that the proposed scheme will not only be able to classify the sensor data, but it will also provide the method of data transmission based on the classification of data.

- We develop a classification model that can be quickly re-trained in an online manner. Due to this ability, the developed classification algorithm can be quickly implemented in any train with different specifications.

- We propose a travel-pattern method for transmission of massive amount of non-critical data via existing WLAN at stations. This enables us to employ several existing rapid offloading technologies that approximately has no operational costs. Additionally, we will not need to install any road side unit (RSU) such as conventional V2I communications. This will also reduce the traffic of public communication networks such as cellular networks.

The rest of the chapter is organized as follows. In Section 3.2, we present an overview of our proposed architecture for on-board data management system. Then, we provide more detail for data classification as a main part of the proposed architecture and describe the related algorithm in Section 3.3. Issues about data transmission part and our proposed methodology to deal with those issues are discussed in Section 3.4. Then, in Section 3.5, we verify our developed classification algorithm through computer simulations . The conclusion is presented in Section 3.6.

3.2 PROPOSED ARCHITECTURE

Figure 3.1 shows our proposed architecture for classified transportation of large scale sensor data in rail networks.
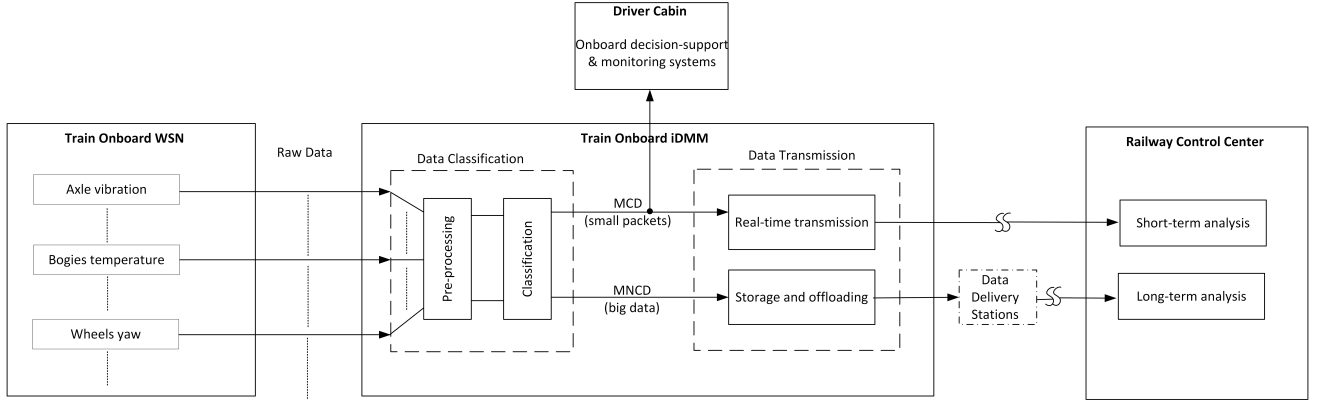
Figure 3.1: Proposed architecture

As illustrated in Figure 3.1, the proposed architecture is composed of two main parts including data analysis and data transmission. Continuously, heterogeneous streaming sensor data indicating several parameters of train equipment is received by an IoT data management module (iDMM). In on-board iDMM, firstly, all received data is processed and classified into two categories: MCD and MNCD respectively. Then, MCD (that can be a warning message with a few bytes length informing the faulty bearing) is sent immediately to the railway control center for fast decision making. While, MNCD, which contains the main volume of sensors data, is logged first and then offloaded at specific points such as train stations to be delivered later to data centers. Hence, based on the location of trains, we consider two functional modes in the proposed approach, as shown in Figure 3.2: Mode 1 is related to when trains are moving between stations and Mode 2 is pertinent to the time when trains are passing or stopping at stations. Classification and real-time transmission of critical data as well as logging of non-critical data are performed in Mode 1. While, rapid offloading of logged data is done in Mode 2. Detail about each part is described in the following sections.

20
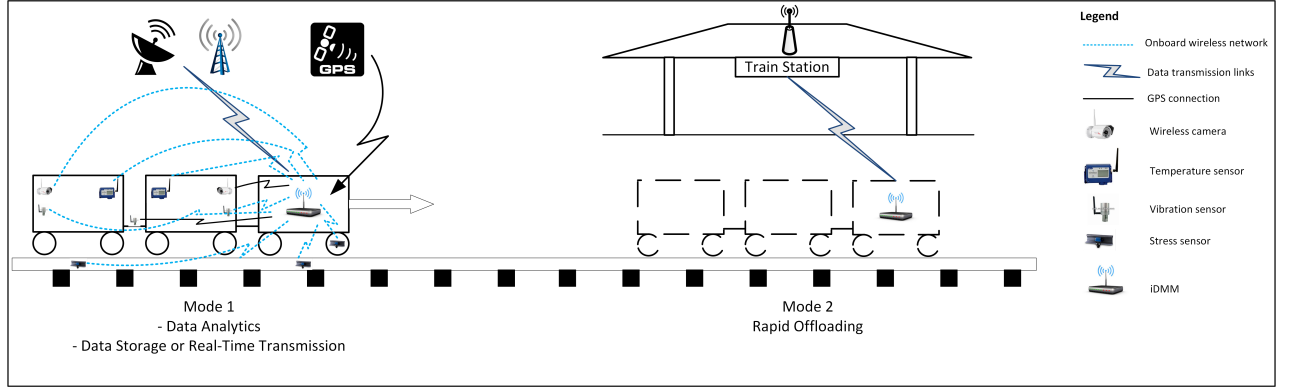
Figure 3.2: Functional modes of data management for trains

3.3 DATA CLASSIFICATION

Future trains will have many IoT devices to sense and send the conditions of different parts and equipment. In this section, we aim to recognize the parts and equipment with higher maintenance priorities, i.e. MCD, based on their transmitted sensors data. To do this, we need to find a data analysis method to enable efficient classification of data into MCD and MNCD. Generally, there are two main ways to perform classification on given data: signal analysis (SA)-based methods and machine learning (ML)-based approaches.

SA approaches include different strategies such as time-domain analysis, frequency-domain analysis or combination of both. A significant part of prior research has applied time and frequency domain analysis to extract some meaningful information from a set of collected data [70]. SA techniques are appropriate choices when we are looking for online analysis rather than offline analysis of data in batch mode because these methods do not need to be trained in essence. Real-time damage detection is an example of such methods [71]. However, for trains that are operated in variable conditions such as different speeds, environments or loads, pure SA methods cannot autonomously follow the changes and may need to be manually reconfigured after a condition change (non-adaptive methods) [72]. Otherwise, SA-based models may give false results after

a condition change. This is because SA-based approaches use pre-defined and fixed settings to detect faults [73]. Additionally, these methods are very sensitive to noise and unwanted samples and may be triggered with some unfiltered spikes [74] and [72].

ML-based approaches have two main methods including supervised and unsupervised methods. In supervised methods (classification), we must know different characteristics (labels) of a given system in various conditions (e.g. healthy or faulty conditions). However, in unsupervised techniques (clustering), our goal is actually to realize the different labels of a given dataset collected from a desired system based on the distribution or shape of data. The weakness of ML methods is to require extra time for training prior to be ready for predcition. However, ML methods unlike SA techniques are robust and autonomously adaptive and can be appropriate solutions for predictive fault detection applications. Given ML's strengths and robustness, we employ such methods for failure prediction of wheel bearings, which is one of the most important parts of a train. This will be explained in the following sections.

3.3.1 Feature Extraction

As discussed, we employ a ML algorithm for data classification. Generally, in ML methods regardless of supervised or unsupervised approaches, the first step is to perform techniques to extract meaningful features from sensor data. The selection of feature extraction (FE) technique is highly dependent on the type and distribution of data. For condition monitoring of a system or equipment, various types of signals or sensor data are used. Some examples of these signals can be vibration data, acoustic data, electrical data, visual data or environmental data [3]. To select the FE method, it is necessary to firstly know the types of signals or data that are available for our monitoring application as each data type requires a different FE method. In the current approach, our purpose is to monitor wheel bearings conditions as one of the most critical elements of a train
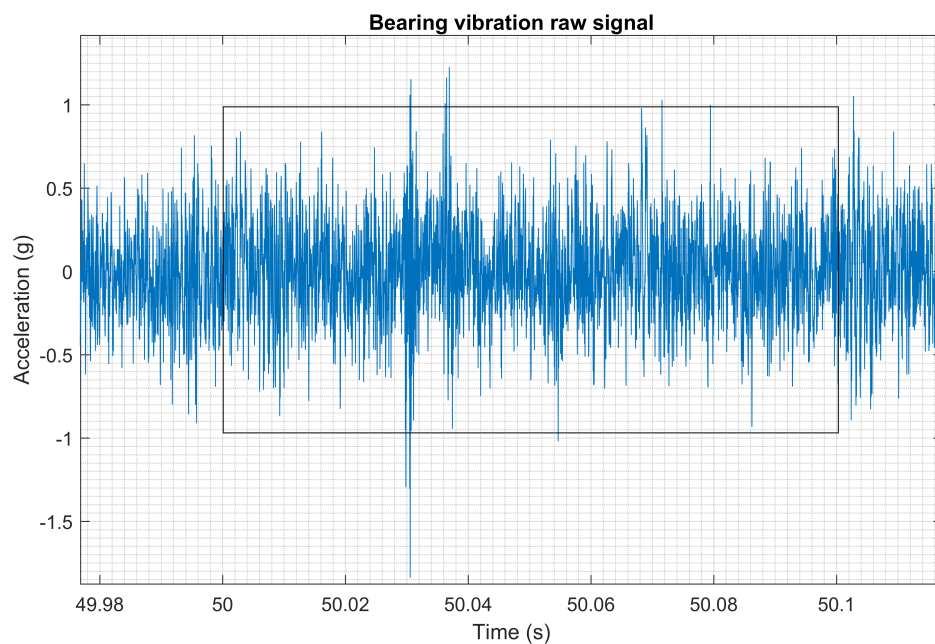
Figure 3.3: 0.1 second snapshot of a raw vibration signal in time-domain

where its failure can cause huge financial and human losses. To predict bearings conditions, there are several techniques [75]. Vibration analysis is one of the most common method that has received significant attention [75]-[77]. As an example, Figure 3.3 shows 100ms snapshot of a bearing vibration signal that has been sampled by a 25.6 kHz data acquisition system [78].

As Figure 3.3 shows, the raw vibration signal of a bearing in time-domain with much noise and spikes is not a proper signal to analyze and thus, the vibration signal is mostly considered in frequency-domain [70]. To convert a signal from time-domain to frequency-domain, Fourier Transform (FT) for continuous signals and Discrete Fourier Transform (DFT) for sampled data are mainly used [77] and [79]. As the vibration signals of current approach is sampled sensor data, we choose DFT for time-frequency conversion. For given data samples of x_n with N samples, DFT, is defined as follows:

$$X_k = \sum_{n=0}^{N-1} x_n . e^{-\frac{j2\pi}{N}kn} \ , \ n\epsilon\mathbb{Z} \tag{3.1}$$

The one thing that can be directly found from (3.1) is that DFT generates complex numbers and needs to extract the related absolute values for analysis purposes. Another useful transform for time-frequency conversion is Power Spectral Density (PSD). PSD unlike DFT, directly produces absolute values based on the following definition:

$$Y_k = \frac{1}{N} \sum_{n=0}^{N-1} |y_n|^2 \ , \ n\epsilon\mathbb{Z} \tag{3.2}$$

Furthermore, PSD gives more smooth features [80] (because PSD calculates the energy of signal) that can be easily seen in Figure 3.4. Additionally, compared to DFT, PSD features result in higher accuracy when used as inputs of classification models (which is the next step of our algorithm after FE) [81]. Hence, we choose PSD for FE.

After FE, we need a technique to find the most effective features. In this way, we can reduce the number of features by removing the redundant features that have very little effect on training of our desired classification model. For this purpose, we employ Principle Component Analysis (PCA) as one of the most common and powerful method [82]. Figure 3.5 illustrates how a few number of features (but the most effective ones) can approximate a whole set of data features. As can be seen from Figure 3.5, the first three number of principle components can cover around 98 percent of the whole data set that has 513 features in total. It means that for analysis of this data set, it is reasonably enough to consider only the first three components instead of the whole number.
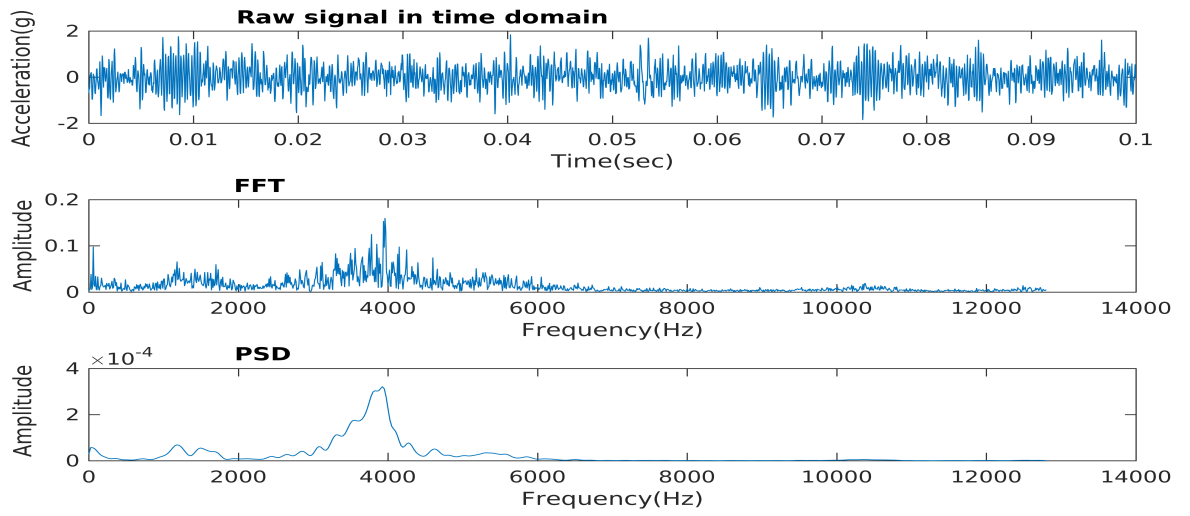
Figure 3.4: PSD vs. FFT
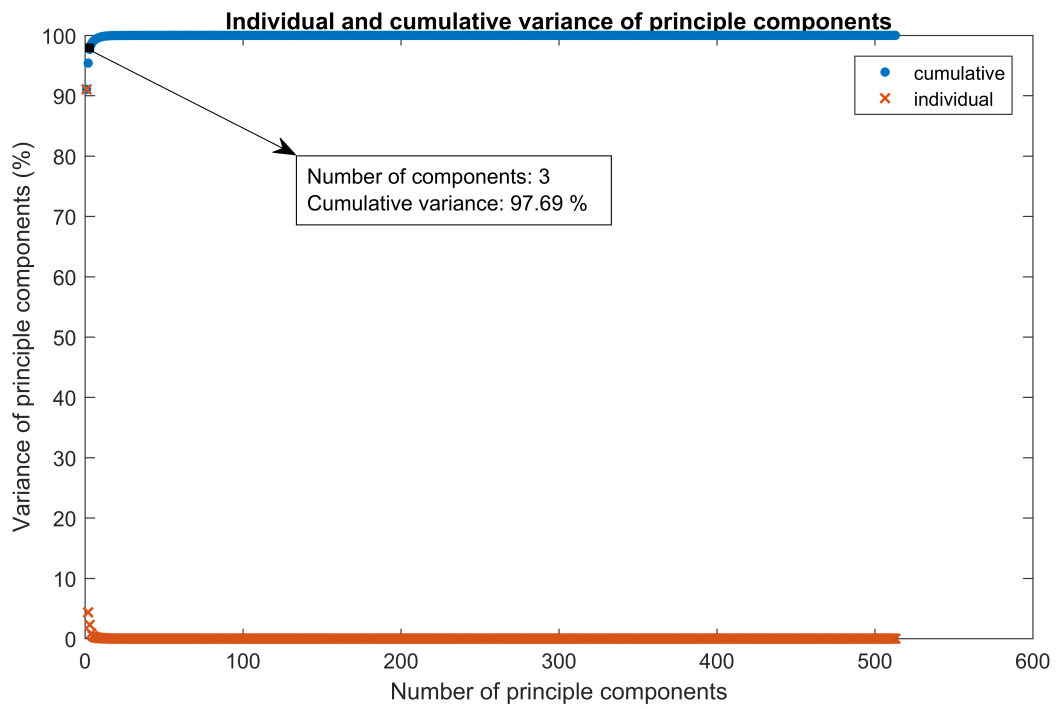
Figure 3.5: Number of features for approximation of the whole data set

3.3.2 Classification Algorithm

After feature extraction, we need to employ an ML method to train a classification model based on the most effective features that have been extracted in Section 3.1. To do so, we should select an appropriate algorithm for the desired classifier. As we do not have enough information about all failure types, we need a ML method that can be trained data during routine operations to detect anomalies. Therefore, we choose One-Class Support Vector Machine (SVM) that is one of the best solution for anomaly detection [83].

As long as the operating conditions of a train (e.g. its load or moving speed) does not experience a major change, the SVM algorithm will function effectively. Otherwise, the algorithm needs to be re-trained because the features of acceleration will change [84] and thus this will result in wrong predictions. Therefore, a pure SVM algorithm cannot cover all working conditions. Furthermore, SVM algorithm can only recognize the failures overall and is not able to realize the faulty part of a system. Given these issues, we developed an algorithm which can be autonomously re-trained upon change of operating conditions.

3.3.2.1 SVM Theory

Assuming X is a data vector with n number of samples, where $i = 1, 2, ..., n$. For a binary classification, we have two classes called positive and negative lebeled by $y_i = 1$ and $y_i = -1$, respectively. For a linear data, we can define a hyperplane f(x) to seperate the given data as follows [85]:

$$f(X) = W^T X + b = \sum_{j=1}^{n} w_j x_j + b = 0 \tag{3.3}$$

$$y_i f(X) = y_i(W^T X + b) \gg 1 \text{ for } i = 1, 2, .., n \tag{3.4}$$

In (3.3), W is an n-dimensional vector and b is a scalar and are used to determine the coordinates of the hyperplane. (3.4) is the constraint of the equation (3.3) and means $f(X) = 1$ if $y_i = 1$ and $f(X) = -1$ if $y_i = -1$.

Among several possible options, the hyperplane with the maximum margin is selected by the algorithm as the optimal hyperplane.

3.3.2.2 Algorithm Explanation

Assuming a given train with n bearings and m sensors per bearing, we apply FE for each sensor independently. Generally, for a set of n bearings and m sensors per bearing, a matrix of $X_{(m.n).w}$ can be formed as shown in (3.5). w is a selective quantity that shows the number of segments (or window length) to be chosen (we discuss the selection criteria of w in Section 3.5). In this expression, each x_{ij}^c is a time segment so that i, j and c indicate bearing number, segment number and channel (or sensor) number respectively. In (3.5), $i \in 1, ..., n$, $c \in 1, ..., m$ and $j \in 1, ..., w$.

Raw Data Matrix:

$$
X = \begin{bmatrix}
x_{11}^1 & & \cdots & & x_{1w}^1 \\
\vdots & & & & \vdots \\
x_{11}^m & & \ddots & & x_{1w}^m \\
\vdots & & & & \vdots \\
x_{i1}^c & \cdots & x_{ij}^c & \cdots & x_{nw}^c \\
\vdots & & & & \vdots \\
x_{n1}^1 & & \ddots & & x_{nw}^1 \\
\vdots & & & & \vdots \\
x_{n1}^m & & \cdots & & x_{nw}^m
\end{bmatrix}
\tag{3.5}
$$

Figure 3.6 shows the flowchart of our proposed algorithm for data classification. As illustrated in the Figure 3.6, raw sensor data is entered to the iDMM as inputs and matrix of raw data set is formed in Data Acquisition block. Then, features of each data set are extracted using PSD. If we had multiple channels per sensor, e.g. for measuring acceleration in different axes, then feature extraction would be performed on each channel of the sensors separately. (3.6) shows the feature matrices of a monitoring system with n sensors, where each sensor has m channels.

Feature Matrix for channel c:

$$
F^c = \begin{bmatrix}
f_{11}^c & \cdots & f_{1k}^c \\
\vdots & \ddots & \vdots \\
f_{nw1}^c & \cdots & f_{nwk}^c
\end{bmatrix}
\tag{3.6}
$$

After feature extraction, the algorithm determines the most effective features using PCA. For a multi-channel system, the algorithm applies PCA for each channel separately and then, re-applies PCA on the obtained matrices and forms a single matrix containing
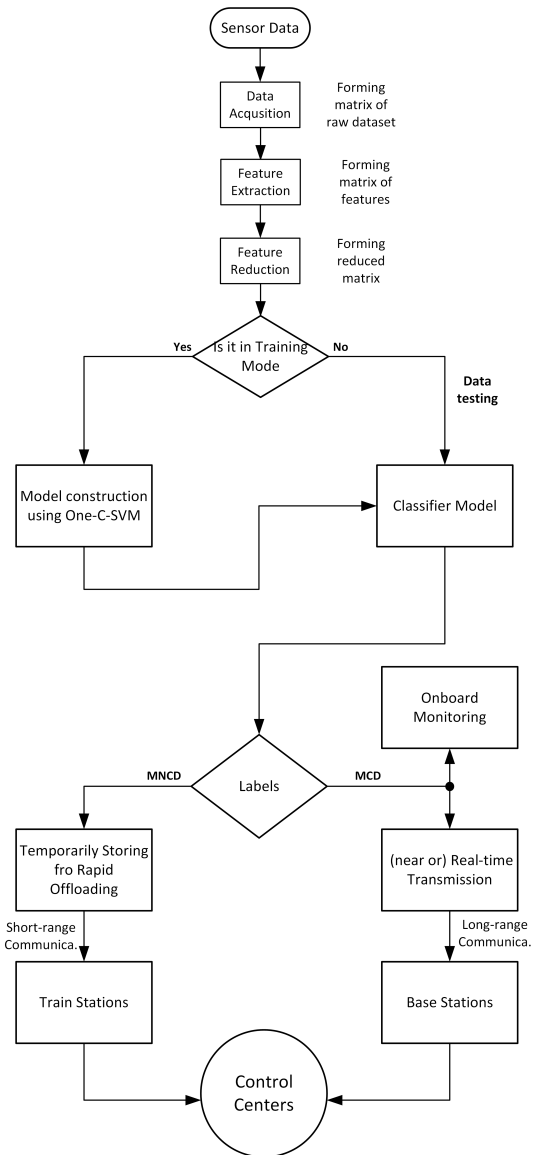
```
                    ┌──────────────┐
                    │  Sensor Data │
                    └──────────────┘
                           │
                    ┌──────────────┐         Forming
                    │    Data      │         matrix of
                    │  Acqusition  │         raw dataset
                    └──────────────┘
                           │
                    ┌──────────────┐         Forming
                    │   Feature    │         matrix of
                    │  Extraction  │         features
                    └──────────────┘
                           │
                    ┌──────────────┐         Forming
                    │   Feature    │         reduced
                    │  Reduction   │         matrix
                    └──────────────┘
                           │
              Yes      ◇ Is it in Training ◇      No
                          Mode
```

Figure 3.6: Classification algorithm flowchart

Figure 3.6: Classification algorithm flowchart

the most effective features among all channels. In this way, regardless of the number of sensors or channels in the system, the final result will be only a single matrix containing the most effective features. This makes the learning phase independent of the number of channels and the classifier will be trained or tested by only one matrix. The final matrix of the most effective features for the above mentioned system is shown in (3.7).

Matrix of effective features:

$$\hat{F} = \begin{bmatrix} \hat{f_{11}} & \cdots & \hat{f_{1l}} \\ \vdots & \ddots & \vdots \\ \hat{f_{nw1}} & \cdots & \hat{f_{nwl}} \end{bmatrix} \tag{3.7}$$

where $\hat{f}_{ij}$ indicates the most effective features. l is the number of the most effective features for all bearings.

This provides us with appropriate data for the training or testing of classifier model. For this purpose, we have defined a binary setting named Training Mode in the algorithm. For training the classifier model, this setting must be logically set '1' i.e. 'Training Mode = 1'. Otherwise, its default setting is zero, which means the algorithm is in Test Mode. Actually, for a given train which is operating in its nominal conditions (i.e. speed and load), the classifier model needs to be trained once only. There are some transients until a train reaches its nominal speed. To prevent the classifier from being trained within transient times, the algorithm can use train speed as an input signal to block the training process during accelerating or braking times. This input blocks the training process until train reaches its nominal speed. Then, the classifier is trained by coming sensor samples during a short time period (e.g. 200 seconds in our experiments). After training process, the mode of algorithm is changed to Test Mode and will be ready to predict the labels of new coming sensor data.

From the obtained labels, we can realize the anomalies or MCD of each group of samples (i.e. segments). It is also possible to extract some statistics for each data set (such as the percentage of anomalies relating to each bearing). This can be a very useful criteria to realize which bearing is not in normal condition and should be serviced soon.

3.4 DATA TRANSPORTATION

3.4.1 STORAGE AND OFFLOADING

The proposed on-board data classifier enables us to determine the non-critical portion of data stream (i.e. MNCD, which is not used to make any immediate operational decisions). Therefore, it is not necessary to employ a costly form of long-range communication for transmission of MNCD from moving trains to grounded centers. Instead, it can be temporarily stored in some on-board storage units and be delivered via a delayed offloading process. Our proposed locations of data offloading are train stations, as they are currently existing infrastructure and available all around the train's paths (except regional and country services that can also be quickly implemented with a few cost) and thus there is no need to construct any new dedicated platforms. Additionally, the challenge of transmission of big data will be technically more feasible as it will change the infrastructure-based methods of long-term mobile communications (which have very high costs as well as low QoS specially for rural areas). This is due to recent advances in technologies that can be used for rapid offloading, the possibility of substitution of long-range communications with short-range wireless standards such as IEEE 802.11ay. This new communication standard that will support data rates of up to 100 GB/s, can be an effective solution for our proposed offloading scheme [86].

In our proposed offloading scheme, the incoming IoT data of a train will be temporarily stored in an on-board storage unit (OSU) on the train. Then the stored data will be

delivered to the next reaching stations. As stopping times of trains at stations and also data offloading rates are both limited, the offloading process may not be completed in one station. In this case, the rest of stored data will be offloaded in the next reaching station(s). As a matter of solution, we can divide the OSU into smaller units (SU) as follows:

$$V_0 = R \times T_0 \tag{3.8}$$

where V_0 is the volume of one SU and R is the data rate of the available transmission protocol at the offloading station. As trains stop more at main stations, different stopping times are expected in a rail network. Of these, T_0 is selected as the minimum stopping time in a given rail network. This guarantees the full offload of at least one SU onto every station.

For example, in a rail network with minimum stopping time of $T_0 = 10s$, if we use a wireless protocol (like WiFi) with a given data rate of $R = 500Mbps$ for example, the volume of each SU, V_0, will be 625 MB. In this case, to store 750 MB for example, we will need two SU's (including one full SU and one with 125 MB stored data). This means that only one SU with 625MB can be offloaded in the stations with 10s stopping times and the next SU containing 125MB can be exchanged in the next reaching station(s). The more data is stored in an OSU, the offloading process will need more stations to be completed. However, with the advent of new coming ultra wide-band (UWB) technologies like IEEE 802.11ay, more and more data with shorter delays can be offloaded onto every station [86].

3.4.2 REAL-TIME TRANSMISSION

As a benefit of using on-board data classification, it is no longer needed to send the whole data in real-time and therefore, current cellular networks can be used for real-time transmission of critical data (which has small size) without experiencing high traffic.. The most recent communication protocol developed for railway applications is LTE-R (Long Term Evolution-Railway), which is the alternative option for the current GSM-R [6]. 5G(which is expected to be available after 2020) can be also the next possible solution for railway communication networks [6]. However, the cost of communication will be significant when we want to apply it for transmission of large scale data. To have an estimation of such huge data, authors in paper [5] have presented an example as follows.

With a wireless sensor network which sends measured data with sampling rate of 25.2 ksamples/sec and 12 B per sample (just for axle bearing vibrations), the amount of data collected in a train working 8 hours a day, is around 9 Terabytes:

$$25200\,(samples/sec) \times 12(B/sample) \times 8hours \times 3600sec \approx 9TB \tag{3.9}$$

Regardless of the feasibility of transmission of such amount of data in real-time (even if with the advent of new technologies such as 5G, high volume data transmission will be possible), the transmission cost will be huge [87]. However, by applying our proposed data classification, we can significantly reduce the amount of data that must be transmitted in real-time (or near real-time).

Table 3.1: Specifications of selected data sets

Data Set	Specifications
1	IMS Data Sets: • Sampling frequency: 20KHZ • Time segments: one-second snapshot • Number of time samples per segment: 20,480 • At the end of the run-to-failure experiment, inner race defect occurred in **bearing 3** and roller element defect in **bearing 4**.
2	IMS Data Sets: • Sampling frequency: 20KHZ • Time segments: one-second snapshot • Number of time samples per segment: 20,480 • At the end of the run-to-failure experiment, outer race failure occurred in **bearing 1**.
3	FEMTO Data Sets: • Sampling frequency: 25.6 kHz • Time segments: 0.1-second snapshots • Time samples per segment: 2560 • At the end of the run-to-failure experiment, an unknown failure occurred in **bearing 2**.
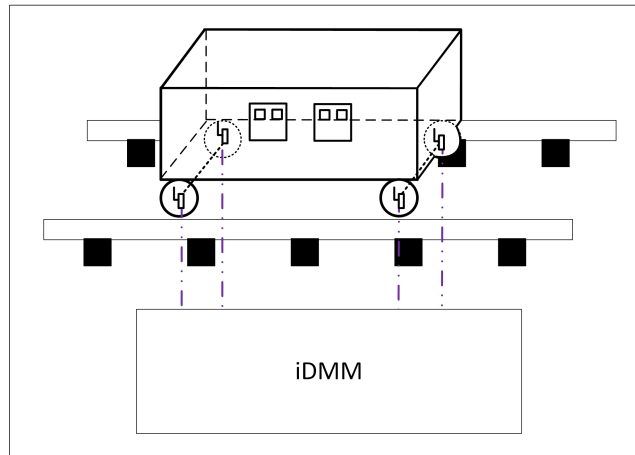
Figure 3.7: A single wagon with four bearings for our experiments
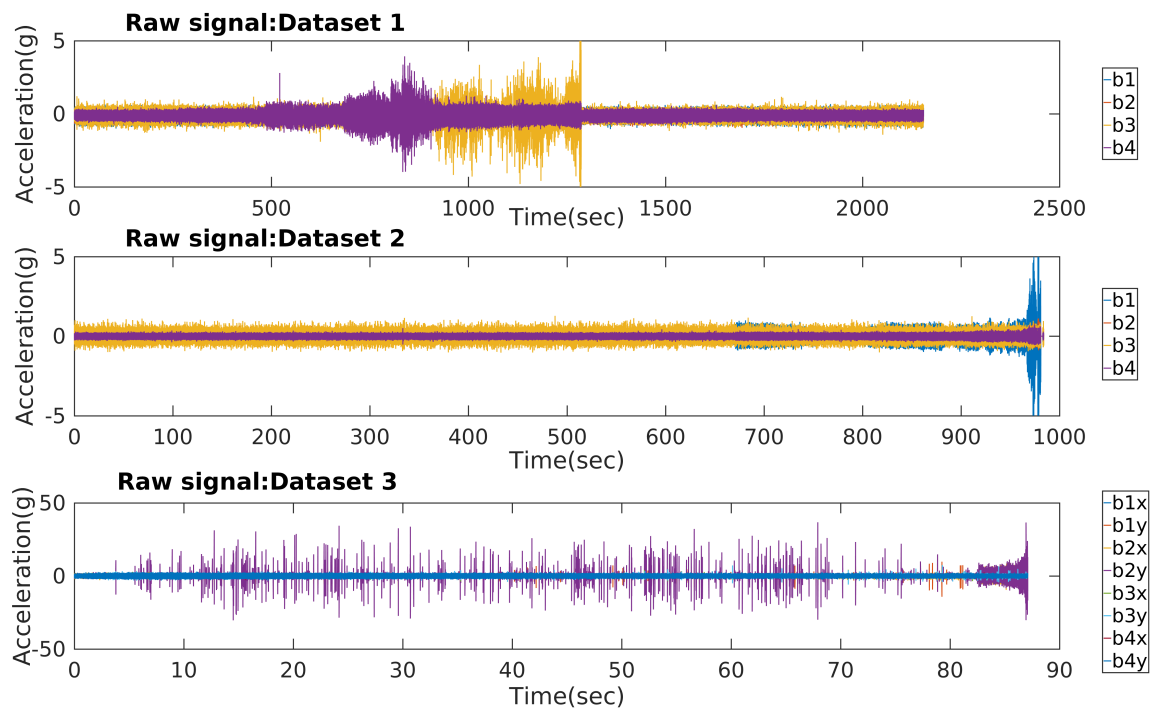
Figure 3.8: Run-to-failure data

Table 3.2: The results obtained from our data classifier algorithm. The numbers in red color and bold format are related to the faulty bearings.

Data Set	Selected Windows (Segments)	Anomalies (%)				Training Time (s)	Testing Time (s)
		b_1	b_2	b_3	b_4		
DS1	windows selected long time prior to failure of bearings.	0	0	1.5	1	36.725	36.987
	windows selected close to bearings failure.	0	1	**6**	**16.5**		
DS2	windows selected long time prior to failure of bearings.	0	0	1	0	37.227	36.680
	windows selected close to bearings failure.	**7**	0.5	0.5	0		
DS3	windows selected long time prior to failure of bearings.	0	3	0.5	0.5	11.671	11.491
	windows selected close to bearings failure.	0	**45**	1.5	1		
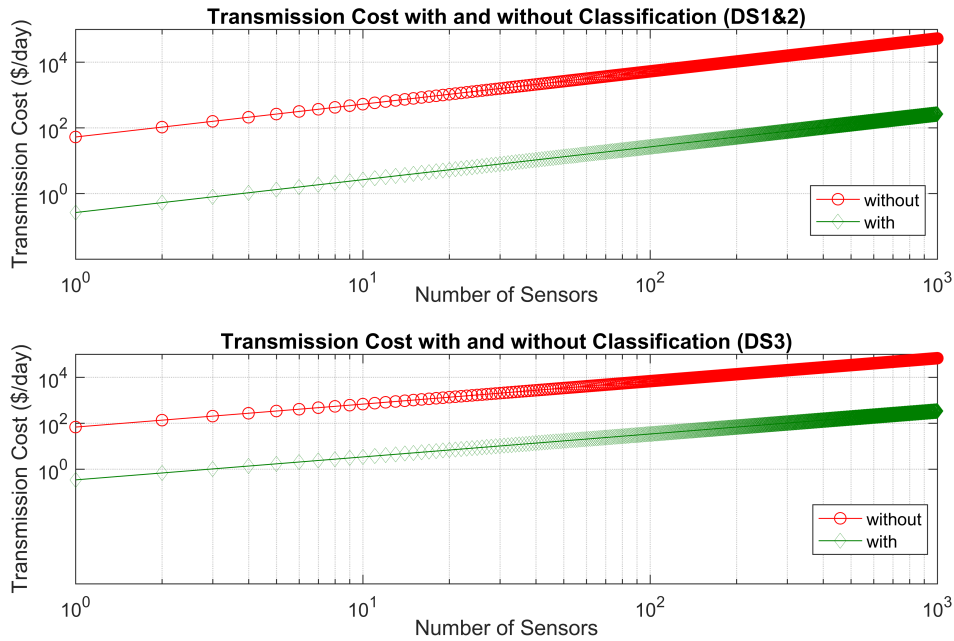
Figure 3.9: Data transmission costs with and without classification algorithm

3.5 EXPERIMENTAL VERIFICATION

In this section, we evaluate the performance of our designed classifier with some sets of real experimental data. We used three different datasets to show the functionality of the algorithm in various conditions. The Data Sets 1 (DS1) and 2 are from [88] and DS3 is from [78]. Table 3.1 shows the specifications of each DS. Each data set contains run-to-failure acceleration samples coming from 4 sensors relating to four bearings. We choose a single wagon with four bearings for our experiments (Figure 3.7). However, it does not matter how many bearings we have for a real train and our algorithm can be applied to a whole train with several wagons. The sensors of DS1 and DS2 are both single-channel but the sensors of DS3 is dual-channel to measure acceleration in two different axes of x and y. Data sets 1 and 2 are selected to verify that our algorithm is independent of the types of bearings failures. As Table 3.1 describes in DS1, we have inner race and roller element defects for bearings 3 and 4 receptively. However, in DS2, there is different type of outer race failure. Additionally, DS3 has been chosen to show that our algorithm is independent of types of bearing failure. As well as the mentioned differences between selected data sets, the specifications of sensor network in DS3 (i.e. types of sensors and sampling frequency) are also different from DS1 and DS2 as described in Table 3.1. Figure 3.8 shows the raw signals of run-to-failure acceleration for the above mentioned data sets.

Feature Extraction: As Table 3.1 shows, number of samples per each time segment is 20480 and 2560 for DS1(2) and DS3, respectively. By applying PSD with similar settings for each data set, we will obtain the same number of 513 features in frequency domain for all three data sets. Then, by separately applying PCA for each DS, different results are obtained as illustrated in Figure 3.10 where DS3x and DS3y are related to the channels x and y of DS3, respectively. According to this, while for DS1 and DS3y, we need only three principle components to cover over 98 percent of data, for DS2 and DS3x,
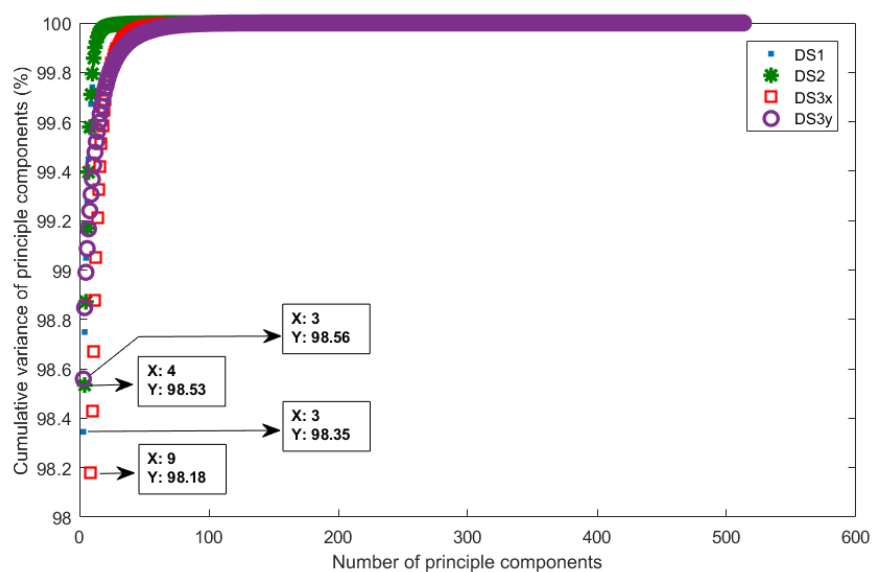
Figure 3.10: Results of PCA for each dataset

we have to use four and nine principle components to achieve the same data coverage, respectively.

In order to know how principle components (PC) changes during a run-to-failure experiment, we have illustrated the scatter diagram of DS1 in Figure 3.11. DS1 is an appropriate choice for graphical representation of PCA results because with requiring only three effective PCs for data coverage, it can be easily illustrated in a 3-dimensional (3D) diagram (in comparison with the two other DSs that needs more than three PCs for representation). Additionally, DS1 as a dataset that is related to the worst case experiment with two faulty bearings (compared with two other experiments with only one bearing failure), can give more information about PCs changes. As illustrated in Figure 3.11, the scatter diagram shows four separated clusters that are related to the four bearings in the experiment. Two clusters show more anomalies addressing the faulty bearings that are $b3$ and $b4$ based on Table 3.1. Between these two latter clusters, the one with more variations is related to $b4$ that compared to $b3$, its failure starts earlier

(as can be seen in Figure 3.8) and with different type of defect (as mentioned in Table 3.1). These results are also matched with the percentage of anomalies shown in Table 3.2 (that will be described in detail in the following paragraphs).
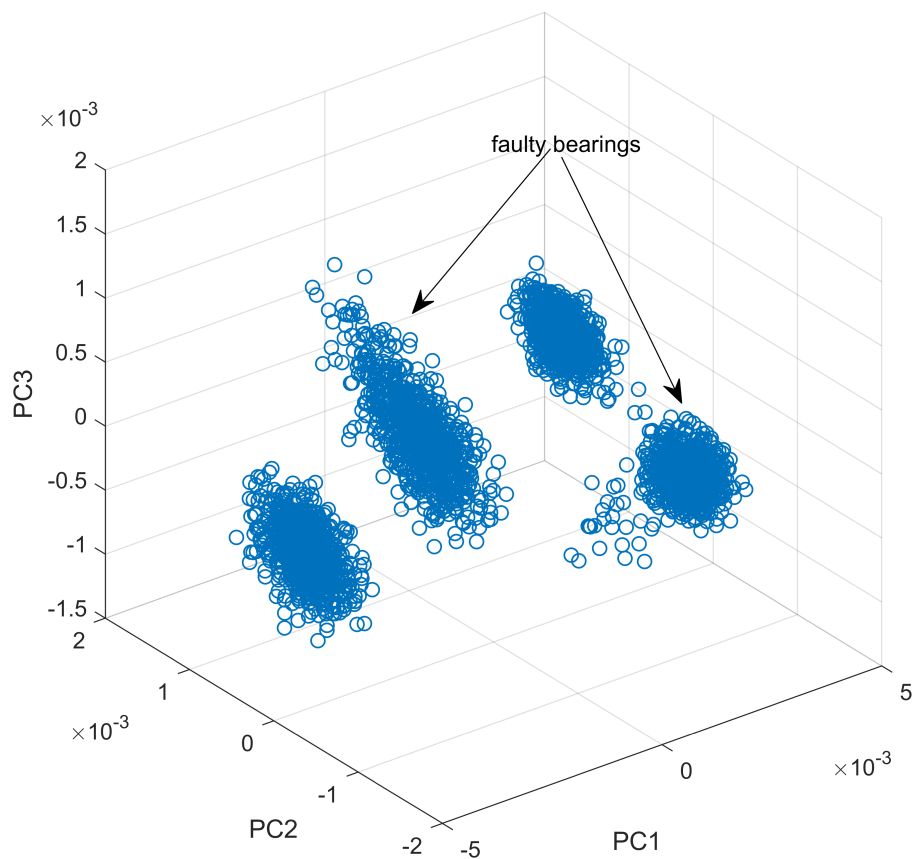
Figure 3.11: Scatter diagram of the obtained principle components for DS1

Window Length Selection: According to our window-based algorithm described in Section 3.3, we need to select an appropriate window length. The selected window length will determine the delay of our classification algorithm. Compared to the long degradation time of bearings that is actually around several hundreds of hours [88], having a delay around a few minutes, e.g. less than 5 minutes, are reasonable for our near-real-

time algorithm. To guaranty a delay of less than 5 minutes, we choose a maximum delay of 4 minutes, for example, for our proposed algorithm. This 4-minutes delay will result in a window length of $w = 200sec$ for DS1 and DS2:

the total delay for DS1 and DS2,

$$d_{tot-D1\&2} = d_{col-D1\&2} + d_{alg-D1\&2} \leq 240s \tag{3.10}$$

where d_{col} is the data collection time and d_{alg} is the training/testing time that algorithm needs for model training or label predicting and is mentioned in Table 3.2. As $d_{alg-D1\&2} < 40s$ for DS1 and DS2 based on Table 3.2,

$$d_{col-D1\&2} \leq 240s - 40s = 200s \tag{3.11}$$

As time segments are one-second snapshots in DS1 and DS2, the obtained collection delay will cause window length of 200. Similarly, we can choose a window length of 200 for DS3 provided that the total resulted delay will not exceed 4 minutes. By applying similar calculations for DS3 with $w = 200$ and 0.1-second snapshots, the total delay will be less than 40 seconds that will strongly confirm the condition of $d_{tot} < 4minutes$:

$$d_{tot-D3} = d_{col-D3} + d_{alg-D3} \leq 40s \tag{3.12}$$

Table 3.2 shows the simulation results for each DS, assuming that we have implemented our developed classification algorithm in an on-board gateway of a train. As each train has differences in bearing types, sensor network specifications (e.g. sampling frequencies and data acquisition intervals) and operating conditions (such as speeds and loads),

the bearing vibration profiles are different for each train [78] and [89]. Hence, the classification algorithm must be trained when used for the first time in a train to make a model specific to that train. After the learning process is over for a train, the algorithm is automatically changed from training mode to test mode. To recognize the time duration of mode conversion from training to test, we use a timer set by d_{tot} that can be obtained from (3.10) and (3.12).

As can be seen in Table 3.2, we have presented the percentages of anomalies in two different time windows of bearings vibration samples for each DS. Since all the data sets are samples of run-to-failure experiments, we have selected a window from the initial samples of each DS as the samples related to the bearings in normal conditions. This can also be found from the time diagrams illustrated in Figure 3.8. Similarly, we have chosen another window from the samples close to the failure of a bearing based on the time diagrams of bearings vibration as the samples of failing bearing (Figure 3.8). As can be seen, the percentages of anomalies relating to failing bearings (numbers with red colors) have tangibly increased in comparison with the other bearings with no fault. Since these samples are all related to the time before bearing failures occurred, we can predict the bearings that are close to fail from the increase of these percentages.

Comparing the results of the Table 3.2 with the time curves of Figure 3.8 also confirms the correctness of the classification algorithm in the early detection of the failing bearings. For example, the time diagram of DS1 in Figure 3.8 shows that firstly, bearing 4 (diagram with purple color) and then bearing 3 (diagram with yellow color) will start to be failed and bearings 1 and 2 remain safe. If we check the obtained anomalies for DS1 in Table 3.2, the percentages of anomalies for bearings 3 and 4 have experienced big increases. As the samples have been chosen from the same time periods when the bearing 4 is in its worst situation of failure, the change of anomalies in bearing 4 is bigger than bearing 3. This shows the failure of bearing 4 has been started earlier than bearing 3 which
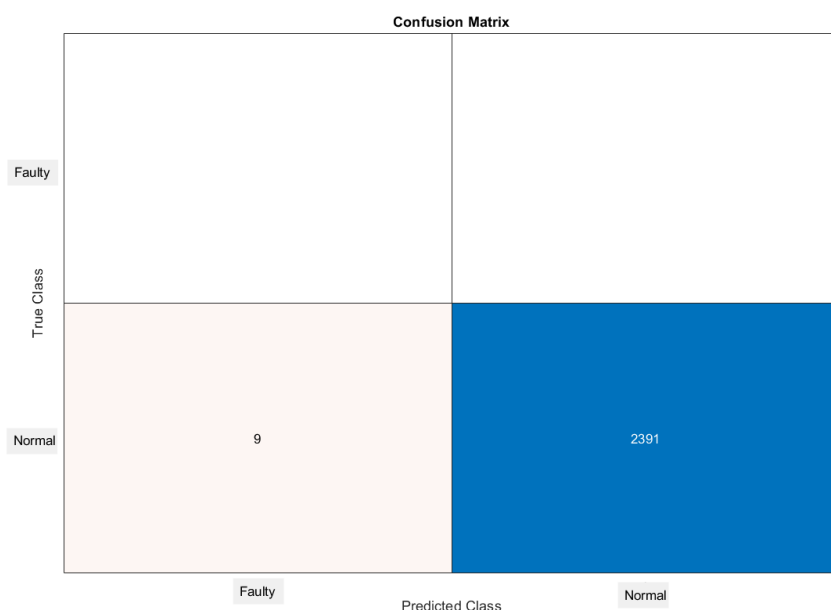
Figure 3.12: Classifier confusion matrix

is matched with the mentioned time diagrams. Additionally, Figure 3.12 shows the confusion matrix of the classifier for DS1. As illustrated, the error of the classification model is 9 out of 2400, which is less than 1%.

There are also two time delays for each experiment in Table 3.2 including times of training and testing; the training time is the time that algorithm needs to build a classification model appropriate for a given DS; and the testing time is the time that algorithm requires to classify and determine the labels of new coming samples. These time delays are based on the specifications of the computer system that we used for our simulations and are shown in Table 3.3.

In addition to the advantage of early failure prediction, our proposed algorithm will also have a significant reduction in cost of real-time communications, as previously discussed in Section 2.3. Figure 3.9 shows the transmission cost with and without classification for different numbers of sensors. As illustrated in Figure 3.9, e.g. for a given train with a

Table 3.3: System specifications for simulation process

Operating System	Hardware
Red Hat Enterprise Linux Workstation, Release 6.7 Kernel Linux 2.6.32-573.12.1.el6.x86_64	Memory: 32 GB CPU: Intel(R) Xeon(R) CPU E5-2690 0 @ 2.90 GHz

hundred sensors, the transmission cost is around ten thousand dollar a day for a system with no classification. However, by using the proposed classification algorithm, the cost will be significantly reduced by 99 percent to less than a hundred dollar.

The prices of Figure 3.9 are according to Telstra's (a leading telecommunications company in Australia) data rates for 4G networks [90] and the related data amounts are calculated based on (3.9). The differences transmission costs between systems with and without classification arise from this principle that in a system with classification, we only need to transmit the percentage of anomalies per each selected window instead of sending all data.

3.6 CONCLUSION

In this chapter, we proposed an on-board data classifier using a mixture of signal processing coupled with ML techniques in order to limit the transmission bandwidth on public mobile network (where it is uneconomical to transmit huge amount of data). We also suggested transmission management plan where data classified as non-critical are logged and then offloaded at specific points (such as train stations). The application was designed for railway condition monitoring and results were presented. The results showed that our algorithm can correctly classify data samples from three various data sets with different specifications.

4 Train-to-Station Communication Method

4.1 Introduction

Future trains will be equipped with many sensors that continuously sense and generate massive IoT (internet of things) data [5]. According to [5], the amount of sensor data produced by only one sensor for sensing the vibration of just one wheel bearing in a train will be as huge as 10 TB during eight operating hours. Thus, for a train with many parts that will be sensed by wide variety of sensors [3], the created data amount will be extremely massive and transmission of such data into data centers will be a challenge.

Based on the risks for passengers and rail equipment, the collected sensor data is classified into two classes comprising of critical data and non-critical data. The critical data can cause serious damages for both people or rolling stocks and should be declared immediately. However, the non-critical data is used for long time analysis and can be evaluated by delay. As trains operate in normal conditions for most of the time, the amount of critical data is tiny and the main part of sensor data is composed of non-critical data. Therefore, if we could classify the sensor data through an appropriate edge processing task [8], we will be able to employ different communication strategies to transfer critical and non-critical data to railway data centers. In this case, it is feasible to send critical data (which contains a small portion of data) in a real-time manner (e.g. via cellular networks) while temporarily store the non-critical data and deliver it later

via an appropriate offloading strategy [91]. In this way, we will significantly reduce the data traffic over expensive and infrastructure-based communication networks (such as cellular or satellite networks) by offloading the massive part of data through an available cheap WLAN's channel such as WiFi[1] networks at stations.

This is the idea behind this chapter which based on that, we propose train stations as potential spots to offload the delay-tolerant non-critical sensor data. In this way, stations has the feasibility to provide more powerful computation and communication capabilities for our offloading task. Additionally, if we employ the available channels of WLAN in stations, this will cause large cost saving because we will no longer need to install any extra infrastructure. The proposed offloading method will be a train-to-station (T2S) communication between on-board units (OBU) in trains and a data sink system in stations.

Therefore, the main contributions of this chapter is:

- we propose a novel scheme for offloading of delay-tolerant part of IoT data in rail networks,

- we develop an analytical model for the proposed offloading scheme that can model the data offloading task for passing stations as well as stopping stations,

- we provide an integrated equation that can estimate the total offloading capacity for a given train during its trip between two terminals including stopping and passing stations

- we embedded used a rate control algorithm that enables the data to be offloaded even with the minimum WiFi signal power. This makes a significant increase in

---

[1]Since, in this chapter, we use IEEE802.11-based networks as the required WLAN, we sometimes use the term "WiFi" instead of "WLAN".
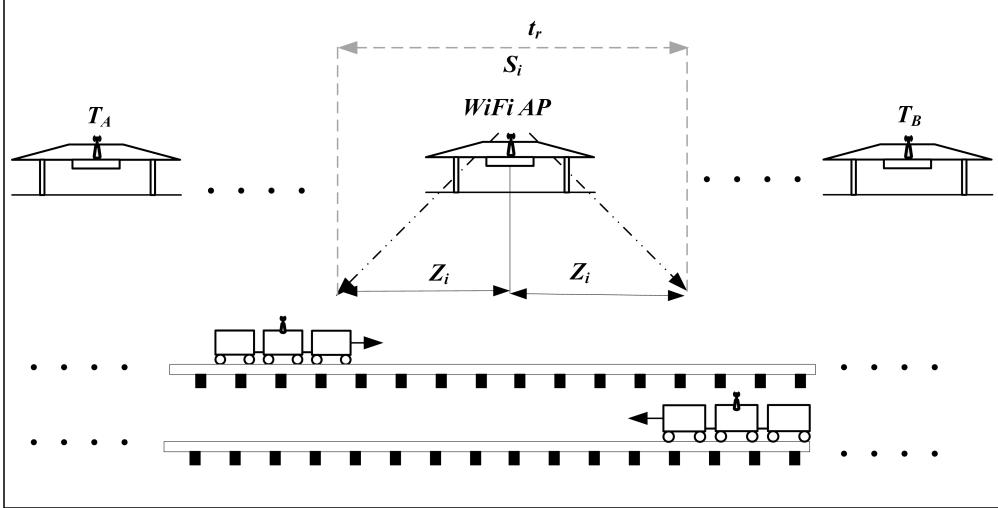
Figure 4.1: Overall diagram of the proposed station-based offloading scenario

the size of data offloading (which equals to the size of data can be offloaded at passing-stations).

The rest of the chapter is organized as follows. We explain our proposed offloading scheme in Section 4.2. In Section 4.3, we develop an analytical model for the proposed offloading task. The simulation results will be presented in Section 4.4 and we conclude the chapter in Section 4.5.

4.2 The Proposed Offloading Scheme

Figure 4.1 illustrates the overall diagram of our proposed station-based offloading scenario. In order to obtain an offloading model that can estimate the offloading capacity of each station, we need to estimate the two following parameters:

1. WiFi contact duration, which is the time of presence of a train inside of WiFi communication zone of a given station, and

2. feasible data throughput of offloading session during such WiFi contact duration

For the short range WiFi networks, the contact duration is limited to the duration that trains are sufficiently close to the stations. The main opportunity that can be considered as the contact duration is trains dwell times for passenger exchange at stations. To increase the efficiency of our offloading task, we add three more time slots for the contact duration. These time slots include when a train is close enough to a station during entering, leaving or passing such station. In fact, our target is to start each offloading session as soon as a train reaches the WLAN communication zone (i.e. upon detecting strong enough beacons from WiFi AP's at the stations).

The dwell time of a train at each station is not constant and varies between a lower and upper bounds [92]. The lower bound, which for each type of train with a given number of doors is the minimum time required for opening and closing its all doors, is a definite quantity. However, the upper bound, which is required for safely exchanging passengers, is variable based on several parameters such as station type, train specifications and hour/day of operation (e.g. peak or off-peak times and weekdays or weekends) [92]. The other time slots are also variable and depend on the speed of train when entering, leaving or passing through a station. Although these time slots are not so long based on the short range of WiFi networks, the amount of offloaded data will be significant thanks to the emerge of rapid offloading protocols such as existing IEEE802.11ac as well as the other new upcoming multi-Gigabit/s standards like IEEE802.11ay [86].

As described, we integrate a rate control scheme into our model that enables the offloading task to be started even with minimum available levels of WiFi signal powers. Generally, there are two types of rate control schemes: physical (PHY) layer based schemes that control the data rate based on the parameters of physical layer such as received signal strength (RSS), e.g. Receiver Based Auto Rate (RBAR) algorithm [93]; the algorithms that work based on network layer parameters such as packet delivery ratio (PDR), e.g. Adaptive Auto Rate Fallback (AARF) [93]. Assuming an RBAR scheme,

the high data rates of a wireless channel can be theoretically determined based on the mapping tables in the related standards such as what illustrated in IEEE802.11 series [94]. These tables maps the minimum levels of received signal-to-noise ratio (SNR) of a radio signal to a modulation and coding scheme (MCS) index. Each MCS can provide up to a definite data rate based on the carrier frequency, the available channel bandwidth and the number of spatial streams. The level of SNR is estimated based on the level of noise and the received signal strength (RSS). RSS is also estimated based on the transmission power, the distance from AP, environmental (obstacles) and weather (temperature, humidity, etc.) conditions [95] and [96]. Data throughput is theoretically a percentage of maximum data rate called MAC efficiency [97]. The actual amount of data throughput is determined via in-field measurements.

4.3 The Analytical Offloading Model

As explained in Section 4.2, to build the proposed offloading model, we need to find the WiFi contact duration and throughput at each station. Generally, a train will not stop at all through stations between terminals and for some stations, it will only have a short passing. To obtain the maximum efficiency in the offloading process, we consider both types of stations including stopping stations (where a train stops for passenger exchange) and passing stations (where a train just passes with no stop).

For simplicity, in the all following equations, we assume that at each station, a dedicated WiFi network has been allocated only for the task of data offloading and only one train will be permissible to offload data at each session. We also suppose equal speeds and accelerations for a given train during entering and leaving a station. Additionally, we do not affect the location of antennas assuming that there are enough number of antennas for data offloading, e.g. in the first, middle and end parts of trains and station platforms.
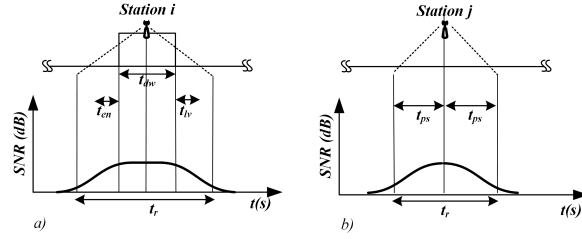
Figure 4.2: Timing diagrams of the offloading model for a) stopping stations, b) passing stations

4.3.1 Offloading Model for Stopping Stations

Figure 4.2 shows the timing diagrams of the offloading model for both stopping and passing stations. Based on this, the related WiFi contact duration, t_r, for stopping stations is obtained as follows:

$$t_r^{stp} = t_{en} + t_{dw} + t_{lv} \tag{4.1}$$

where t_{en}, t_{dw} and t_{lv} are entering, dwelling and leaving times of a train at a given station, respectively.

Assuming that a train enters (leaves) the communication zone of a given station at a given speed and gradually decreases (increases) its speed with an acceleration of a until it stops. If suppose a train enters or leaves a station with equal speed and acceleration, then for a maximum communication zone of d_{max} for the WiFi network at that station, entering (leaving) time, $t_{en(lv)}$, is simply obtained:

$$d = \frac{1}{2}at^2 + v_0 t + d_0 \tag{4.2}$$

where d_0 is the distance between transmitter and receiver during dwelling time for passenger exchange, and d is the distance after t second of leaving the station from the

transmitter. For leaving scenario, $v_0 = 0$, as a train starts leaving a station from standstill situation:

$$t_{en} = t_{lv} = \sqrt{\frac{2d_{max}}{a}} \tag{4.3}$$

For calculation of data throughput, we firstly need to estimate the received signal strength (RSS) at distance d from a WiFi access point (AP) using log-normal shadowing path loss model as follows [98]:

$$rss(d) = P_{ref} - 10\gamma log(d/d_{ref}) + X_\sigma \tag{4.4}$$

where P_{ref} is the received power at reference distance d_{ref} and , γ is the path loss component (PLE), and X_σ is the normally distributed random variable with zero mean and σ standard deviation (SD). γ and σ reflect the environmental conditions and are two and zero for free space, respectively.

P_{ref} can be theoretically obtained by $P_{ref} = P_t - 20log(\frac{4\pi d_{ref}}{\lambda})$ in dBm (supposing free space environment), where P_t is the transmitter power and $\lambda = c/f$ (c is the light speed and f is the radio carrier frequency).

By substituting d from (4.2) to (4.4) and considering $d_{ref} = 1m$, rss based on time is obtained as follows:

$$rss(t) = P_{ref} - 10\gamma log(\frac{at^2}{2} + d_0) + X_\sigma \tag{4.5}$$

According to IEEE802.11ac Standard [94], the maximum bit rate of a WiFi physical link (PHY) is estimated based on the level of signal to noise ratio (SNR). SNR can be calculated as follows:

$$snr^{dBm} = rss^{dBm} - n^{dBm} \tag{4.6}$$

where n^{dBm}, is the background noise level based on dBm at receiver. From (4.5) and (4.6), SNR based on time can be obtained:

$$snr(t) = P_{ref} - n^{dBm} - 10\gamma log(\frac{at^2}{2} + d_0) + X_\sigma \tag{4.7}$$

The maximum bit rate of WiFi PHY link is obtained from MCS mapping tables based on available channel bandwidth, bw, number of spatial streams, N_{ss}, and duration of guard interval (GI) as follows:

$$\forall snr^i_{min} \leq snr(t) < snr^{i+1}_{min} \xrightarrow{\mathcal{F}} bitrate_{max} = r^i \tag{4.8}$$

where $\mathcal{F}$ is the mapping function that maps every minimum snr to a defined bit rate based on IEEE802.11ac Standard, and $i = \{0, 1, 2, ..., 9\}$ representing the MCS indexes in IEEE802.11ac Standard. r^i is the maximum bit rate that can be reached based on level of snr.

Assuming a MAC efficiency of ρ, the throughput during moving is simply obtained from the following equation, where $th_{en(lv)}$ is the throughput within entering or leaving the station and r is the maximum data rate obtained in (4.8).

$$th_{en(lv)} = \rho r \tag{4.9}$$

Therefore, the offloading equation for stopping stations is obtained as follows:

$$A_{stopping} = th_{dw}.t_{dw} + 2 \sum_{t=0}^{t_{en}} th_{en}.\Delta t \qquad (4.10)$$

In (4.10), th_{dw} is the maximum throughput during dwelling and Δt is the time resolution for calculating snr.

4.3.2 Model for Passing Stations

For non-stopping stations, the related equation is simpler than previous section, as the train only passes the station with a constant speed (i.e. $a = 0$) with no stop:

$$t_r^{ps} = 2t_{ps} \qquad (4.11)$$

where $t_{ps} = \frac{d_{max}}{V_{ps}}$, if the train speed during passing is supposed as V_{ps}.

By substituting $d = V_{ps}.t + d_0$ in (4.4):

$$rss(t) = P_{r_0} - 10\gamma log(V_{ps}t + d_0) + X_\sigma \qquad (4.12)$$

Hence, model for non-stopping stations will be as follows:

$$A_{passing} = 2 \sum_{t=0}^{t_{ps}} th_{ps}.\Delta t \qquad (4.13)$$

where th_{ps} is obtained through similar steps in (4.6)-(4.9).

4.3.3 Total Model for Offloading in a Rail Network

Based on equations in (4.10) and (4.13), for a train with N_{stp} stopping and N_{pss} passing stations through its trip, the total model, A_{total}, will be obtained as follows:

$$A_{total} = \sum_{i=1}^{N_{stp}} (th_{dw}^i . t_{dw}^i + 2 \sum_{t=0}^{t_{max}^i} th_t^i . \Delta t^i) + \sum_{j=1}^{N_{ps}} (2 \sum_{t=0}^{t_{max}^j} th_t^j . \Delta t^j) \qquad (4.14)$$

In (4.14), i refers to the station number for stopping stations and $1 \leq i \leq N_{stp}$. Similarly, j is to the station number for passing stations and $1 \leq j \leq N_{pss}$.

4.4 Simulation

To validate our developed analytical model, we compare the results of our analytical model with the results obtained from Omnetpp version 5.4.1, which is a powerful open source tool for network simulations. We assume that an IEEE802.11ac-based WLAN is dedicated for data offloading at each train station. IEEE802.11ac Standard can support channel with different bandwidths including 20MHz, 40Mhz, 80MHz and 160MHz. Therefore, we firstly, assume that only 20 MHz channels are available for data offloading, as the worst case scenario. In this case, we obtain the results for different values of transmitter powers and path loss components in both analytical and simulation environments. Then, we estimate the maximum capacity of data offloading for both types of stopping stations and passing stations.

Assuming that the dwell time of a train is permissible to vary between 20 to 60 seconds, we set $t_{dw} = 20 sec$, as the minimum guaranteed value and the worst case scenario for data offloading at stopping stations. For acceleration of trains during reaching and leaving a station, we suppose a similar value of $1 m/s^2$ with negative and positive signs,
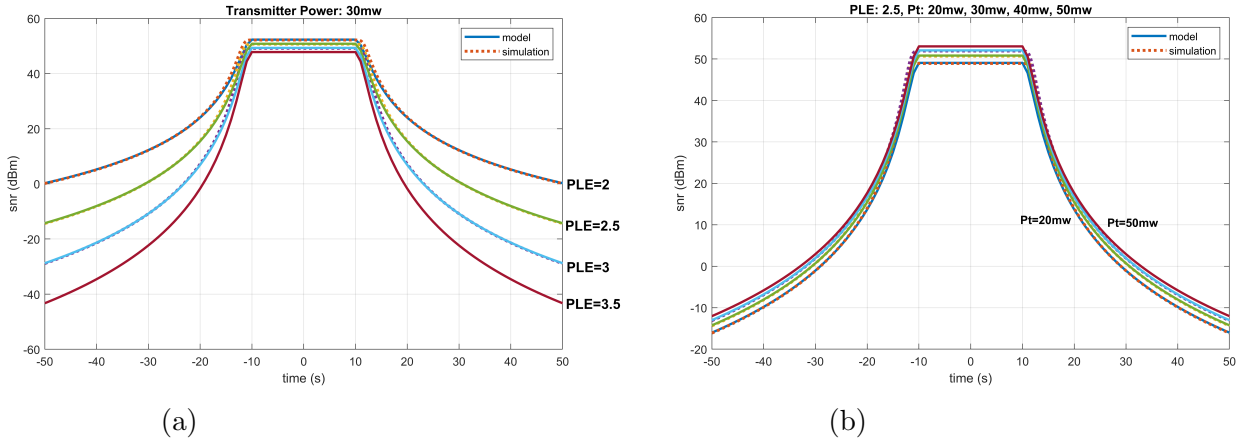
Figure 4.3: SNR vs. time for different values of a) PLE and b) transmitter power

respectively. For PLE, we apply different values including 2 (as free space), 2.5, 3 and 3.5 to show the performance of our model for stations in different environments. This is a realistic assumption for a rail network, as every station might be located in places with different environmental conditions. Additionally, we apply different values of transmitter power including $20mw$, $30mw$, $40mw$ and $50mw$ as one of the effecting element in the test results. However, due to the similarities and to avoid repeated figures, we only illustrate the results of some selected scenarios. For the all case studies, the background noise level at receiver is set to $-90dBm$. We also set the MAC efficiency to 44 percent in our analytical model, which is directly obtained from simulation results in Omnetpp. Additionally, to avoid generating results with stochastic elements, we have not considered the shadowing effect at stations as this effect causes random elements at every simulation which is not the scope of this work.

Figure 4.3 shows SNR versus time for different environments and transmitter powers for both analytical and simulation results. As illustrated, the analytical model can accurately follow the simulation model and can achieve up to 98.67 percent with reference to the simulation results from Omnetpp.

Figure 4.4 shows the estimated throughput versus time for different values of PLE and
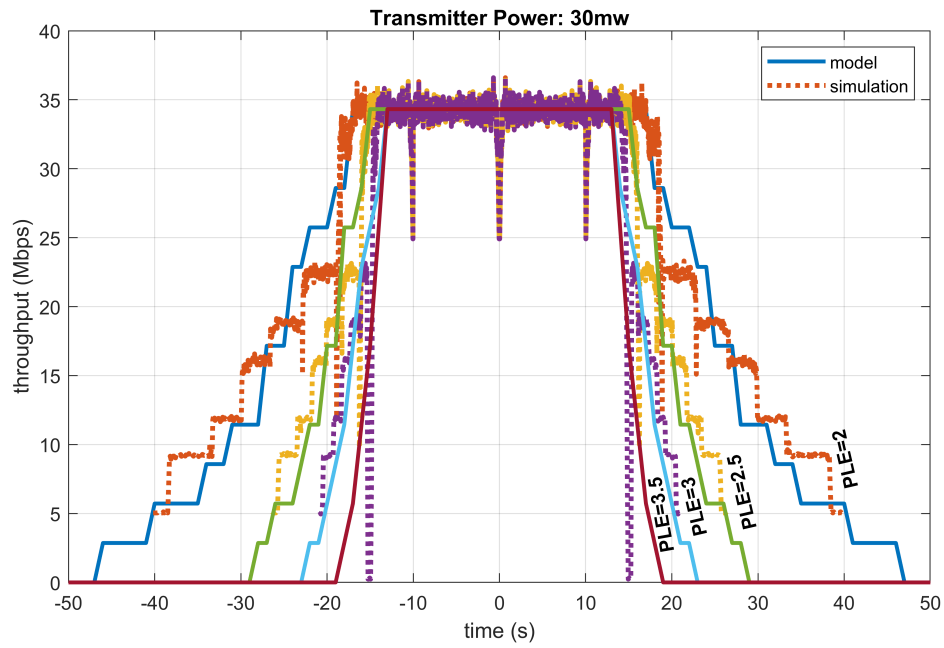
(a)
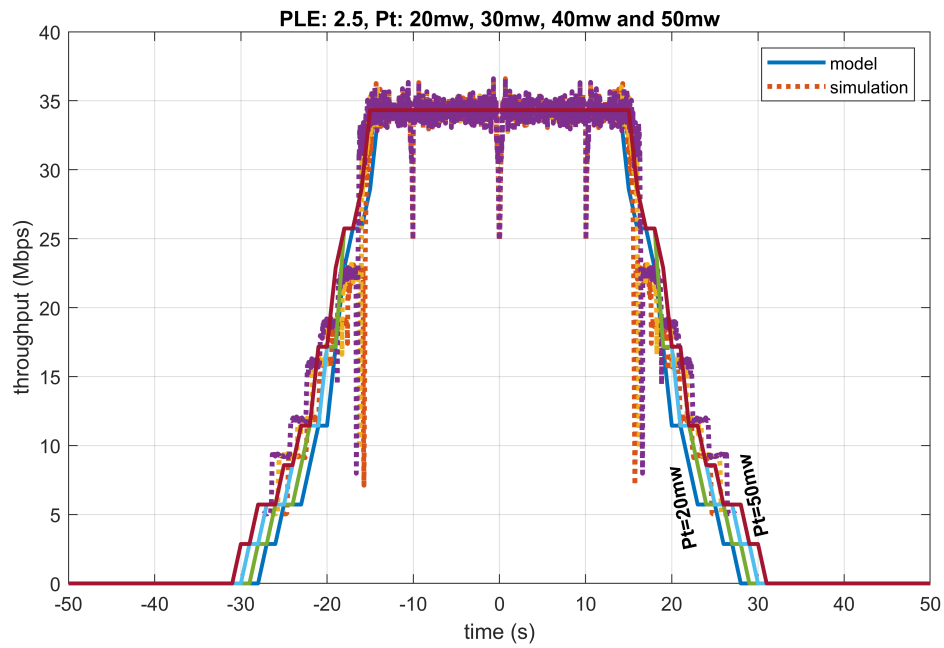
(b)

Figure 4.4: Throughput vs. time for different values of a) PLE and b) transmitter power

transmitter power. The differences between proposed and simulation results are due to employ different data rate control methods. We applied RBAR method in our analytical model to theoretically estimate the maximum data throughput. However, for simulations in Omnetpp, we used AARF algorithm. Additionally, this figure shows the dependency of transmitter power and station environment on data throughput.
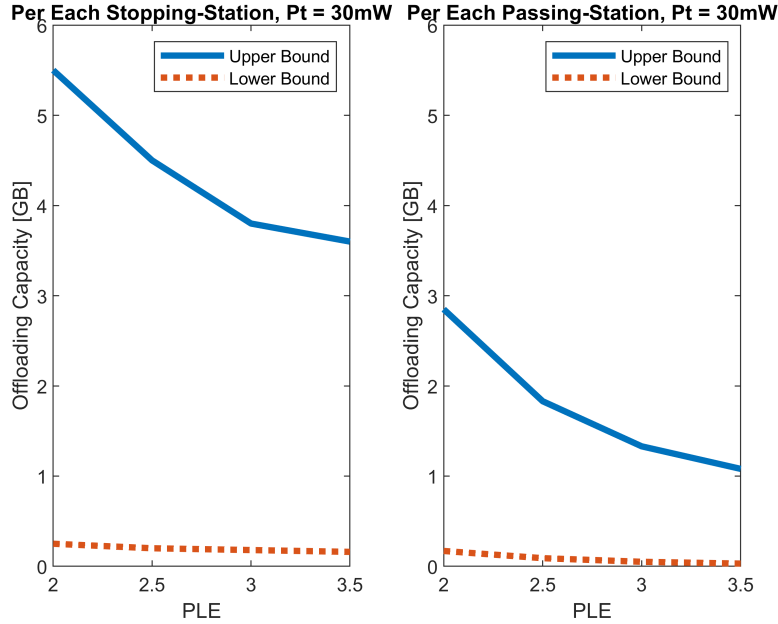
Figure 4.5: The theoretical lower and upper bounds of offloaded data estimated by the proposed model for different environments at $P_t = 30mw$: a) for stopping stations, b) for passing stations

To estimate the upper and lower bounds of offloading capacity based on the proposed model, we have illustrated the capacity for different environments for a given transmitter power of $30mw$ in Figure 4.5. For the lower bound, we assume that only wireless channels with 20MHz bandwidth and one spatial stream is available and $GI = 800ns$. However, for the upper band, we have supposed 160MHz channels, 3 spatial streams and $GI = 400ns$. According to these assumptions, for example for an environment with PLE=2.5, we can theoretically achieve up to 4.42 GB and 1.85 GB offloading capacity for every stopping or passing stations, respectively (Figure 4.5).

It is imperative to re-emphasize that in the all above results, we have assumed IEEE802.11ac protocol for the physical layer of WLAN. However, by employing new emerging technologies with higher data rates, we can offload much more data using our proposed scheme.

4.5 Conclusion

In this chapter, we proposed the existing WiFi networks at stations as intermediate access points for delayed offloading of big sensor data from trains to data centers. As the proposed method uses the existing WLAN at stations for data offloading, we will not need to install any extra communication infrastructure. We developed an analytical model for the offloading task that can estimate the offloading capacity for passing as well as stopping stations. Simulation results showed an accuracy of 98.67 percent for our developed model. Additionally, by using the proposed station-based offloading scheme, we can theoretically offload up to 5.43 GB with current offloading standards and several hundreds of GB with the future ultra-fast offloading technologies such as IEEE802.11ay.

In this chapter, we did not affect the location of antennas in the offloading model assuming that there are enough antennas along the trains and stations. Therefore, as a future work, we can make the model more realistic by considering the antennas effects on the offloading capacity. We have also assumed there is a dedicated WLAN at every station that are always available for data offloading for any train. However, WLAN may be unavailable in some situations due to poor signal conditions, lack of free channels because of other communications, etc. Hence, in the next chapters, we will consider alternative methods such as train-to-train and train-to-wayside communications.

5 Train-to-Train Data Communication Method

5.1 Introduction

Intelligent transportation systems (ITS) will generate several Petabytes of data from various IoT devices such as condition monitoring sensors and surveillance cameras [99]. To give a picture of such data, [?] provided a calculation that showed the size of vibration data relating to only one bearing of a train can be over one terabyte per hour. Such massive data should be transferred to data centers for further analysis using big data analytics (BDA) to improve operation and maintenance services of a rail network [100]. Since the current cellular networks cannot transfer such huge amount of data in an efficient, cost-effective and reliable manner, transmission of that data from ITS to data centers will be a significant issue [101]. Even with the new emerging cellular-based vehicular communication methods such as LTE-V2V or 5G-V2V [?]-[?], having an alternative method due to the rapidly growing demand for cellular networks specially during peak times can be highly beneficial [7]. Rail transportation systems (RTS) as one of the main type of ITS is not also an exception from such massive issue [102].

There are two types of IoT data in RTS include *'mission-critical'* data (MCD) and *'mission-non-critical'* data (MnCD). MCD is related to safety issues and must be im-

mediately informed and transferred to control centers. However, MnCD will be used for improvement of future maintenance and operation services and therefore, can be delivered with delay. Assuming every train has an appropriate edge processing unit that can classify IoT data (collected from various sensors along a train) in a real-time manner, the MnCD can be distinguished and separated from MCD [8]. In this way, we can safely use delayed offloading methods for MnCD.

In the two previous chapters, we firstly proposed a classification algorithm in [8] that could determine the class of data. Then, we developed a model for transmission of delay-tolerant part of data using train-to-station (T2S) communications [39]. Therefore, as a continuation of the previous two works, in this chapter, we propose train-to-train (T2T) communications as well as T2S communications to increase the amount of data offloaded through moving trains. In this way, not only the amount of offloaded data will be increased, but it will also provide faster data transfer for applications such as a request-based data demand with restricted delay. Such data can be part of a video stream from a surveillance camera in a moving train requested by a police officer. In such situation, assuming lack of reliable and robust cellular network (that is quite common in the harsh environment of rail tracks), a T2T data exchange strategy can be a vital solution by forwarding the requested data by a train that is running as fast as 110 km/h towards the requested location.

Generally, in vehicular communications (that is T2T communications in this chapter), we need to determine the specifications of contacts (i.e. duration, rate and location of contacts) among vehicles (i.e. trains in this chapter). To do that, we need to have access to the real traffic traces of all vehicles running in our desired network. In our current approach , we need to know those traffic traces (i.e. speed and location vs. time) for all trains running in a given rail network. Such data can be manually collected by a data collection system equipped with a GPS module, or can be provided by railway

authorities. However, manual collection of such massive data for a large network with many trains (e.g. for large cities) will take a long time. Additionally, such method will not be an accurate or even possible in all situations due to lack of GPS data. One example of such situation can be when trains running around urban areas with tall and congested buildings. Another example of GPS issue is when a train is moving inside of a tunnel. In such situations GPS signal can be highly intermittent and really poor or even be totally lost during data collection. This can also occur for Assisted GPS (A-GPS) because signals from cellular networks, which is used by A-GPS, can also be poor or unavailable in some locations such as tunnels.

Due to the above issues, we propose a novel method to provide train mobility traces based on trip timetables. As timetables are currently available in real-time, the obtained mobility traces will be in real-time too and can be upgraded upon occurring changes in the train services. Additionally, we use an energy optimization algorithm in the proposed mobility model. Therefore, the results generated by the model can be used as a travel trajectory guidance (optimal speed-position diagram of a train) that can advise the train driver (in a manual mode operation) or the driving system (in an autonomous drive-less train) to operate in the most optimum manner.

We firstly develop an algorithm that determines a train mobility model (TMM) based on its timetable. Then, we develop another algorithm to estimate the specifications of train contacts (i.e. duration, frequency and location of contacts) using the mobility traces generated from the proposed TMM. As the velocities of the contacting trains are determined by the proposed TMM, we can also estimate the Doppler shift (DS) during T2T communications. Finally, we determine the T2T offloading capacity based on the contact specifications obtained from the proposed train contact model (TCM) via simulations in Omnetpp network simulator. On this basis, the main contributions of our current chapter can be briefly described as follows:

1. We propose a novel method to determine a train traffic model using its trips timetable. Since the trip timetables are currently available in real-time, the proposed model can provide real-time mobility traces.

2. We provide a solution for collection of GPS-related data when GPS signal is poor or unavailable such as around urban areas or inside of tunnels.

3. We provide a travel trajectory guidance to have energy-optimized operation for train services.

4. We develop an algorithm to find the contact specifications between trains needed for T2T communications, in a deterministic (vs. stochastic) and real-time manner;

5. We estimate the data volume that can be offloaded through T2T communications in a rail network.

The rest of the chapter is organized as the following. We present our proposed models for train mobility and contact in Section 5.2. Simulation and results are presented in Section 5.3. The last section is related to the conclusion and our future work.

5.2 system models

As Figure 5.1 shows, various communication methods that could be supposed for data exchange in a rail network include T2S, T2I and T2T. T2S and T2I methods have been studied in other approaches (such as [39] and [1]) and are not the focus of this study. For T2T communications, we need to find the details of inter-train contacts. Such details includes duration, rate and location of contacts. In order to determine contact details (CD), we need to know the instantaneous location and speed of each train. Therefore, we propose a mobility model to provide instantaneous traffic traces for every train. Then,
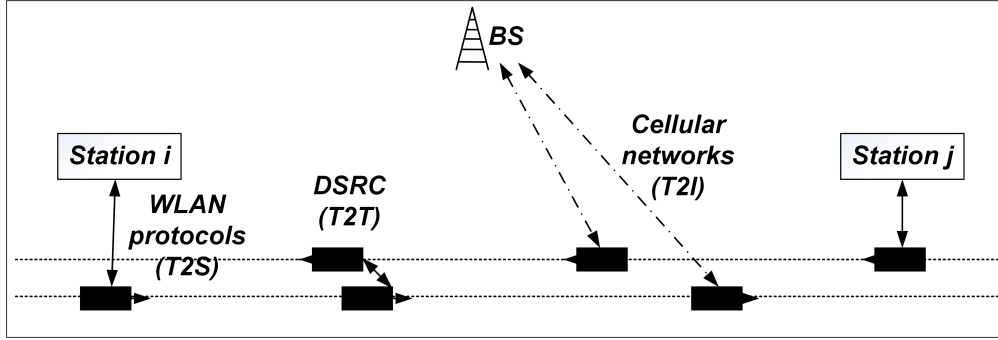
Figure 5.1: System overview

we develop a second model to determine T2T CD by using those traffic traces obtained from the proposed TMM.

5.2.1 Train mobility Model (TMM)

To estimate the T2T offloading capacity in a rail network, we need to determine the specification of contacts between trains. To find the contacts, we firstly need to know the traffic traces of the running trains. Therefore, in this section, we develop a mobility model that can provide the traffic traces of trains without using GPS. This enables the TMM to provide train traffic traces even where GPS signal is unavailable or very poor such as inside tunnels. To do this, we propose an innovative algorithm that can provide the train mobility traces only based on trip timetables. As trip timetables are currently available in real-time, the proposed model will be also in near real-time and can be quickly updated when a change happens in a trip time. For example, if a train stops longer than its expected dwell time at a given station that causes a change in train departure time at such station. In this case, the algorithm will update the model based on the new departure time. Figure 5.2 shows our proposed methodology for building the TMM.

Figure 5.3 shows a trip for two trains that are commuting between two stations. To run
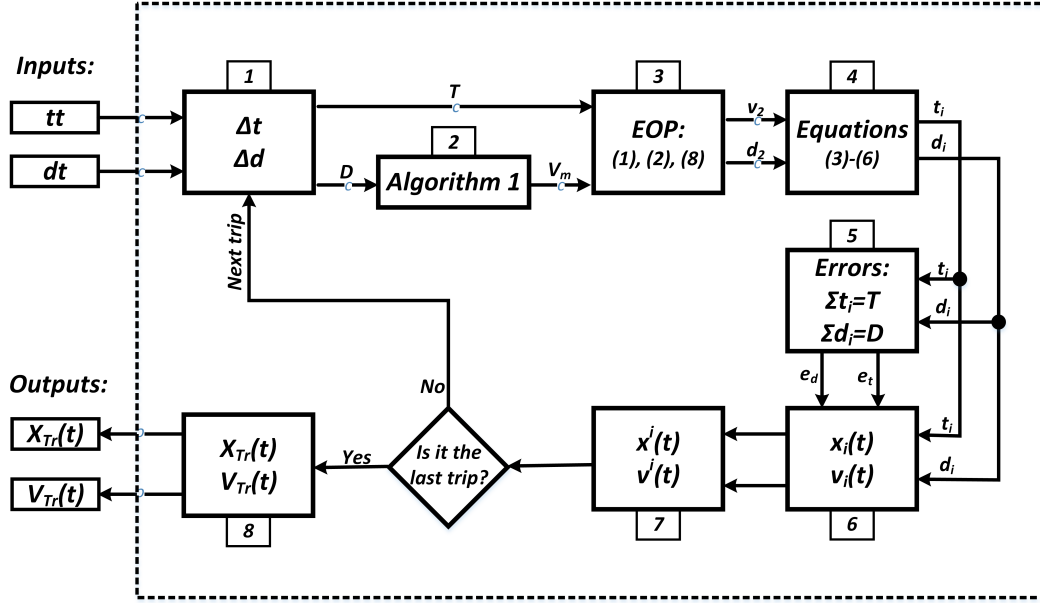
Figure 5.2: Train mobility model

TMM for every train, we firstly need to have the tables of time and distances for all trips that such train will have during its total travel. Trip time-tables, *tt*, are easily accessible through the websites of railway authorities and are available to public in both versions of offline and online. Distance-tables, *dt*, that shows the distances between stations can be also easily created from the network specifications. Based on those two tables, the TMM algorithm will extract the duration (T) and distance (D) of all trips of every train through the step 1 in Figure 5.2. The algorithm will do this for trains in both directions as each direction has a different timetable that sometimes has different trip duration. This enables the TMM to work correctly for trains in both directions. Then, based on the extracted trip distance, the TMM algorithm will determine the maximum optimal speed (MOS) of train for that trip using another internal algorithm shown in Algorithm 5.1 (step 2).

Algorithm 5.1 that can quickly calculate the MOS for a train in a given trip is developed based on the concept proposed in [103]. Feng et. al in [103] showed that to have an

S_i Tr_{ij} Tr_{ji} S_j

Trip distance: D
Trip time: T
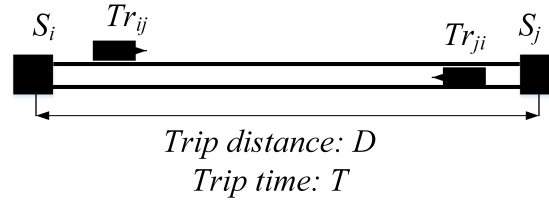
Figure 5.3: Train traffic model

optimal trip from both aspects of energy and time, the maximum speed should be within the range of: 1) 30-70 km/h for trip distances less than 1800 m, and 2) 70-V_{max}km/h for distances longer than 1800 m. V_{max} is the maximum allowed speed (MAS) of trains set based on the design specifications of the given rail network. On this basis, we define one limit for distance and three limits for speed through steps 1 and 2 in Algorithm 5.1. Then, based on the trip distance, the MOS will be calculated through the rest of steps of the algorithm.

Generally, trains should follow a trajectory guidance to consume optimum energy during every trip. To meet this goal, we developed an energy optimization problem (EOP) based on the concept described in [1]. Zhu et. al. in [1] proposed an optimal trajectory guidance for trains to have optimal energy trips. An optimal trajectory guidance is the pattern of change of train speed versus distance for a trip between two consecutive stations so that the energy consumed by the train gets optimum during such trip. Figure 5.4 shows such travel trajectory guidance for a typical train. As shown, there are 4 stages for a train during its trip include: 1) traction, 2) speed-holding, 3) coasting and 4) braking. In order to have an optimized-energy trip, in the second stage (i.e. speed-holding), the train kinetic energy should be equal to the energy required to overcome the friction (total mechanical and wind frictions). Therefore, the difference of kinetic and friction energies should be minimized. On this basis, we developed an EOP represented through (5.1), (5.2) and (5.8).

Algorithm 5.1 MOS determiner

Input:

- **trip distance** (D)

- **MAS of trains in the given network** (V_{max})

output: MOS (v_m)

1. define distance limit to 1800 m:

- $distance.limit = 1800m,$

2. define three speed limits as follows:

- $speed.limit\,1 = 30 \times 0.28 \; m/s,$

- $speed.limit\,2 = 70 \times 0.28 \; m/s,$

- $speed.limit\,3 = V_{max} \times 0.28 \; m/s,$

3. if $D \leq distance.limit$, **calculate** v_m **as follows:**
$v_m = speed.limit\,1 + (speed.limit\,2 - speed.limit\,1)/distance.limit \times D$
else define speed as follows:
$speed = speed.limit\,2 + (speed.limit\,3 - speed.limit\,2)/distance.limit \times (D - distance.limit)$
4. if $speed \leq speed.limit\,3,$
$v_m = speed$
else
$v_m = speed.limit\,3$
5. end

In the developed EOP, (5.1) is the minimum function that defines the minimum required energy per unit of train mass to keep the train speed at v_2 during stage 2 in the presence of friction. In (5.1), v_2 and d_2 are the speed and trip distance of train at the second stage and a_f is the acceleration required for speed-holding against friction. The first and the second parts of (5.1) are kinetic and friction energy per unit of train mass, respectively. The first constraint of the EOP, i.e. (5.2), is the MOS that was obtained from Algorithm 5.1. As the variables of the EOP min-function i.e. (5.1) are v_2 and d_2, all constraints must be based on these two variables only. The first constraint i.e. (5.2), is already adapted to the min-fucntion. For the second constraint, we combine the time and distance constraints i.e. (5.3) and (5.4), together with the mobility equations i.e. (5.5)-(5.7) and
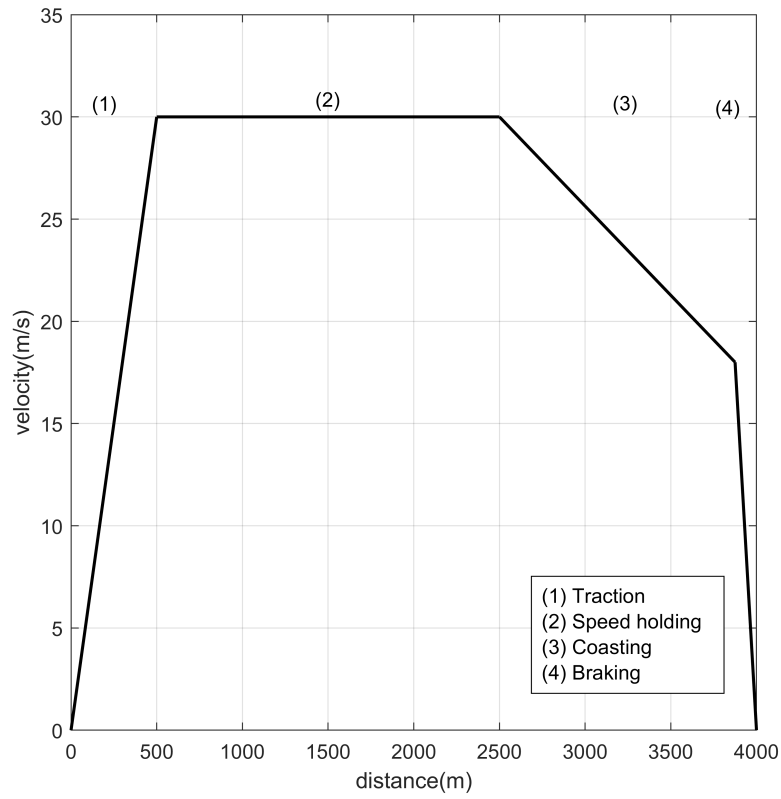
Figure 5.4: Optimal train guidance trajectory [1]

extract a nonlinear constraint represented by (5.8). To have clearer formulation, we define a simple rule for notations as follows. In all the equations, the parameters with notations 1-4 are related to the phases of traction, speed-holding, coasting and braking, respectively. Such notations can be also found from Figure 5.4.

minimum function:

$$f(v_2, d_2) = \frac{1}{2}v_2^2 - a_f d_2 \tag{5.1}$$

first constraint:

$$v_2 \leq v_m \tag{5.2}$$

trip time constraint:

$$\sum_{i=1}^{4} t_i = T \tag{5.3}$$

trip distance constraint:

$$\sum_{i=1}^{4} d_i = D \tag{5.4}$$

The relations between trip times and distances at different stages can be easily obtained from train mobility equations as follows:

$$t_1 = \frac{v_2}{a_1}, \ t_2 = \frac{d_2}{v_2}, \ t_3 = \frac{v_4 - v_2}{a_3}, \ t_4 = \frac{-v_4}{a_4} \tag{5.5}$$

$$d_1 = \frac{v_2^2}{2a_1}, \ d_3 = \frac{v_4^2 - v_2^2}{2a_3}, \ d_4 = \frac{-v_4^2}{2a_4} \tag{5.6}$$

Then, from (5.3) and (5.5),

$$v_4 = \frac{a_4(\frac{a_3}{a_1} - 1)v_2^2 - Ta_4a_3v_2 + a_4a_3d_2}{(a_3 - a_4)v_2} \tag{5.7}$$

Finally, from (5.4), (5.6) and (5.7), the second constraint is obtained as follows:

$$\frac{a_1^2 + a_3a_4 - a_1a_4 - a_1a_3}{2a_1^2(a_3 - a_4)}v_2^2 + \frac{Ta_4(a_1 - a_3)}{a_1(a_3 - a_4)}v_2 +$$

$$\frac{a_3a_4}{2(a_3 - a_4)}\frac{d_2^2}{v_2} + \left[\frac{a_4(a_3 - a_1)}{a_1(a_3 - a_4)} - 1\right]d_2 + \frac{a_3a_4T^2}{2(a_3 - a_4)} + d = 0 \quad (5.8)$$

(5.8) is a nonlinear constraint for the EOP.

$$\overline{x(t)} = [\overline{x_1(t)}\ \overline{x_2(t)}\ \overline{x_3(t)}\ \overline{x_4(t)}] \qquad (5.9)$$

$$\overline{v(t)} = [\overline{v_1(t)}\ \overline{v_2(t)}\ \overline{v_3(t)}\ \overline{v_4(t)}] \qquad (5.10)$$

EOP represented by (5.1), as minimum function, is a linear problem (LP) that contains an inequality and a nonlinear equation as constraints, which are (5.8) and (5.2), respectively. To solve such EOP, we used an interior-point algorithm - the barrier method, which is an appropriate method for the problems with nonlinear constraints.

By solving the EOP, the optimal values of v_2 and d_2 will be determined through the step 3 in Figure 5.2. Then based on the obtained optimal values of v_2 and d_2, the numerical values of t_i and d_i will be calculated through (5.5) and (5.6). In order to increase the accuracy of TMM, the algorithm will calculate the errors of the time and distance obtained from the EOP through (5.3) and (5.4). Then, the errors will be re-applied to the algorithm to obtain results with higher accuracy. Finally, the vectors of location and speed for the every stage of travel trajectory will be determined by using (5.3)-(5.7) and mobility equations related to every stage. The total vectors of speed and location for one trip are obtained by concatenating the vectors of every stage through (5.9) and (5.10). Obviously, for a train with more than one trip, we will have several

vectors for locations and speeds. Hence, the TMM algorithm will repeat the whole steps until the end of train travel. Mobility vectors of a train can be shown as follows:

$$\overline{X_{Tr}^m(t)} = [\overline{x^1(t)}\ \overline{x^2(t)}\ ...\ \overline{x^{k_m}(t)}] \tag{5.11}$$

$$\overline{V_{Tr}^m(t)} = [\overline{v^1(t)}\ \overline{v^2(t)}\ ...\ \overline{v^{k_m}(t)}] \tag{5.12}$$

where $\overline{X_{Tr}^m(t)}$ and $\overline{V_{Tr}^m(t)}$ are the obtained location and speed vectors, respectively, for a train with identity number of m and k_m number of trips through its travel.

5.2.2 T2T contact model (TCM)

By running the TMM for a given network, we will obtain the total traffic traces of all trains for that network. In this case, we can simply extract the contacts between two trains through seeking the intersections among location vectors of all desired trains. Based on this simple technique, we develop the TCM represented by Algorithm 5.2. As a general methodology, we can search the whole database related to all lines of a network, to find the trains with location intersections. However, this method will be extremely time-consuming specially for large networks. The effective method is to find the lines/trains that have common sections/routes with each other. This can be found from the map of a rail network. One example of such case with common routes can be the trains that are traveling in the same line, however, in the opposite directions. In this way, Algorithm 5.2 will only seek the traffic traces of the trains with common routes. This will cause a significant reduction in the algorithm processing time.

As shown in Algorithm 5.2, to find the contact between two given trains, we need to have the traffic traces of both trains, i.e. $\overline{X_{Tr}^m(t)}$ and $\overline{X_{Tr}^n(t)}$, as well as the range of T2T

Algorithm 5.2 TCM algorithm

---

inputs: $\overline{X_{Tr}^m(t)}$, $\overline{X_{Tr}^n(t)}$ and z
outputs: $\overline{X_c(t)}$, N_c and $\overline{T_c}$
1. define location difference as follows:
$diff_x = \left| x_i^m(t) - x_j^n(t) \right|$
2. for all traffic traces of both trains if $diff_x \leq \epsilon$, then set number and location of contacts as follows:
$N_c = N_c + 1$
$\overline{X_c(i)} = [x_i^m(t)]$ **or** $\overline{X_c(i)} = [x_j^n(t)]$
3. For every contact point, find the locations that:
$|x_i^m(t) - x_c(i)| \leq z/2$
4. calculate the contact duration from step 3 and set it as $\overline{T_c}$.

---

communication zone, z. We assume that the trains m and n have common routes with each other through daily operations. z can be theoretically obtained from summation of the maximum communication range and total lengths of the trains assuming that trains equipped with multiple antennas appropriately installed to provide equal coverage along train length (e.g. in the first, middle and last carriages). By having traffic traces of both trains, Algorithm 5.2 compares all elements of $\overline{X_{Tr}^m(t)}$ and $\overline{X_{Tr}^n(t)}$ and determines all the points that have differences less than a small pre-defined value, ϵ. ϵ is set to a small non-zero value such as $5m$ to avoid losing any T2T contact. After finding the all contact points, the contact duration is determined based on the communication zone through steps 3 and 4 of the algorithm. In Algorithm 5.2, $\overline{X_c(t)}$, N_c and $\overline{T_c}$ are vector of contact locations, number of contacts and vector of contact duration, respectively.

5.2.3 T2T communication approach

Figure 5.5 illustrated the T2T communication approach with respect to the TMM and CM. The transmitter (TX) sends signals (S_T) through the communication channel (CC). For path-loss modeling, we use Two-Ray Interference best fitted to the short-range communications in the rail environment [104] and [105]. The availability of CC between trains is equal to the contact duration (CD) that obtained from the CM. (5.13) shows
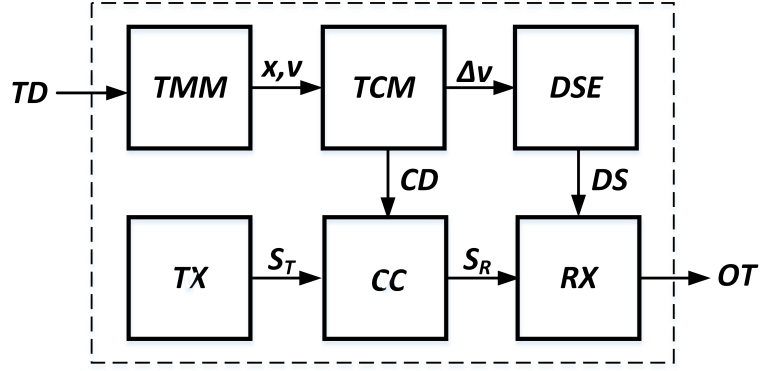
Figure 5.5: T2T communication approach

the mentioned CC model assuming that $S_T = A_T.cos(2\pi f_0 + \varphi_T)$, where A_T, f_0 and φ_T are the transmitter power, the carrier frequency and the transmitter phase, respectively.

$$S_R = A_R.cos[2\pi(f_0 \pm \Delta f) + \varphi_T + \varphi_C] \tag{5.13}$$

In (5.13), S_R and A_R are the received signal and amplitude, respectively. φ_C is the phase shift through the CC and Δf is the Doppler shift (DS) due to the trains mobility. DS is estimated through Doppler Shift Estimator (DSE) as follows:

$$\Delta f = \pm \frac{|\Delta v|}{c}.f_0 \tag{5.14}$$

where in (5.14), Δf is the frequency change caused by Doppler Effect (DE), c is the speed of light and f_0 is the carrier frequency. DS is positive when trains are approaching and will be negative if trains are moving away. DS is used in the receiver for Doppler shift compensation (DSC).

Finally, based on the level of the received power and the modulation and coding scheme (MCS), the offloading throughput (OT) is estimated in the receiver.

5.3 Simulation and Results

In this chapter, as we used general concepts in developing the algorithms described in Section 5.2, our proposed models can be applied in any rail network. However, as we currently have access only to Sydney Trains of Australia for our data collection, we selected Sydney Trains as a real rail network to evaluate our proposed models. Sydney Trains with coverage of 1,643 km of track across 9 lines, 175 stations and 360 million trips per year is one of the largest rail network in the world [106]. To show the ability of our algorithm in various situations, we did our simulations for three different lines including T1, T6 and T7. Figure 5.6 shows a section of Sydney Trains network that we selected as our case study. Line T1 with trips over 20 km is an appropriate case to test our algorithm for long trips and line T6 with stations less than 1 km distance is a proper choice for short trips. We also selected T7 to evaluate of our proposed TCM. In the all cases, we show the both results obtained from simulations as well as the related actual data. The type of trains used in our experiments are known as Waratah trains. According to the performance specification of Waratah trains that are widely operating in Sydney Trains of Australia, the accelerations related to each phase of traction, braking and coasting are set as follows [**?**]. The traction acceleration varies within 0.8-1.0 m/s^2 depending on the performance level defined by rail corporation authorities. Braking and costing decelerations are set within 0.9-1.05 m/s^2 and 0.18-0.22 m/s^2, respectively. Additionally, for Waratah trains weighing from 400 ton (empty) to 500 ton (full passenger), the friction acceleration falls within the range of 0.0129-0.0105 m/s^2 [**?**]. Therefore, for every acceleration, we firstly assume the maximum value of every range as the setting. Then, we show the impact of possible change of every acceleration within its range. To benefit from the strengths of A-GPS vs. traditional GPS, we used a mobile phone for collecting the required GPS data. The mobile phone that we used for data collection was an iPhone 6 through an application called *Sensor*
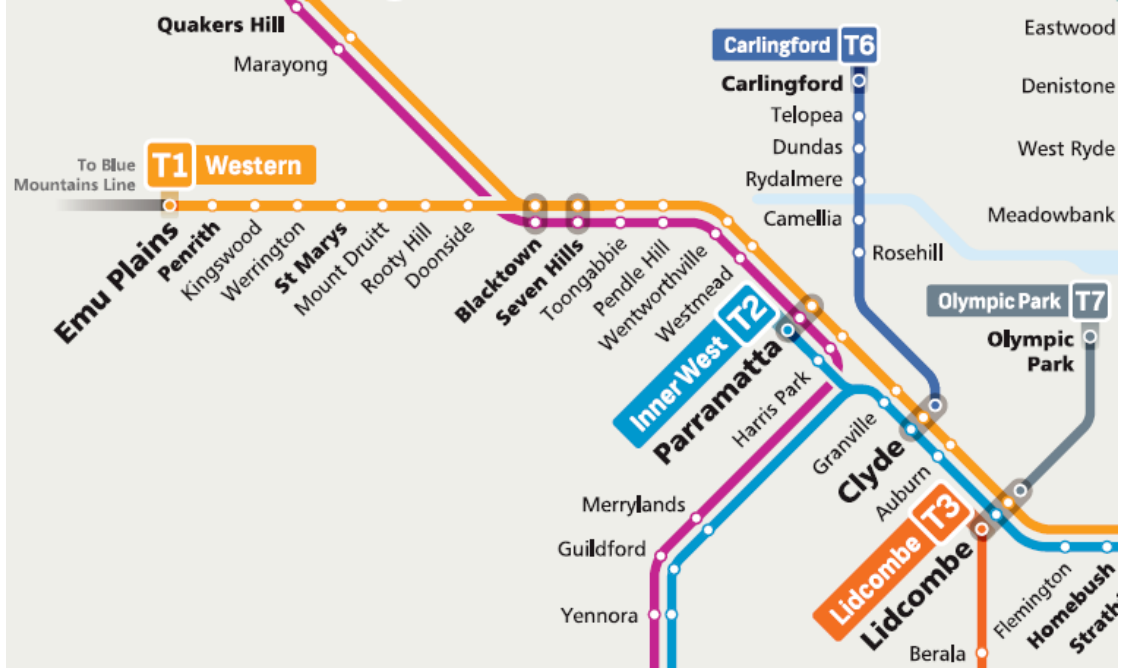
Figure 5.6: A section of Sydney Trains network as our case study [2]

Play. This mobile application can record GPS data with sample rate of 10 Hz.

5.3.1 Mobility Model Simulations

In this section, we use the most well-known model for train motion to compare with our proposed TMM with reference to the real data obtained from GPS [**?**] and [**?**]. This model works based on a differential equation, called Lomnossoff's equation, shown in (5.15). In the Lomnossoff's equation, M, x and F are the total mass of the train, displacement and train traction effort, respectively. Additionally, A, B and C are Davis constants and can be obtained from the technical specification of the given train.

$$M\frac{d^2x}{dt^2} = F - (A + B\frac{dx}{dt} + C\frac{d^2x}{dt^2}) \tag{5.15}$$

By solving (5.15), we can find the displacement of a train vs. time. However, as F is

Table 5.1: Comparison between accuracy of our proposed method and Lomnossoff-based method

Method	Distance Accuracy [%]		Speed Accuracy [%]	
	Line T6	Line T1	Line T6	Line T1
The proposed	98.61	98.31	75.43	85.83
Lomnossof	94.00	92.73	58.91	63.43

different for every stage of traction, holding, coasting and braking, we must solve such equation separately for every stage. This means that we need to solve four 2nd-order differential equations for every trip. Therefore, we will totally have $4n$ equations for a train with n trips. This compared with our model that is developed based on simplified algebraic relations is more complex and time-consuming. Additionally, speed of a train cannot be directly obtained from (5.15) and therefore, we need to use another differential equation, i.e. $v = dx/dt$ to find the speed at every stage. Moreover, as we see in the following, in comparison with our proposed model, the accuracy of the Lomnossoff-based model is lower. Table 5.1 shows the accuracy of every model with reference to the real GPS data for two different lines from Sydney Trains of Australia.

Figure 5.7 shows the simulation results obtained from both the proposed and Lomnossoff-based models with reference to the real data for a train in a round trip in line T6. Line T6 starts from Clyde as terminal A, and after passing 5 stations, will arrive at Carlingford (as terminal B), where the train will start to return after a couple of minutes stopping. Line T6 with total length of around 7 km is one the shortest line in Sydney Trains network and therefore, this line is a proper choice to test our algorithm for short trips. For better understanding, terminal A is assumed as the reference point in all the diagrams. Then, we converts the collected GPS data from degree to meter based on the distance from terminal A. As described in Section 5.2.1, our algorithm uses trip timetable and distances between stations as inputs to estimate traffic traces of a running train. Table 5.2 shows such inputs extracted from Sydney Trains website [2].
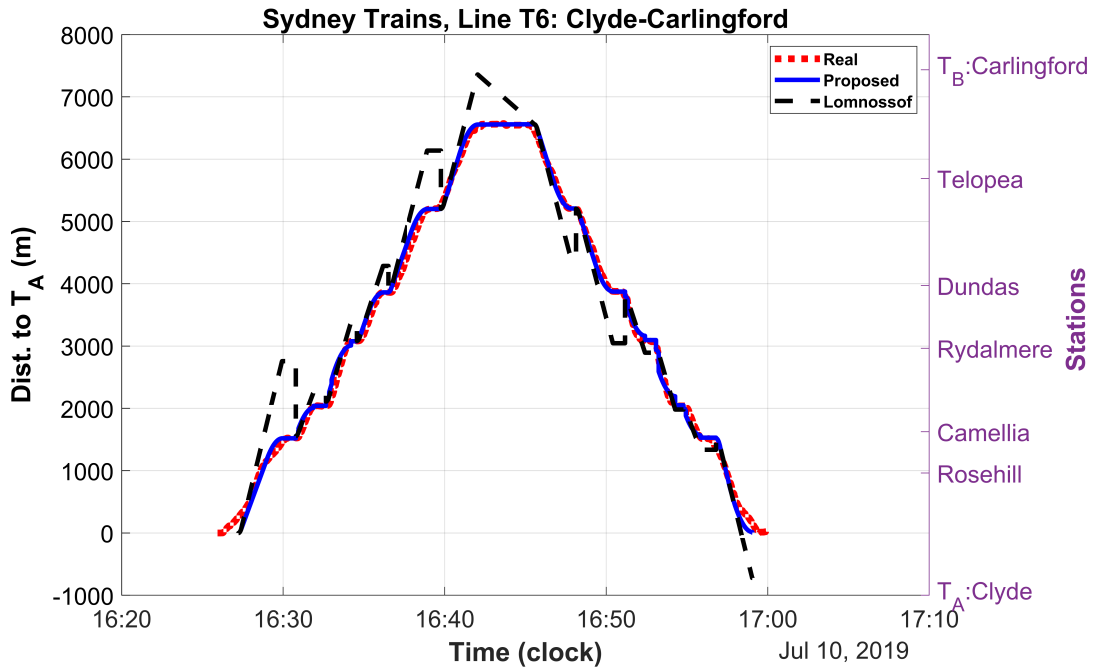
a) Distance vs. time

we did our simulations for three different lines including T1, T6 and T7. Figure 5.6 shows a section of Sydney Trains network that we selected as our case study. Line T1 with trips over 20 km is an appropriate case to test our algorithm for long trips and line T6 with stations less than 1 km distance is a proper choice for short trips. We also selected T7 to evaluate of our proposed TCM. In the all cases, we show the both results obtained from simulations as well as the related actual
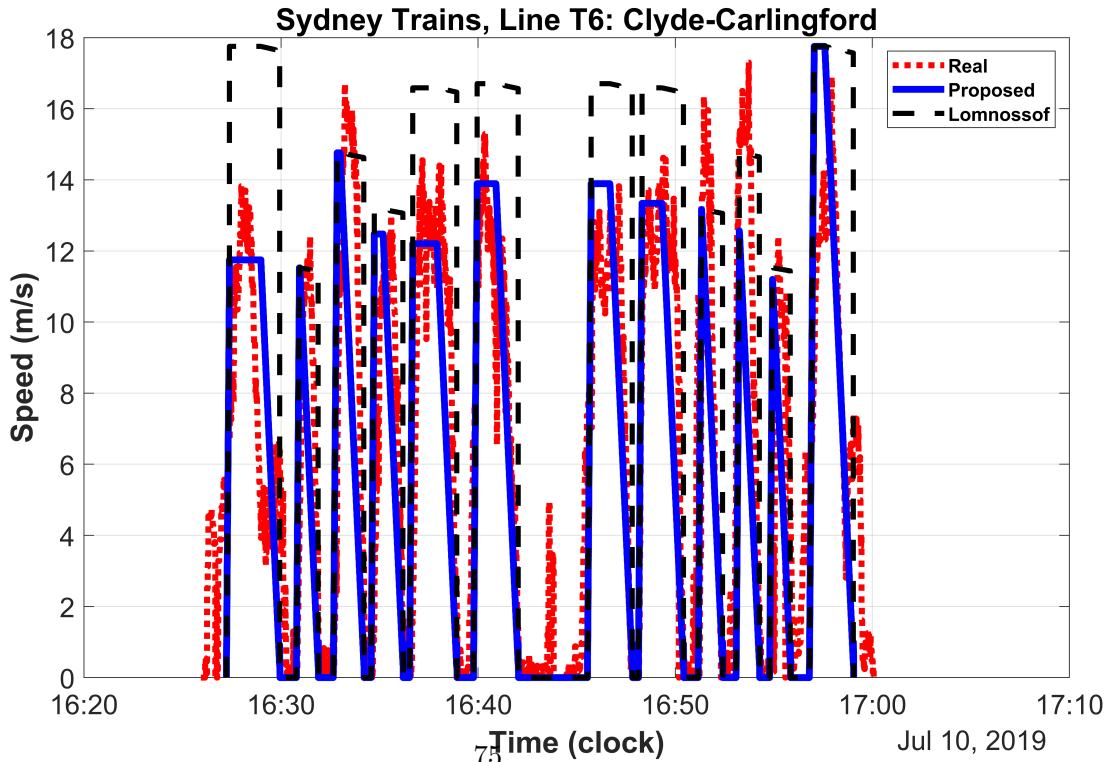
data.

b) Speed vs. time

Figure 5.7: Simulation results and real data for line T6, between Clyde and Carlingford

Table 5.2: Input data for simulation at line T6

Stations	Distance (from Clyde)	Trip Time 1: (from Clyde)	Stations	Trip Time 2: (from Carl.)
Clyde	0	16:26	Carlingford	16:45
Rosehill	1526	16:30	Telopea	16:47
Camellia	2043	16:32	Dundas	16:50
Rydalmere	3084	16:34	Rydalmere	16:52
Dundas	3867	16:36	Camellia	16:54
Telopea	5204	16:39	Rosehill	16:56
Carlingford	6560	16:42	Clyde	17:00

As can be seen in Figure 5.7, the results obtained from our proposed mobility model can properly follow the real data with high accuracy. Using Root Mean Squared Error (RMSE), the accuracy (which is defined as $100[1 - RMSE/trip\text{-}distance]$) in terms of distance is obtained 98.61 percent based on our calculation. As described, we also tried our algorithm for line T1, to ensure its correct performance within long trips. Figure 5.8 and Table 5.3 show the results and related input data, respectively. In this case, the accuracy of the proposed model in terms of distance is obtained 98.31 percent. As described in the beginning of Section 5.3, such result obtained assuming that the maximum possible value of every acceleration is set. However, all the accelerations may change within the defined ranges. Therefore, in Table 5.4, we show the maximum error that may occur due to the possible change of every acceleration in the range. As illustrated, for both lines, the maximum error is related to the change of coasting acceleration. Additionally, we calculated the combined error assuming that the maximum change is concurrently occurred in all the accelerations. Therefore, according to Table 5.4, we ensure that the maximum error due to the change of the accelerations will be less than 1%.

Contrary to distance, the results obtained for speed is less accurate. Using RMSE, the accuracy of simulated speed is obtained 75.43% and 85.83% for lines T6 and T1, respec-

76

tively. In order to understand the possible effect of such lower accuracy, it is necessary to know where the simulated speed is used. The simulated speed can be used for estimating the duration of contacts between two trains. However, as shown in Algorithm 5.2, the key element used for estimation of contact details is the distance, which has less than 2% error. Additionally, the simulated speed can be used for estimating Doppler shift (DS) in carrier frequency. In Section 5.3.3, we show the effect of speed error in DS estimation, which will be almost zero (for detail, see Table 5.6 in page).

Table 5.3: Input data for simulation at line T1: between Penrith-Parramatta

Stations	Distance (from Parra.)	Trip Time 1: (from Parra.)	Stations	Trip Time 2: (from Penr.)
Parramatta	0	14:22	Penrith	14:47
Blacktown	10580	14:30	Blacktown	15:01
Penrith	29540	14:44	Parramatta	15:10

5.3.1.1 Discussion about timetable changes

Railway authorities' ultimate goal is to provide on-time services with minimum changes compared to their announced timetables. However, in practice, they are not always successful and customers experience delays and changes in the announced timetables in

Table 5.4: The maximum errors obtained due to the possible change of the accelerations

Acceleration	Setting $[m/s^2]$	Possible Change [%]	Distance Error [%]	
			Line T6	Line T1
a_1	1.0	20	-0.07	-0.07
a_2	0.0129	19	-0.01	0.00
a_3	0.22	20	0.15	0.28
a_4	1.05	14	-0.05	0.00
Combined Error			0.07	0.22
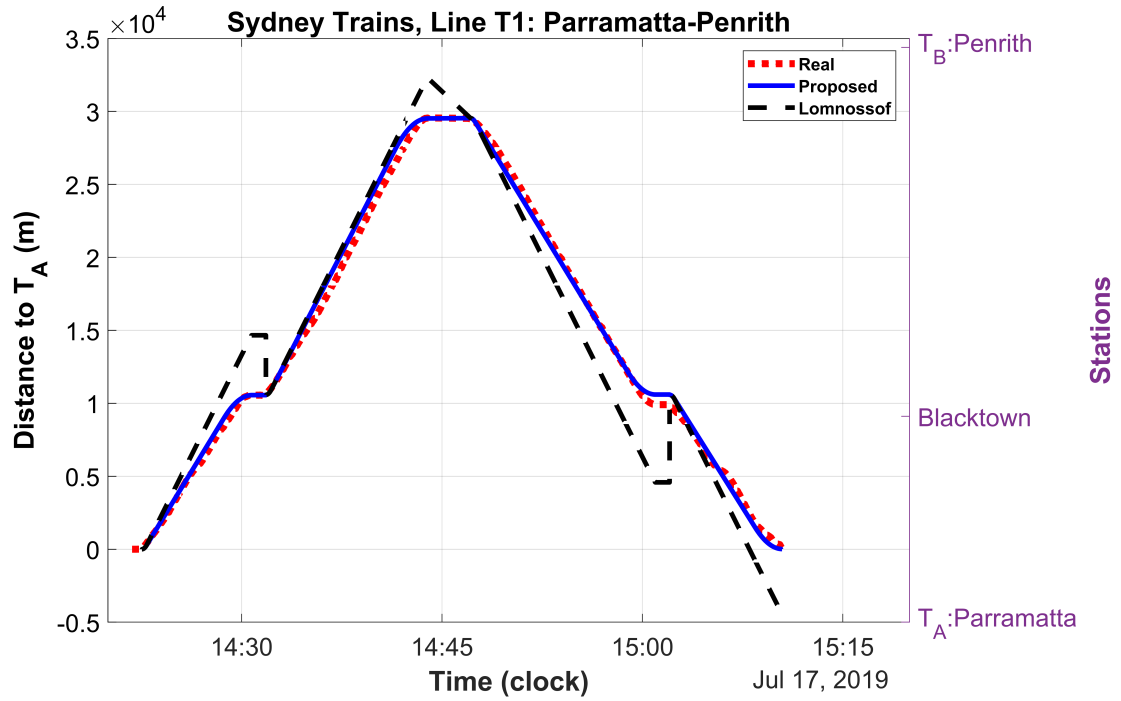
a) Distance vs. time
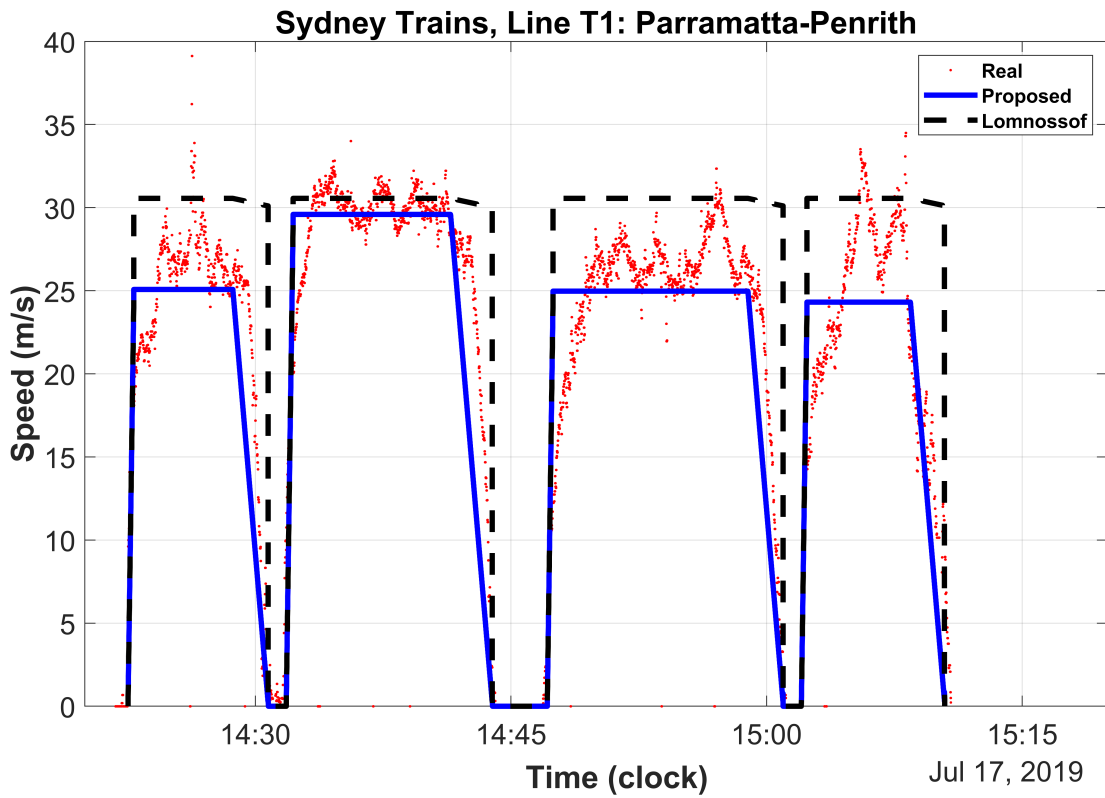
b) Speed vs. time

Figure 5.8: Simulation results and real data for line T1, between Penrith and Parramatta

some trips. As described, our proposed TMM worked based on timetables and can be quickly updated if a change in a given timetable is recognized and announced. However, unrecognized or delayed announcement of a change will cause error in the results of TMM. Therefore, we performed numerical analysis to determine the model error due to unannounced changes in timetables.

We assumed a range of zero to 30% changes in the timetables shown in tables 5.2 and 5.3. Then, we calculated the percentage of distance errors for every 5% intervals within the change range, i.e. at 0, 5%, 10%,...,30%. Figure 5.9 shows the obtained results. As illustrated, the errors of results obtained from TMM for 5% change in both timetables will be less than 3%, which means over 97% accuracy. Additionally, even with 10% unrecognized change in both timetables, the level of distance errors of TMM will be less than 5%. This means that 95% accuracy in TMM results will be guaranteed if the probable unrecognized changes in timetables are not exceeded 10%. Similarly, up to 20% change in the timetables will cause less than 10% distance errors. It should be emphasized again that all those errors will occur if the change in timetables is not recognized or announced. Otherwise, for any indicated change in a timetable, the TMM will quickly update the results based on the new time. Quickly update means that the total processing time of the TMM algorithm upon receiving the new change in timetable is quite short around $0.5\,s$. This ensures the quick performance of our algorithm. However, the extra transmission time from the external communication system (ECS) that is responsible for reporting the change will be added to the algorithm processing time. Such extra time will depend on the volume of the reporting massage and the wireless protocol that is used in the ECS. As the dataset relating to all the trips is already stored in TMM internal memory, such message only needs to report the changes instead of all the timetable. As an example, assuming that the ECS uses LTE network to send a 1 MB message. In this case, given the theoretical data rate of a LTE channel with 20 MHz bandwidth that is around 100 Mbit/s, the extra transmission time will be much
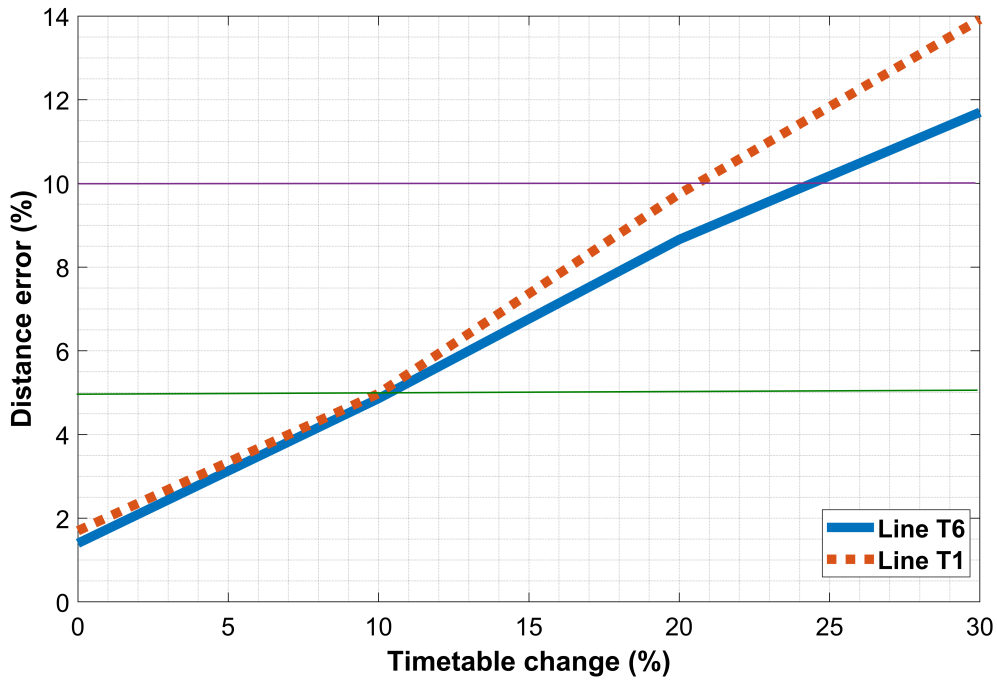
Figure 5.9: Distance errors caused by unrecognized changes in timetables for lines T1 and T6

less than half a second.

5.3.2 T2T Contact Simulation

Through simulations in Section 5.3.1, we ensured that the performance of the proposed TMM is reliable and accurate. Therefore, we can use the results obtained from TMM to validate the Algorithm 5.2 and estimate the train contact features (i.e location, speed and duration). We investigated two different scenarios: a) for two trains only, to have a more clear illustration of location and speed diagrams during a contact, and b) for all services running in line T7 every day, to find the total capacity of contacts among all trains in this line. Firstly, we evaluate the TCM by simulation of line T6 which we have investigated for train mobility algorithm in Section 5.3.1. We used this line to confirm the correct and accurate performance of Algorithm 5.2. Figure 5.10 shows the
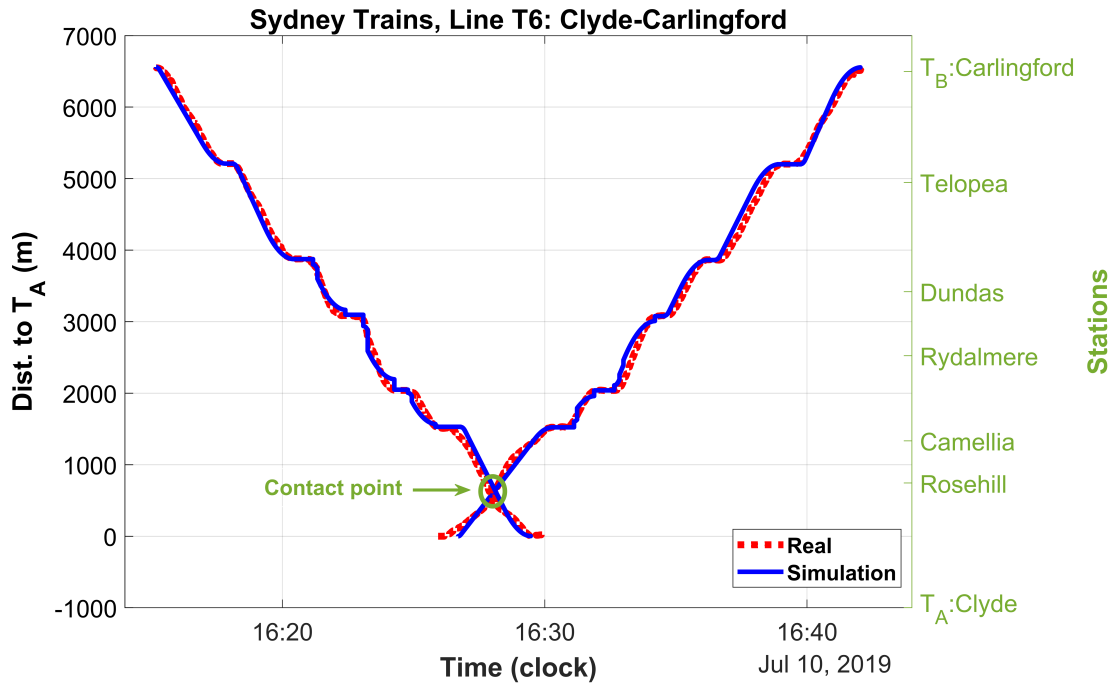
Figure 5.10: Distance vs. time during contacts for two trains in line T6

simulation results and real data during train contacts. As shown, the contact point in both simulation and real data is properly matched with high accuracy (98.61 percent).

To show the ability of our proposed TCM in a different line, we also chose line T7. Line T7 is selected as the worst case scenario because with 414 trips per day (compared to other lines such as T1 with more than 1000 trips per day) has the minimum capacity of occurrence of contacts between trains. Figures 5.11 and 5.12 show the results obtained from our proposed TCM for two-trains and all-trains scenarios, respectively. The black cross line illustrates the points of contacts between trains commuting in line T7. Simulation results for line T7 show, 294 contacts occurs among running trains in this line everyday that each contact has a duration of 32.550 seconds assuming $600m$ as communication zone (Figure 5.11).
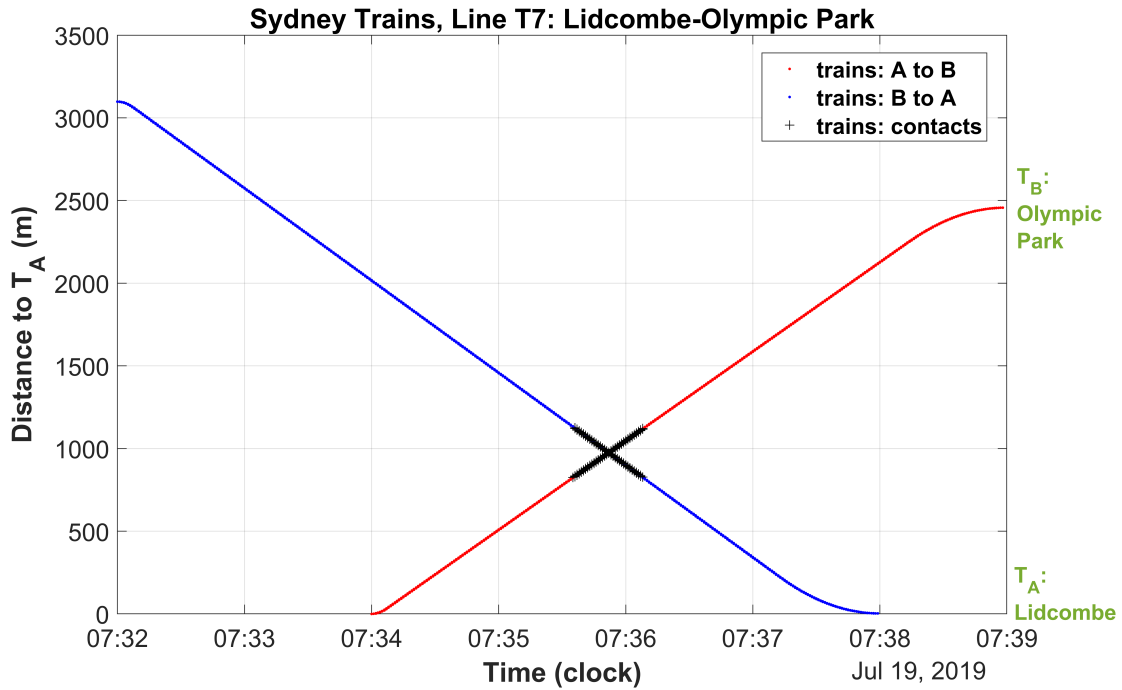
a) Distance vs. time
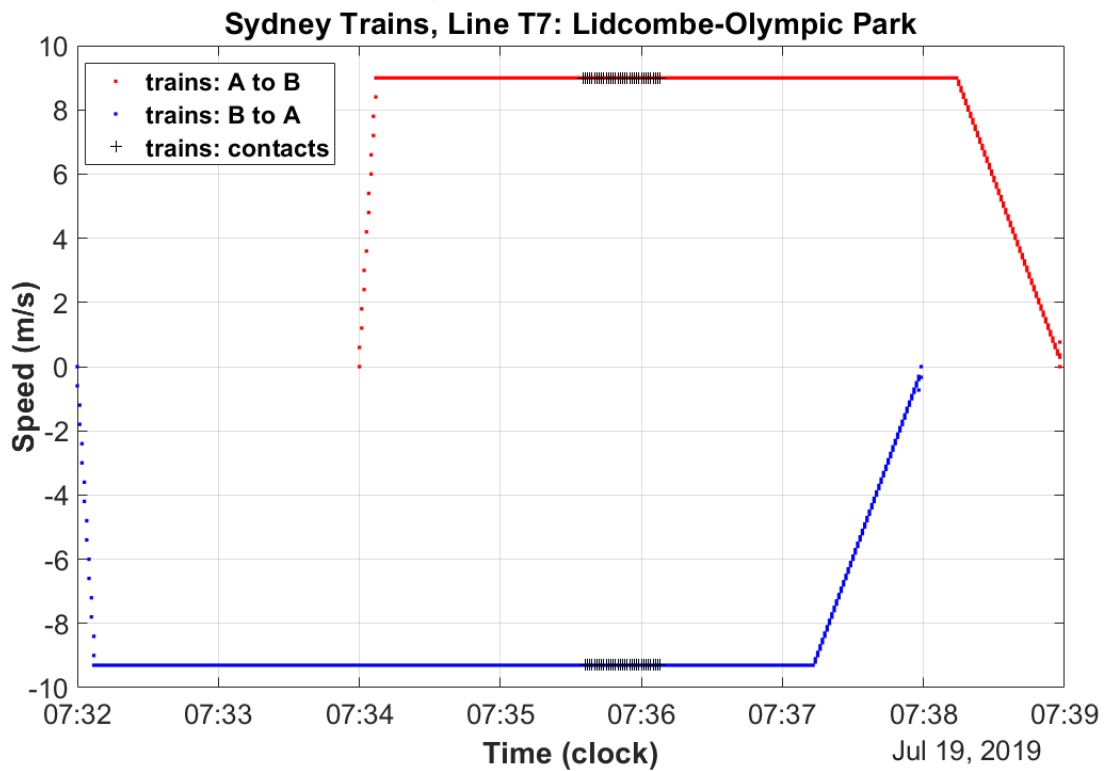
b) Speed vs. time

Figure 5.11: Results of TCM in line T7 for two-trains scenario
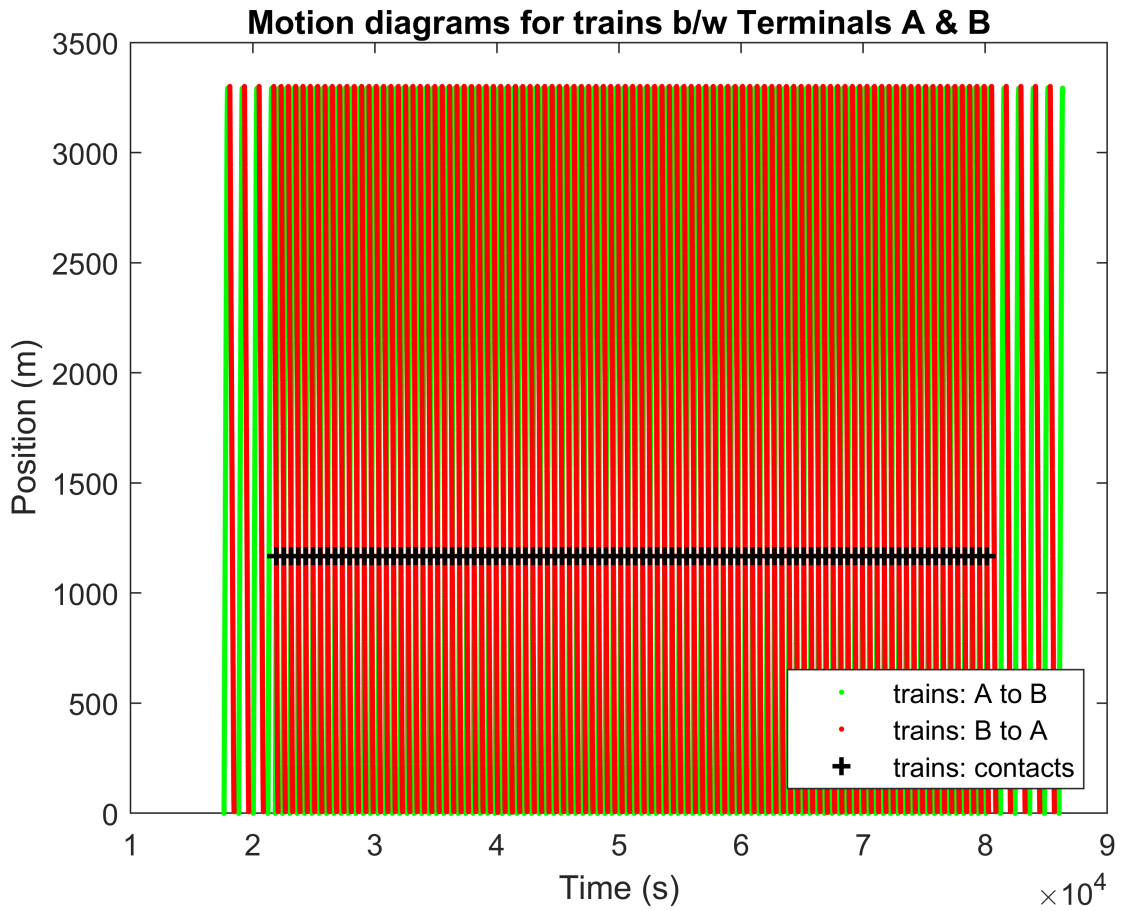
a) Distance vs. time
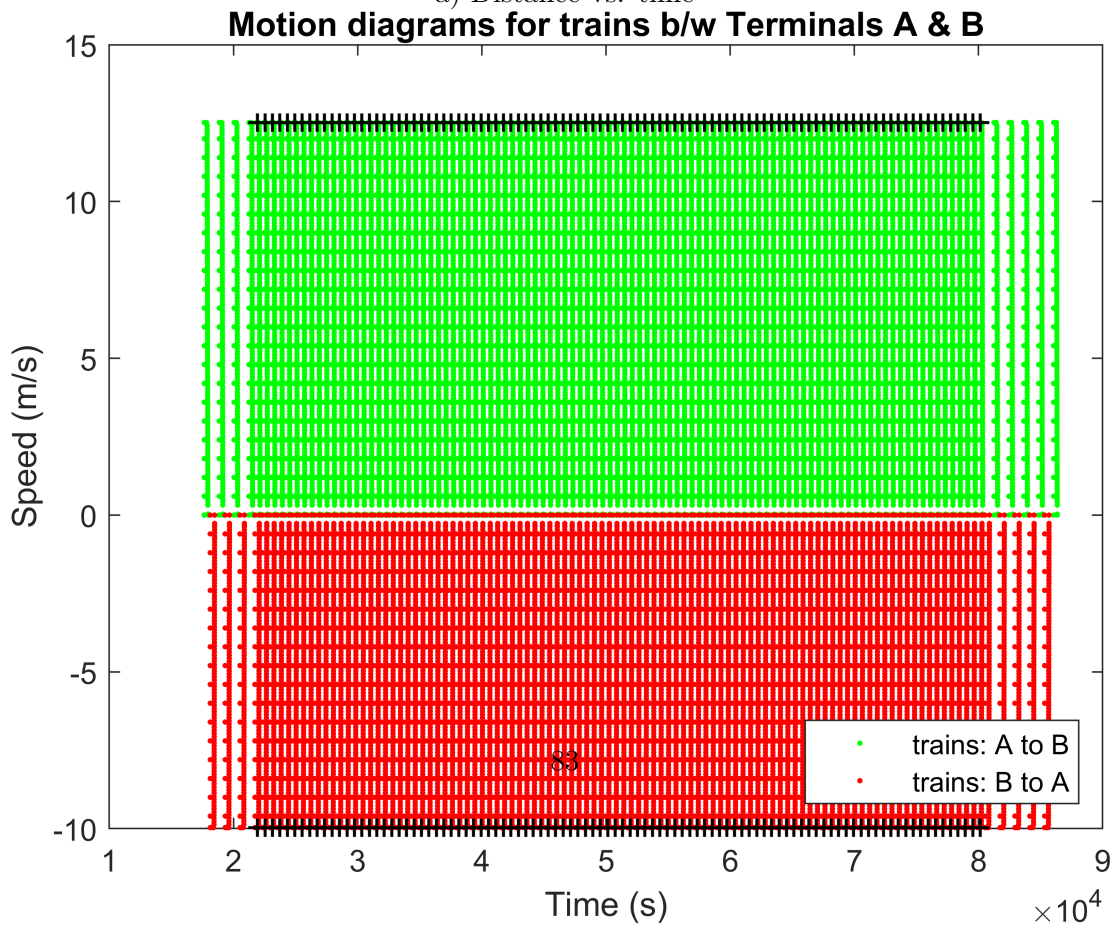
b) Speed vs. time

Figure 5.12: Results of TCM between trains in line T7 for all-trains scenario

5.3.3 T2T offloading simulation

By determining the model of contacts between trains running in a given section of a network, we can estimate the total offloading capacity of T2T communications for that section. we suppose a communication zone of 600 meter. This means that the trains will start to communicate when the distance between their antenna modules decreases to less than 600 m. If we assume that every train is equipped with multiple antennas equally distributed along the train (e.g. three antennas at first, middle and last carriages), the communication zone will include the lengths of contacting trains plus to the distance between their first carriages. According to the contact simulations in Section 5.3.2, for line T7, 294 contacts occurs among running trains everyday where each contact has a duration of 32.550 seconds assuming $600m$ as communication zone. This means that the total contact time between all the trains running in line T7 is around 3 hours during one operating day.

In order to find the possible offloading capacity for a given T2T communication in line T7, we performed a simulation between two trains a few seconds before getting contact. We used Omnet++ version 5.4.1 for this simulation. Figure 5.13 shows the scenario that we used for simulation. As shown, two trains are approaching together. The speed of trains during contacts can be any amount from zero to the maximum allowed speed (MAS). However, from Section 5.3.2, we have already found that the speeds of trains during contact is $9m/s.$. Table 5.5 shows the settings that we used for simulation of a T2T offloading network with two trains running in line T7.

Figure 5.14 shows the throughput and end-to-end delay obtained from the T2T offloading in our selected network. We simply assumed a range of $200m$ for T2T communications. However, in practice, the communication range depends on many parameters such as transmission power, path-loss model and obstacles. Generally, with some short-range communication protocols such as IEEE802.11p dedicated for VANET, we can reach up
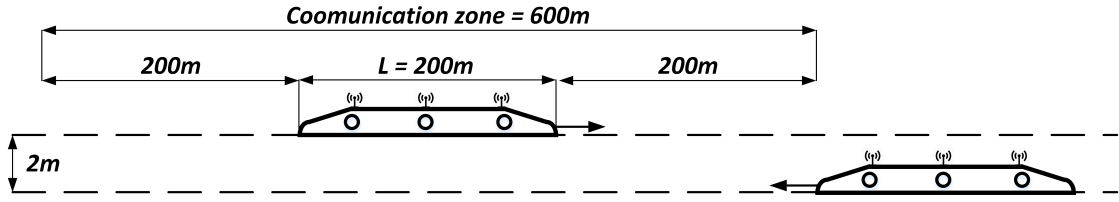
Figure 5.13: The case study used for simulation of T2T offloading

Table 5.5: Settings used for simulation of the T2T offloading network

Parameter	Setting
Communication protocol(s)	IEEE802.11p, 5.9 GHz IEEE802.11g, 5 GHz
Operational mode	Ad hoc (P2P)
Channel bandwidth	10 MHz for 11p 20 MHz for 11g
Transmission power	$20mW$
Antenna module	type: Omnidirectional length:$0.1m$ axis: z
Background noise power	$-90dBm$
Receiver sensitivity	$-85dBm$
Receiver SNIR threshold	$2dB$
Rate control algorithm	Aarf
Path-loss model	Two-Ray Interference
Transport layer	type: UDP message length: $1000B$ send interval: $0.1ms$
speed of trains during contact	$9m/s$

to $1Km$ range in line of sight (LoS). That is why, in Figure 5.14, we have still throughput even when the trains are located over $200m$ far from each other. We also used a rate control algorithm to maximize the offloading capacity. This is the reason for having variable throughput and delay as trains move.

As illustrated in Figure 5.14, we used IEEE802.11g as another WiFi-based protocol

that compared to IEEE802.11p has higher throughput. Although IEEE802.11p is the main communication standard customized for vehicular communications, it is specially equipped with more safety features appropriate for reliable transmission of critical data between close vehicles [107]. However, as described, we have already assumed that critical data has been recognized and extracted and we only have non-critical data for offloading process. Therefore, we only used IEEE802.11p for analysis purposes rather than its reliability features. As seen in Figure 5.14, the maximum throughput obtained through IEEE802.11g is around $25Mbps$ in comparison with IEEE802.11p with less than $18Mbps$. That is the same for end-to-end delay as well. While we can reach a minimum delay (end-to-end transmission delay) of around $30ms$ with IEEE802.11g protocol, the minimum delay obtained via IEEE802.11p will be a little less than $50ms$.

As described, we assumed a range of $200m$ inter-train distance to start data offloading. This assumption will have two benefits: firstly, the offloading task will be started at maximum throughput, and secondly, as a result of the first benefit, it will restrict the end-to-end delay to less than $50ms$. This ensures that the data delivery time will not exceed an allowed range that is specially important for request-based data demand.

As shown, Figure 5.14 has three time steps: approaching, passing and leaving. It takes around 10s for trains to enter in communication zone of each other. Then, it approximately takes around $11s$ per every time steps assuming $200m$ inter-train distance as well as $200m$ total length for every train. It will cause a total duration of $33s$ that confirms the contact duration already obtained from the TCM simulation in Section 5.3.2. Additionally, as the relative speed between the trains during every contact is $\Delta v = 9 + 9 = 18m/s$, the related DS is calculated from (5.14), $\Delta f = \pm354Hz$, and $\pm324Hz$ for protocols IEEE802.11p and IEEE802.11g, respectively (due to difference between the carrier frequencies). In order to find the effect of such DS on T2T offloading, we need to know how the throughput is determined in a wireless communication.

Table 5.6: The amount of FSPL with and without DS

Condition	FSPL [dB]	
	IEEE802.11p	IEEE802.11g
Without Doppler effect	47.8588	47.0896
With Doppler effect but no speed error	47.8588	47.0896
With Doppler effect and 25% speed error	47.8588	47.0896

For each communication protocol, there is a modulation and coding scheme (MCS) chart that shows the maximum data rate for every modulation index, based on the signal to noise ratio (SNR) at receiver. The received signal in dBm is obtained from the transmitter power (also in dBm) minus path-loss (PL) of the wireless channel. PL has two parts including a variable part and a fixed part. The variable part changes based on the distance between transmitter and receiver and characteristics of the wireless channel. The fixed part is obtained from the calculation of free space path-loss (FSPL) for a reference distance. FSPL in dB is calculated from $20Log(4\pi d_0 f/c)$, where d_0, f and c are the reference distance, the carrier frequency and the light speed, respectively. Therefore, assuming $d_0 = 1\,m$, the amount of FSPL for each protocol of IEEE802.11p and IEEE802.11g is shown in Table 5.6. As illustrated, the amount of FSPL is the same for the assumed conditions. This clearly confirms that mobility and speed error have no effect on the offloading throughput. The reason is that the speed of trains is extremely lower than the speed of light.

The total offloading capacity can be calculated through the discrete integral of throughput along time within contact duration as follows:

$$C_{T2T} = \sum_{t_1}^{t_2} throughput.\triangle t \tag{5.16}$$

where $t_2 - t_1 = contact\ duration$ and C_{T2T} is the total capacity of T2T offloading based

on *bps*.

From the calculations based on (5.16), we obtained 568 *megabits* and 809 *megabits* for IEEE802.11p and IEEE802.11g, respectively, for every contact in line T7. Given 294 times of contact occurrence in every operating day, we can offload up to 167 *gigabits* and 238 *gigabits* using IEEE802.11p and IEEE802.11g as communication protocols, respectively, through T2T communications in line T7.

5.4 Conclusion and Future Work

In this chapter, we proposed a train mobility model that can accurately provide train traffic traces only based on trip timetables. As timetables are currently available in real-time, the TMM can be quickly updated in near real-time. We validated our proposed TMM through extensive simulations for different lines of Sydney Trains. Our simulation results showed that the traces from TMM can accurately follow the actual GPS data with less than 2% error. Additionally, we applied energy optimization considerations in the proposed TMM. This enables the model to provide a travel guidance trajectory that causes optimal trips for trains in a rail network.

We also developed another model for T2T communications that determines duration, location and rate of T2T contacts by using traffic traces obtained from the TMM. We tried the TCM in Sydney Trains Network and showed that we can theoretically offload around 167 *gigabits* and 238 *gigabits* using IEEE802.11p and IEEE802.11g as communication protocols, respectively, through T2T communications.
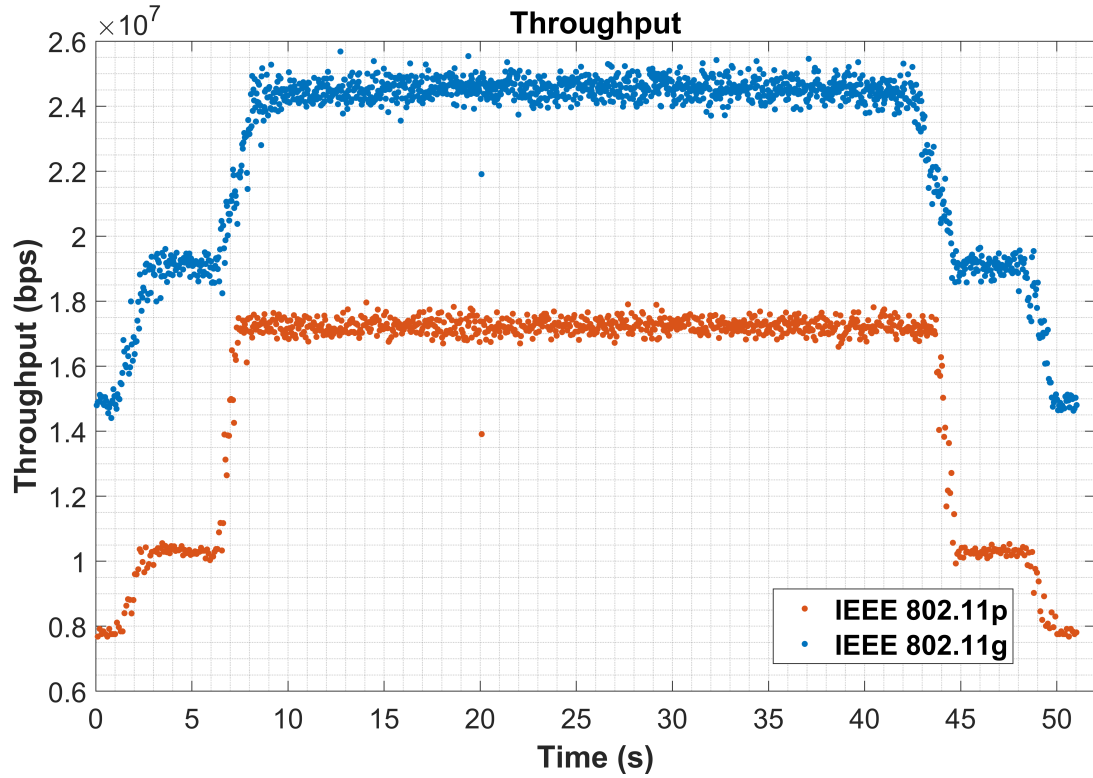
a)
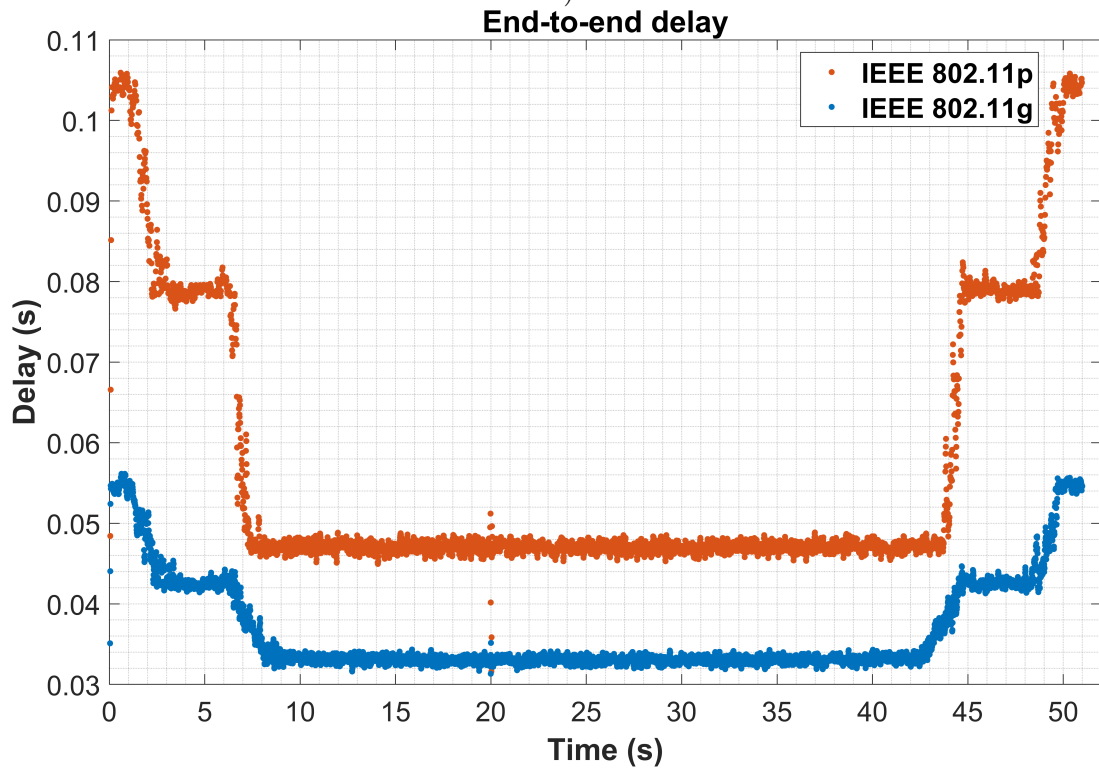
b)

Figure 5.14: T2T Offloading diagrams: a) throughput, b) end-to-end delay

6 Train-to-Wayside Communication Method

6.1 Introduction

With the advancement of data-driven intelligent transportation systems (ITS), the amount of data generated by IoT devices is significantly increasing. Therefore, it is essential to find a reliable and cost-effective communication method for exchange of such massive data between ITS and data centers. Cellular communications especially upcoming 5G as the latest revision, can be one of the main solution for such data exchange. However, due to rising public demand for cellular networks especially during peak hours, exchange of data through other communication platforms in a reliable and cost-effective manner will be highly beneficial [108].

In chapters 4 and 5, we proposed models for data transportation in rail networks through train-to-station (T2S) and train-to-train (T2T) communication methods, respectively [9] . We used T2S and T2T communication methods for transmission of delay-tolerant part of data assuming that the critical part of data had been already realized via an appropriate classification scheme [8]. Figure 6.1 shows such different communication methods in rail networks.
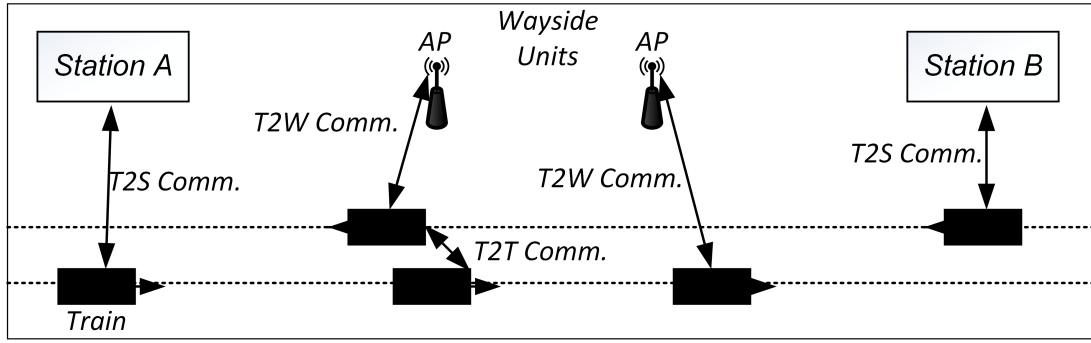
Figure 6.1: Three different short-range communication methods in rail networks

In this chapter, we enhance the previous methods by proposing train-to-wayside (T2W) communications. In this way, the data exchange process can be performed not only partially within T2S or T2T communications but also along the whole rail track and in a continuous manner. To implement such method, we assume that there are wayside units along the rail that can exchange data with trains via an appropriate wireless local area network (WLAN) like WiFi. Then, to have an efficient data exchange (which means maximum data transmission with minimum transmission energy), we develop three different algorithms to optimally place the access points (APs) of those wayside units in various conditions. All the three algorithms can accurately adapt to any rail track with different geometric paths by using the related GPS data. We also propose a method to model the change of communication characteristics when trains pass through various environment-based scenarios. Assuming several railway scenarios include urban, suburban, rural, cutting, viaduct, tunnel and river, such model cannot only simulate a rail line with single scenario but also a line with multiple scenarios. The main contributions of the current chapter are as follows:

1. The first algorithm is developed to find the minimum number of APs required for a rail network based on a desired path-loss (PL) threshold. Through this algorithm, we can also find appropriate initial places for APs based on an equally distributed

placement (EDP) strategy that is useful for quick estimations. Additionally, such initial places are needed to run the other two algorithms described in the following. The proposed EDP algorithm is also used as a baseline method for comparison purposes with next two algorithms.

2. The second algorithm is developed for optimal placement (OP) of a given number of APs so that the average PL is minimum. This algorithm can be used when we have a specific number of APs and seek the best places to implement them.

3. The third algorithm is a hybrid placement (HP) scheme that can optimally determine both the number and places of APs so that the following two conditions are concurrently met. Firstly, the PL does not exceed a desired threshold at any point. Secondly, the average PL of the network is minimum.

4. We propose a method to model the changes of communication characteristics for different scenarios in rail networks. The railway environment model (REM) proposed for wireless communications in rail networks shows how the problem of AP placement can be affected by different scenarios. Additionally, through REM, modeling the rail lines with multiple scenarios will be also possible.

The rest of the chapter is presented as follows. We review the related works in the next section. The proposed algorithms, the system models and the theoretical framework are explained in Section 6.2. We validate the performance of the models for different scenarios in Section 6.3. The conclusion of the chapter is presented in the last section.

6.2 System Models and Problem Formulation

In this section, we explain our proposed methodology and models for AP placement in rail networks. Firstly, we show the problem formulation. Then, we present EDP
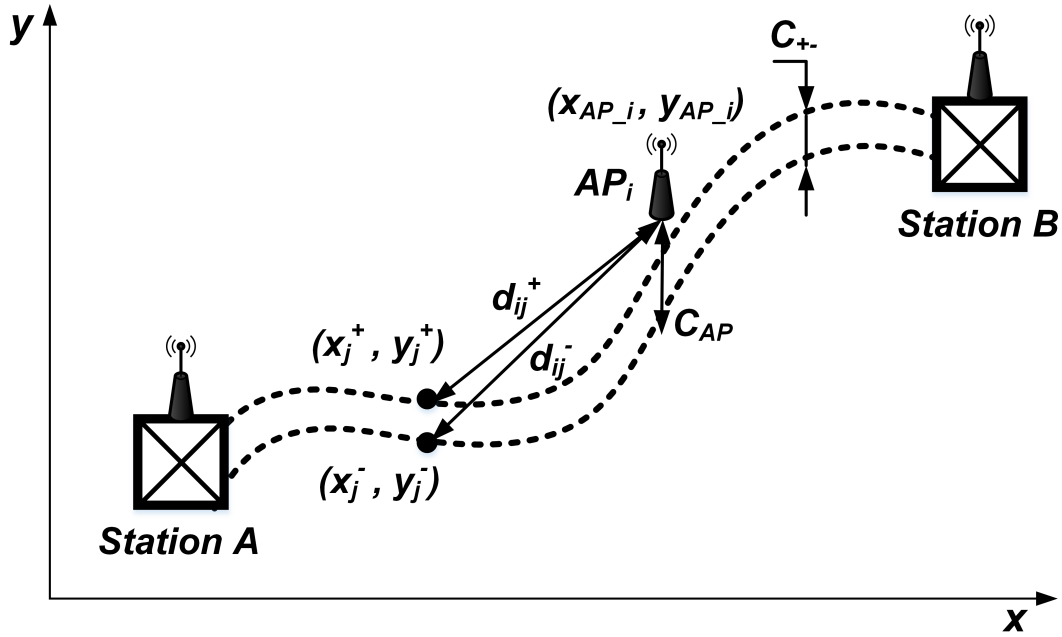
Figure 6.2: Problem overview

algorithm that can simply determine the required number and initial places of APs for a rail track. Then, we describe a method to model the environmental changes along a rail track. Finally, we present OP and HP algorithms for AP optimal placement in rail networks.

6.2.1 Problem Formulation

Assuming a rail *path*, which represents a rail section between two stations, our goal is to find the optimal places for installation of minimum number of APs along such path. Figure 6.2 illustrates such situation for a double-track railway in a two-dimensional (2-D) coordinates. As previously described, the APs are connected to wayside units to exchange IoT data with moving trains. Our objective is to achieve 1) maximum throughput with 2) minimum number of APs and 3) in a continuous manner. For throughput, our focus is on the main data payload rather than frame overheads and

inter-frame intervals as these parameters are highly dependent on the wireless network protocol. A key factor that affects the throughput is the power of wireless signal at receiver called received signal strengths (RSS). To achieve the maximum throughput, RSS should be maximized or PL should be minimized. On this basis, we define an optimization function aimed at minimizing the average PL of the network. We also aim to use the minimum number of APs to reduce the system cost. The number of APs is determined based on the minimum throughput that we expect from the given network. To obtain a desired throughput, for a specific transmitter power, PL does not exceed a defined threshold at any point of the network. To have continuous data exchange, we can simply set such threshold to the minimum sensitivity of the receivers in the network.

For the communication channel propagation, we use the most cited log-normal shadowing model that is an effective model for network design purposes across various environments [109]. Generally, linear PL of a wireless channel is defined as the ratio of transmitter power, p_{Tx}, to receiver power, p_{Rx}, [109]. Equation (6.1) shows the PL for a channel between a transmitter at point i and a receiver at point j. PL can be also defined in dB as shown in Equation (6.2) and is determined by (6.20).

$$pl_{ij} = \frac{p_{Tx_i}}{p_{Rx_j}} \quad [mW/mW] \tag{6.1}$$

$$PL_{ij} = 10log_{10}(pl_{ij}) \quad [dB] \tag{6.2}$$

$$PL_{ij}^{+(-)} = PL_0 + 10\gamma log_{10}(d_{ij}^{+(-)}/d_0) + X_\sigma - G_a, \ d_{ij}^{+(-)} \geq d_0 \tag{6.3}$$

$$d_{ij}^{+(-)} = \sqrt{[(x_{AP_i} - x_j^{+(-)})^2 + (y_{AP_i} - y_j^{+(-)})^2 + +(z_{AP_i} - z_j)^2]} \tag{6.4}$$

Where, $i = \{1, 2, ..., n_{AP}\}$ and $j = \{1, 2, ..., n_{AB}\}$. In (6.20), γ is the PL exponent (PLE) and X_σ is a random variable with normal distribution, zero mean and standard deviation (STD) of σ and is used to model the shadow fading (SF) in a network. The values of PLE and STD are different across various environments. $d_{ij}^{+(-)}$ is the euclidean distance between i^{th} AP and j^{th} point of the given rail path and is represented in (6.4) for a three-dimensional (3-D) network. Positive and negative signs at all equations are related to the lines from stations A to B and B to A, respectively. In practice, the antenna heights are selected between 0.5-1 meter higher than the train roof ensuring that the wireless signals are not affected by the train body [86]. However, for simplification, we assume that APs antennas are installed at the same height of train antennas. This is just a simple assumption to reduce the dimension of the problem from 3-D to 2-D and does not affect the total design of the network. We also suppose Omni-directional antennas and the same transmission power is set for all transceivers in the given network. We further assume that adjacent APs use different communication channels through an appropriate channel assignment scheme to avoid interference between APs.

In (6.20), PL_{ij}^+ and PL_{ij}^- are the PLs of the wireless links between AP_i and j^{th} point of rail track at each line from stations A to B (L_{AB}) and B-to-A (L_{BA}), respectively. n_{AP} and n_{AB} are the numbers of required APs and the sample points of the path between stations A and B, respectively. $G_a = G_{Tx} + G_{Rx}$ is the total antenna gains of transmitter and receiver. PL_0 is the amount of PL in a reference distance and can be obtained from the free-space model as (6.5) [109]:

$$PL_0 = 20log_{10}(4\pi d_0/\lambda) \tag{6.5}$$

In (6.5), λ is the wavelength in meter and is obtained from $\lambda = c/f_c$ (c is the speed of light and f_c is the radio carrier frequency). (x_j^+, y_j^+) and (x_j^-, y_j^-) are 2-D Cartesian

coordinates of path sample points for L_{AB} and L_{BA}, respectively. z_j is the coordinate of train antenna height along z-axis, which is the same at both lines. Assuming similar geometric paths for lines L_{AB} and L_{BA}, following relations can be applied between every two corresponding points of L_{AB} and L_{BA} (details for double lines with different geometric paths are explained in the following).

$$x_j^+ = x_j^- \tag{6.6}$$

$$y_j^+ = y_j^- + C_\pm \tag{6.7}$$

In (6.7), $C_\pm$ is the clearance between L_{AB} and L_{BA}. We also define C_{AP} as the minimum clearance between APs and rail track required for practical issues such as collision avoidance with trains. Assuming that the distance between APs and rail track across y-axis is set to C_{AP}, the following relation is obtained.

$$y_{AP_i} = \begin{cases} y_{AP_i}^+ = F(x_{AP_i}) + C_\pm + C_{AP} \\ \\ \qquad\qquad or, \\ \\ y_{AP_i}^- = F(x_{AP_i}) - C_{AP} \end{cases} \tag{6.8}$$

In (6.8), y_{AP_i} is the coordinate of APs across y-axis and $y_{AP_i}^+$ and $y_{AP_i}^-$ means that APs can be installed in every side of rail line. The most optimal side for installation of APs will be determined through the proposed placement algorithms. F is a function so that $F : x \longmapsto y_j^- \forall j \epsilon \{1, 2, ..., n_{AB}\}$ and is determined by performing grid-based linear interpolation technique on the GPS data set collected from the given rail line.

(6.6)-(6.8) were all obtained assuming similar paths for round trips, which is almost

the same for most lines. However, there are some lines that have different paths for round trips due to natural or economic constraints. An example of such line is Line T7-OlympicPark in Sydney Trains, Australia with two different round paths in a loop shape. The methodology for those lines are the same and the only difference is that the placement of APs should be independently considered for each of L_{AB} and L_{BA} as represented in (6.9) and (6.10). The rest of the methodology will be the same.

$$y_{AP_{i+}}^{+} = F^{+}(x_{AP_{i+}}) + C_{AP}, \ F^{+} : x_j^{+} \longmapsto y_j^{+} \forall j \epsilon \{1, 2, ..., n_{AB}\} \tag{6.9}$$

$$y_{AP_{i-}}^{-} = F^{-}(x_{AP_{i-}}) - C_{AP}, \ F^{-} : x_j^{-} \longmapsto y_j^{-} \forall j \epsilon \{1, 2, ..., n_{AB}\} \tag{6.10}$$

Where $\{F^{+}, \ i+, \ n_{AP}^{+}\}$ and $\{F^{-}, \ i-, \ n_{AP}^{-}\}$ are the interpolation function, APs index and number for each line of L_{AB} and L_{BA}, respectively.

In order to find PL at any point of a line, we calculate the amounts of PL from all the existing APs through (6.20). Then, the minimum amount is selected as the final PL of such point as represented in (6.11).

$$PL_j^{+(-)} = \min_{i} \{PL_{ij}^{+(-)}\},$$

$$\forall i \epsilon \{1, 2, ..., n_{AP}\} \ and \ j \epsilon \{1, 2, ..., n_{AB}\} \tag{6.11}$$

Therefore, the total average PL at both lines L_{AB} and L_{BA} is obtained in (6.12).

$$PL_{av}^{total} = \frac{1}{2n_{AB}} \sum_{j=1}^{n_{AB}} (PL_j^{+} + PL_j^{-}) \tag{6.12}$$

To achieve maximum throughput, the average PL should be minimized. Therefore, we define an optimization problem in (6.13) to find the minimum PL.

$$\begin{cases} \forall i\epsilon\{1,2,...,n_{AP}\}, & and \ j\epsilon\{1,2,...,n_{AB}\}: \\ minFunction: & \min_{x_{AP_i},y_{AP_i},n_{AP}} \{PL_{av}^{total}\}, \\ 1^{st}\,constraint: & PL_j^{+(-)} \leq PL_{th}, \\ 2^{nd}\,constraint: & min\{x_j\} \leq x_{AP_i} \leq max\{x_j\}, \\ 3^{rd}\,constraint: & y_{AP_i} = \{y_{AP_i}^{+} \ or \ y_{AP_i}^{-}\} \end{cases} \quad (6.13)$$

(6.13) shows a constrained nonlinear problem. The first constraint indicates that the amount of PL must not exceed a desired threshold (PL_{th}) at any point. As previously described, this constraint ensures the continuity of data exchange when PL_{th} is set to the minimum receiver sensitivity of the network. The second constraint shows the limits of APs places across x-axis, which equals to the beginning and end point of the line. The last constraint shows the possible places of APs at every side of the line as defined in (6.8).

To solve the problem defined in (6.13), we use sequential quadratic programming (SQP), which is an appropriate method for constrained nonlinear optimization [110] and [111].

6.2.2 Initial Placement Approaches

Based on the availability of WLANs in the stations at every side of a rail path, three different approaches can be assumed for the placement of APs as follows.

- *Approach 1 (A1)* for the paths that WLANs are already available at the stations at both ends as shown in Figure 6.2.

- *Approach 2 (A2)* for the paths that a WLAN is available only at one side.

- *Approach 3 (A3)* for the paths that there is no WLAN available in the stations at any side.

Such classification is essential as we must consider the effects of those APs that are already available in stations. Therefore, we propose the following scheme for initial placement of APs.

- for A1, we set the initial number of APs to two, i.e. $n_{AP}^{initial} = 2$, and those two APs are assigned to the stations at both sides. The places of the rest of APs are determined through the proposed algorithms described in sections 6.2.3-6.2.5.

- for A2, we set the initial number of APs to one and that AP is constantly placed at the station with WLAN. The next APs are placed via the placement algorithms.

- for A3, the initial number of APs is set to zero and all APs are placed via the placement algorithms.

On this basis, the proposed algorithms determine the total number of APs including the ones that are already available at stations as well. Therefore, the required number of APs needed to be added in a given network is obtained by subtracting two and one from the final n_{AP} for *A1* and *A2*, respectively. The required number for A3 is equal to the final n_{AP} as no AP is already available at any station.

6.2.3 Equally Distributed Placement (EDP) algorithm

The purpose of EDP algorithm is firstly to find the minimum required number of APs based on a desired threshold. Secondly, we determine the initial places of APs through a simple algorithm working based on equal distances between APs. Such initial places are essential for running OP and HP algorithms. Additionally, EDP algorithm can be used as a quick basic method for the approximate placement of APs . Although,

EDP algorithm is a simple method, it provides an effective baseline for evaluation of performances of OP and HP algorithms.

The basic parameter needed in EDP algorithm is the total length of the given rail line. As rail lines are not straight and have several curves, we cannot simply calculate the lengths of lines via measuring the distances between start and end points of the lines. Instead, we should use the following recursive equation to accurately find the length of a given rail line (6.14).

$$l_{k+1} = l_k + \sqrt{(x_{k+1} - x_k)^2 + (y_{k+1} - y_k)^2} \tag{6.14}$$

In (6.14), $k = \{0, 1, 2, ..., n_{AB} - 1\}$ and l_k is the line length from beginning to the kth point with Cartesian coordinates of (x_k, y_k). Therefore, the total length of the rail line is obtained from (6.14) when $k = n_{AB} - 1$.

Figure 6.3 shows the flowchart of EDP algorithm. The inputs of EDP algorithm include as follows:

- GPS data samples related to the location points of the given rail tracks. The GPS data is internally converted to Cartesian coordinates in meters.

- Network clearances include $C_\pm$ and C_{AP}.

- Specifications of REM that are explained in Section 6.2.6.

- Initial number of APs based on the type of initial placement approach.

As shown in Figure 6.3, assuming an A1 situation, one AP is assigned to the station at each end of the path. Then, the rest of APs are equally placed along the path. Next, PL is calculated at every point and is compared with the desired PL threshold. If PL

Inputs
- **GPS data of rail line**
- **Clearances**
- **PL threshold**
- **ECA and ECR**
- **Initial n_{AP}**

Assigning one AP to each station

Equal placement of the rest of APs

$n_{AP} = n_{AP} + 1$

Check If $PL_j^{+(-)} > PL_{th}$

Calculating PL

Yes

No

Outputs
- **Number of APs**
- **Initial places of APs**
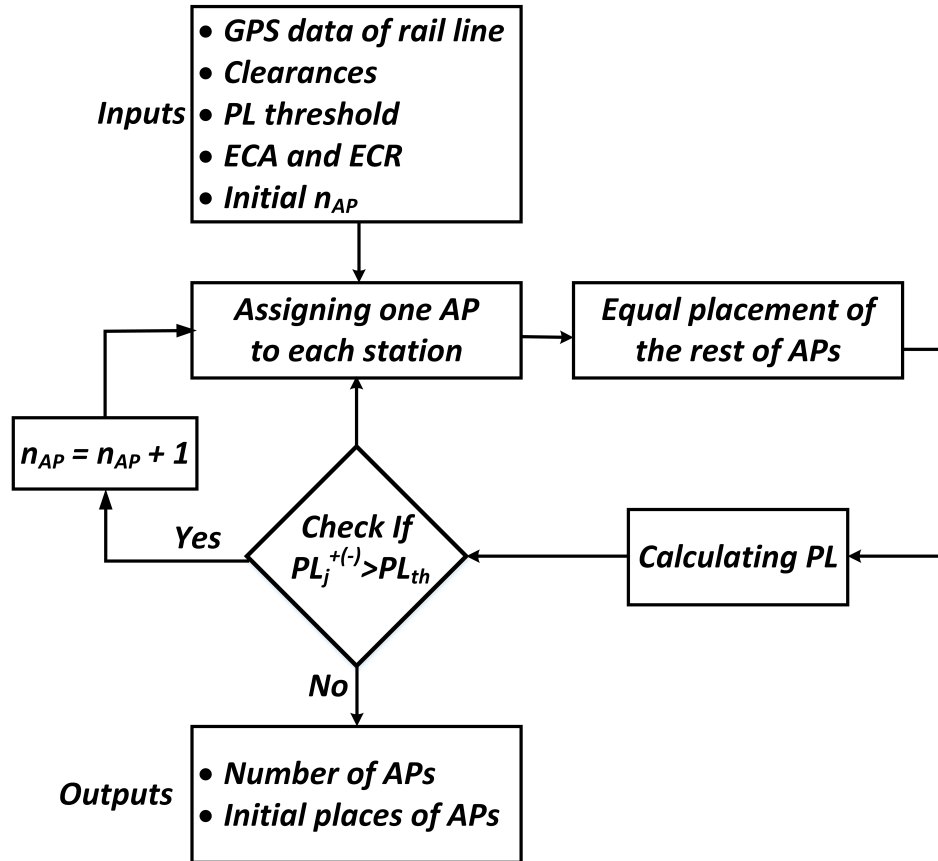
Figure 6.3: EDP algorithm

exceeds the threshold at any point, another AP is added to the network and the previous steps are repeated. Otherwise, the algorithm returns the final number and places of APs.

6.2.4 Optimal Placement (OP) Algorithm

This algorithm is proposed to determine the optimal places for a given number of APs in a rail network. Assuming a known number of APs, the initial places of those APs is proposed by EDP algorithm. Then, the PL at every point and the total average PL is calculated from (6.11) and (6.12), respectively. Finally, the optimal places of APs are determined by solving the optimization problem defined in (6.13). To solve (6.13) in OP
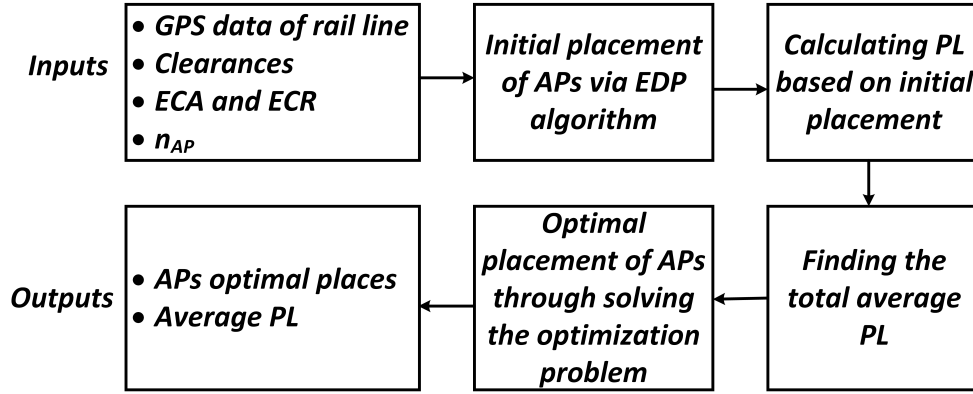
Figure 6.4: OP algorithm

algorithm, the first constraint is not considered as such constraint is used to find the required number of APs in a network. However, OP algorithm is used when the number of APs is known and we seek the most optimal places for those APs. Figure 6.4 shows the flowchart of OP algorithm.

6.2.5 Hybrid Placement (HP) Algorithm

This is a comprehensive algorithm for OPAPs in a rail network. As shown in Figure 6.5, firstly, an initial number of APs is temporarily placed via EDP algorithm. Then, instantaneous and average values of PL is calculated. Next, the optimal placement of APs is performed via the Optimization problem defined in (6.13). Then, PL is calculated for the second time but based on new optimal places of APs. If the PL at any point exceeds the threshold, another AP is added and all the steps will be repeated until PL does not exceed the threshold at any point.
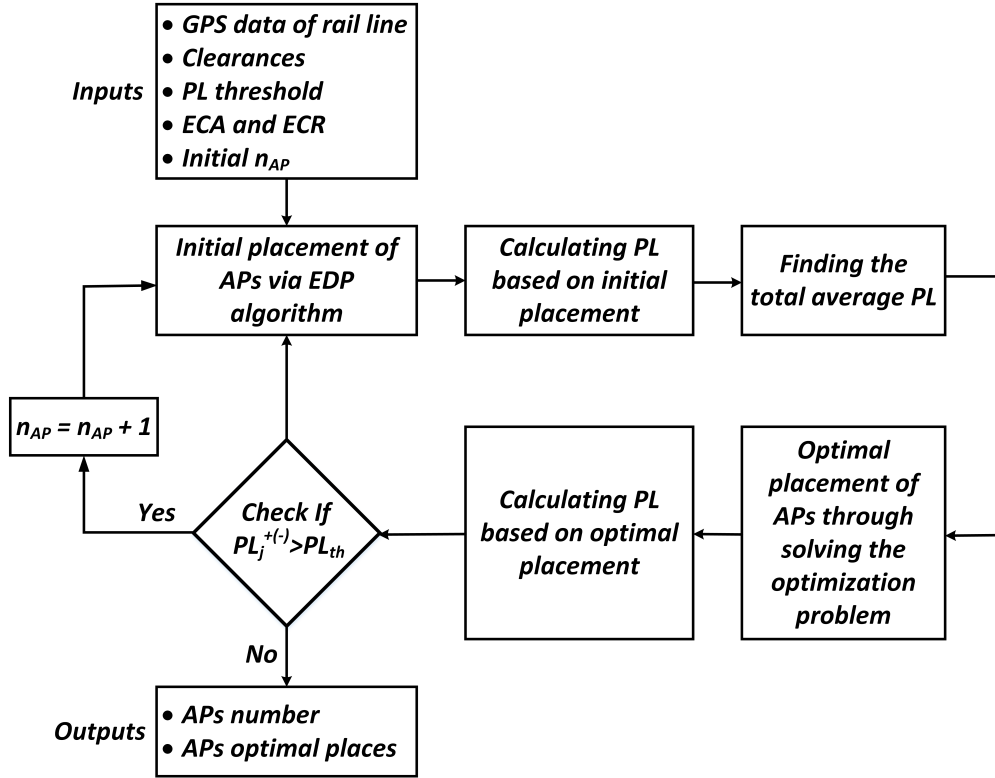
Figure 6.5: HP algorithm

6.2.6 Railway Environment Model (REM) for Wireless Communications

For wireless communications in rail networks, there are several environments such as viaducts, cuttings and tunnels [112]. As wireless signals propagate differently in every environment, all the environments that form the given rail path should be considered in the placement of APs.

In [113] the authors generally classified the wireless environments based on the different ranges of PLE and STD collected through extensive experiments in different sites. We use such concept but specifically in railway networks with 8 different environments include urban, suburban, rural, viaduct, cutting, station, tunnel and river [27]. On this basis, we define 8 environment classes (ECs)in (6.15), where γ_k^{min}, γ_k^{max}, σ_k^{min} and σ_k^{max} are the lower and upper limits of PLE and STD for EC_k, respectively. Table 6.1 shows the

Table 6.1: PLE and STD for different environments in rail networks

Class No.	Environment	PLE	STD(dB)
1	urban	4-7	3-5
2	suburban	3-5	2-3
3	rural	2-5	2-3
4	viaduct	2-4	2-4
5	cutting	2.5-4	3-5
6	tunnel	1.8-3	5-8
7	river	2-4	2-3
8	station	3-5	3-5

ranges of PLE and STD for every EC in a rail network [112].

$$EC_k : \begin{cases} \gamma_k^{min} \leq \gamma_k < \gamma_k^{max} \\ \sigma_k^{min} \leq \sigma_k < \sigma_k^{max} \end{cases} \quad \forall k \epsilon \{1, 2, ..., 8\} \tag{6.15}$$

Since a train may pass through a variety of environments, the total REM for a rail line can be a combination of different ECs. Therefore, an REM can be generally defined as (6.16) for a given rail path.

$$REM : \bigcup_{k=1}^{n_e} \alpha_k EC_k, \ \forall k \epsilon \{1, 2, ..., 8\} \tag{6.16}$$

In (6.16), $\alpha_k \epsilon [0, 1]$ is the ratio of EC_k to the total REM so that $\sum_{k=1}^{n_e} \alpha_k = 1$. This ratio for every EC can be simply obtained through dividing the length of line with that EC by the total line length. For any EC that $\alpha = 0$, it means that such EC does not exist in the path. However, $\alpha_k = 1$ shows that there is only one EC_k along the entire path.
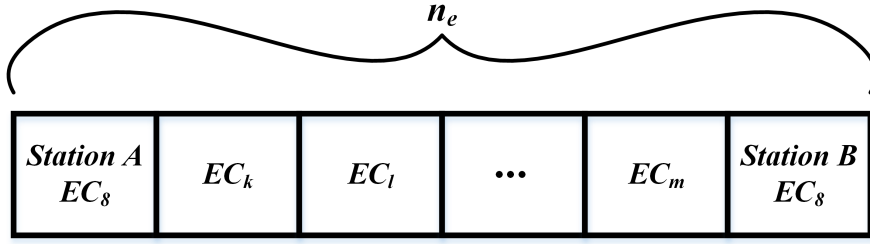
Figure 6.6: The proposed REM for a given rail path

n_e is the total number of ECs that exist along the path. A given path may also have similar ECs but at different sections of the path. An example can be a path that has two tunnels at different parts of that path.

Figure 6.6 shows our proposed diagram of an REM for a rail path. In Figure 6.6, k, l and m are indices from $1, 2, .., 7$ and EC_8 refers to the EC at stations as defined in Table 6.1. As shown, an REM for a given path is defined by two vectors: 1) the ratio of every EC to the total length of the path, called ECR (environment class ratio), and 2) the arrangement of ECs existed in the REM and is named ECA (environment class arrangement). ECR and ECA are represented in (6.17) and (6.18).

$$ECR = [\; \alpha_8 \quad \alpha_k \quad \alpha_l \quad ... \quad \alpha_m \quad \alpha_8 \;]_{(1 \times n_e)} \tag{6.17}$$

$$ECA = [\; EC_8 \quad EC_k \quad EC_l \quad ... \quad EC_m \quad EC_8 \;]_{(1 \times n_e)} \tag{6.18}$$

6.3 Simulation and Results

To validate the proposed placement algorithms, we select Line T6, between Clyde-Rosehill from Sydney Trains of Australia, with the total length of 1717 meter. For

the required position data related to Line T6, we use the dataset that we collected by a GPS module through extensive site measurements. The sampling frequency of the GPS module was 10 Hz. In all experiments, we assume that every train is equipped with two omni-deirectional antennas, one at each end to provide full coverage along the whole length of the train [86]. Furthermore, for data transmission, we used IEEE802.11-based WiFi in two different communication modes. Mode 1 is the basic mode with the lowest data rate in IEEE802.11n series and Mode 2 with 5 GHz frequency and 160 MHz bandwidth has the highest data rate in IEEE802.11ac series. In fact, we choose these two modes as the lowest and highest bounds of current IEEE802.11n(ac)-based WiFi protocols. Although, there are other new coming WiFi protocols with much higher throughput such as IEEE802.11ay, our goal is to show the impact of the proposed algorithms on the data transmission and the amount of data rate is not interested in the current chapter. Table 6.2 shows the list of all settings used in our experiments.

6.3.1 REM Selection Strategy in Simulations

Based on the definition provided in Section6.2.6, theoretically, the total number of different cases for REM is infinite. Although, there is no limit for the proposed AP placement algorithms in terms of REM selection, it is practically impossible to test all the cases. Therefore, we have to use a strategy to limit the test cases. As previously stated, a train passes through various environments and therefore, considering a single-environment model does not give the right results. To show this, we consider 9 different scenarios include 8 single-environment scenarios and one multi-environment scenario. Each single scenario contains one of the ECs shown in Table 6.1. However, the multiple scenario contains all the ECs illustrated in Table 6.1. In this way, it can be seen how different the results will be if a multi-environment scenario is assumed as a single-environment scenario. On this basis, the total ECA and the specific ECR related to every scenario

Table 6.2: Simulation settings

Parameter	Setting
Communication modes	Mode 1 protocol: IEEE802.11n frequency: 2.4 GHz bandwidth: 20 MHz NSS: 1 GI: 800 ns Mode 2 protocol: IEEE802.11ac frequency: 5 GHz bandwidth: 160 MHz NSS: 3 GI: 400 ns
Transmission power	$20mW - 50mW$
Antenna modules (Tx/Rx)	type: Omni-directional gain: $15\,dB$ axis: z
Background noise power	$-90\,dBm$
Receiver sensitivity	$-85\,dBm$
Receiver SNIR threshold	$2\,dB$
ρ	0.4
Rate control algorithm	Aarf
Transport layer	type: UDP message length: $1000B$ send interval: $0.1ms$
Speed of train	$100\ Km/h - 300\ Km/h$
Sampling frequency	$10\,Hz$
$C_\pm$	$2\,m$
C_{AP}	$1\,m$
PL_{th}	$100\,dB$
d_0	$1\,m$

EC_8	EC_1	EC_2	EC_3	EC_4	EC_5	EC_6	EC_7	EC_8

Figure 6.7: The proposed REM for evaluation of the proposed placement algorithms

are defined as shown in Figure 6.7 and Table 6.3, respectively.

6.3.2 Efficiency

As in this chapter, the application of AP placement is in data transportation (DT) via T2W communications in rail networks, we define the ratio of transmitted data to the consumed energy as the efficiency of the proposed algorithms [56] and [59]:

$$eff = \frac{D}{E} \tag{6.19}$$

In (6.19), *eff* is the efficiency based on bit per joule (bit/J), D is the amount of transmitted data and E is the energy consumed for transmission of such amount of data during a trip. As EDP algorithm is a baseline algorithm, for every scenario, we show the effectiveness of OP and HP algorithms in comparison with EDP algorithm. Additionally, to find the amount of improvement of system efficiency through OP and HP algorithms over EDP algorithm, we use the following formula.

$$improvement = (\frac{eff_{OP}(eff_{HP})}{eff_{EDP}} - 1) \times 100 \ [\%] \tag{6.20}$$

By calculating (6.20), we can easily find how many percent the efficiency of the network is increased when using OP or HP algorithms compared to EDP algorithm.

Table 6.3: The scenarios proposed for evaluation of the placement algorithms

Scenario No.	$ECA =$ $[8\,1\,2\,3\,4\,5\,6\,7\,8]$	REM
	ECR	
1	$[0\,1\,0\,0\,0\,0\,0\,0\,0]$	Single-environment, EC1
2	$[0\,0\,1\,0\,0\,0\,0\,0\,0]$	Single-environment, EC2
3	$[0\,0\,0\,1\,0\,0\,0\,0\,0]$	Single-environment, EC3
4	$[0\,0\,0\,0\,1\,0\,0\,0\,0]$	Single-environment, EC4
5	$[0\,0\,0\,0\,0\,1\,0\,0\,0]$	Single-environment, EC5
6	$[0\,0\,0\,0\,0\,0\,1\,0\,0]$	Single-environment, EC6
7	$[0\,0\,0\,0\,0\,0\,0\,1\,0]$	Single-environment, EC7
8	$[0\,0\,0\,0\,0\,0\,0\,0\,1]$	Single-environment, EC8
9	$[0.01\,0.14\,...\,0.14\,0.01]$	Multi-environment, all ECs

In order to find the efficiency defined in (6.19), we need to calculate the amount of energy required for transmission of a given volume of data. Assuming a desired throughput, the amount of transmitted data is simply obtained as follows:

$$D = thr.T \tag{6.21}$$

In (6.21), thr is the desired throughput and T is the duration of a trip. Due to have data overhead and inter-frame intervals that highly depends on the wireless MAC protocol, the actual throughput of a wireless channel is always less than the maximum data rate mentioned in the related standards. Therefore, throughput can be obtained as a ratio of maximum data rate, named MAC efficiency factor. This is represented in (6.22), where ρ is the MAC efficiency factor and r is the maximum data rate of the wireless channel [9].

$$thr = \rho.r \tag{6.22}$$

Additionally, the required energy can be obtained as (6.23), where $p_{Tx}(t)$ and $\overline{p_{Tx}}$ is the instantaneous and average amounts of transmission power.

$$E = \int_0^T p_{Tx}(t).dt \simeq \overline{p_{Tx}}.T \tag{6.23}$$

As the data rate of a wireless channel depends on the level of signal to noise ratio (SNR), for a given data rate, we firstly find $P_{Tx}(t)$ in dBm from (6.24), where P_{Rx} and P_N are powers of received signal and noise at receiver in dBm, respectively.

$$SNR^{(dBm)} = P_{Rx} - P_N \tag{6.24}$$

From (6.1) and (6.2),

$$P_{Rx} = P_{Tx} - PL \tag{6.25}$$

Then, for a given SNR, the average transmitter power in dBm, $\overline{P_{Tx}}$, can be obtained from (6.24) and (6.25) as follows, where PL_{av}^{total} was defined in (6.12):

$$\overline{P_{Tx}} = PL_{av}^{total} + P_N + SNR^{(dBm)} \; [dBm] \tag{6.26}$$

Finally, $\overline{p_{Tx}}$ based on watt (W) can be simply obtained from (6.27).

$$\overline{p_{Tx}} = 10^{(\frac{\overline{P_{Tx}}}{10} - 3)} \; [W] \tag{6.27}$$

6.3.3 EDP Algorithm

As described in Section 6.2.3, the required quantity as well as the initial places of APs can be determined by EDP algorithm so that PL does not exceed a threshold at any point. Through sections 6.2.4 and 6.2.5, we use the initial placement obtained by EDP algorithm as the situation of the network before optimization to show the efficiency of OP and HP algorithms. However, the ability of EDP algorithm to find the required number of APs based on a desired threshold needs to be independently verified, which is shown in this section.

Figure 6.8 shows the total number of APs required for every scenario that is obtained by EDP algorithm based on $PL_{th} = 100 \, dB$. The results shown in Figure 6.8 are the total number of APs, i.e. n_{AP}, that is required to be added to an A3-based path.

However, the number of extra APs required to be added to a path in A1 and A2 are $(n_{AP} - 2)$ and $(n_{AP} - 1)$, respectively. Since the values of PLE and SF randomly change within the ranges, we showed the minimum, average and maximum number of APs for every scenario (which are obtained for the lower, middle and upper values of the ranges, respectively). Assuming that the changes of PLE and SF at every scenario can be modeled by an appropriate normal distribution, we can restrict the PL of the network to the desired threshold with a probability of more than 50% if the average number is selected. By choosing the maximum number, the threshold can be guaranteed for all the possible values of PLE and SF within the range but at a higher cost. This is because the extra APs, which should be added to the network compared to the average number, will be only used to achieve the threshold condition for the values of PLE and SF with less than 50% probability. Additionally, selection of the minimum number of APs at every scenario will be also the worst choice as the threshold condition will not be met for the most possible values of PLE and SF with the probability of over 90%. Therefore, in a network design, if the cost of extra APs overcome the possible breach of the threshold, choosing the average number can be the most efficient solution. Otherwise, the maximum number can be selected as the required number of APs.

As illustrated in Figure 6.8, the maximum number of APs required for the first scenario is 39, which belongs to a rail path in an urban area. This happened because the values of PLE and SF are the highest in an urban scenario among other scenarios. However, the maximum number of APs needed for scenario 6, which is related to tunnels, is only three. This is also quite predictable for tunnels that have the lowest attenuation among other scenarios, as shown in Table 6.1.

We also show the maximum values of PL for every scenario in Table 6.4. As can be seen, in all the scenarios,the amount of PL does not exceed the threshold. This clearly confirms the ability of EDP algorithm to limit the PL of the network to the desired
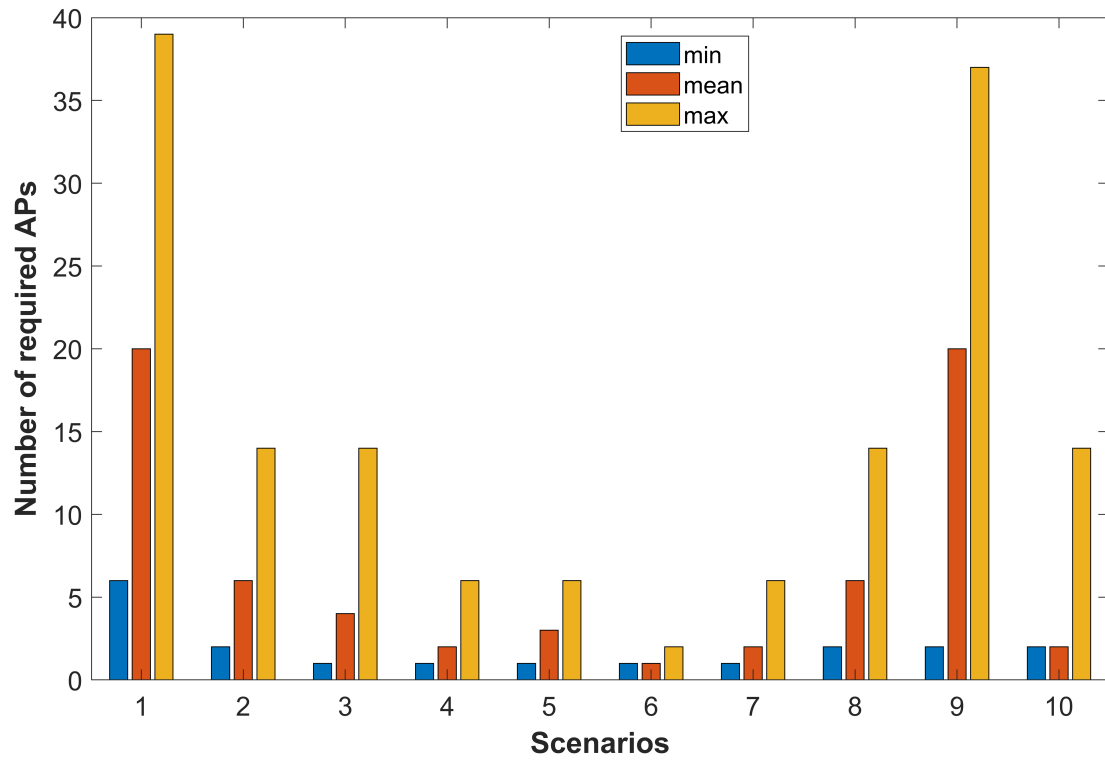
Figure 6.8: APs required number for every scenario obtained by EDP algorithm

Table 6.4: Maximum PL obtained by EDP algorithm at every scenario

Scenario no.	1	2	3	4	5	6	7	8	9
Maximum PL [dBm]	99	99	95	96	95	86	96	99	99

threshold.

6.3.4 OP Algorithm

As described in Section 6.3.4, OP algorithm finds the most optimal places for a desired number of APs so that the average PL is minimum in a given rail line. In this section, we evaluate the performance of OP algorithm for different number of APs within the range of 3-6 number for all the scenarios. For comparison, we also show the results of EDP algorithm in the same condition for every scenario. Figure 6.9 shows the statistics of the average PL for both OP and EDP algorithms for scenarios 1 and 9 assuming that 4 APs are available to be placed in the network. Due to similarity, statistics related to other scenarios are ignored. As illustrated, the distributions of the average PL obtained by OP algorithm have been shifted to the left compared to the results of EDP algorithm. This confirms the effectiveness of OP algorithm in reducing PL of the network. Although, due to similarity, we do not show such statistics for other number of APs in the range of 3-6, the efficiency of the algorithms will be represented for all the available AP numbers within the range. For this purpose, we calculate the efficiency defined in (6.19)-(6.23) for both EDP and OP algorithm for different AP numbers, two different communication modes and for all the scenarios. Table 6.5 shows such efficiencies in a comparative manner. As shown, by using OP algorithm for AP placement, the efficiency of the network will be improved at least 31% and up to 218% for various conditions within the defined benchmarks. The interesting thing is that the improvements shown in Table 6.5 are constant for all the possible values of SNR in the network and therefore, do
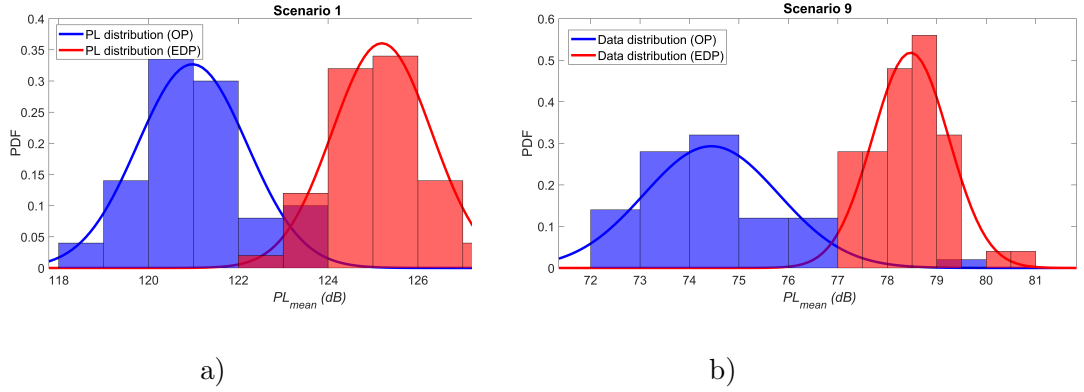
Figure 6.9: Statistics of mean PL for a) scenarios 1, and b) scenario 9.
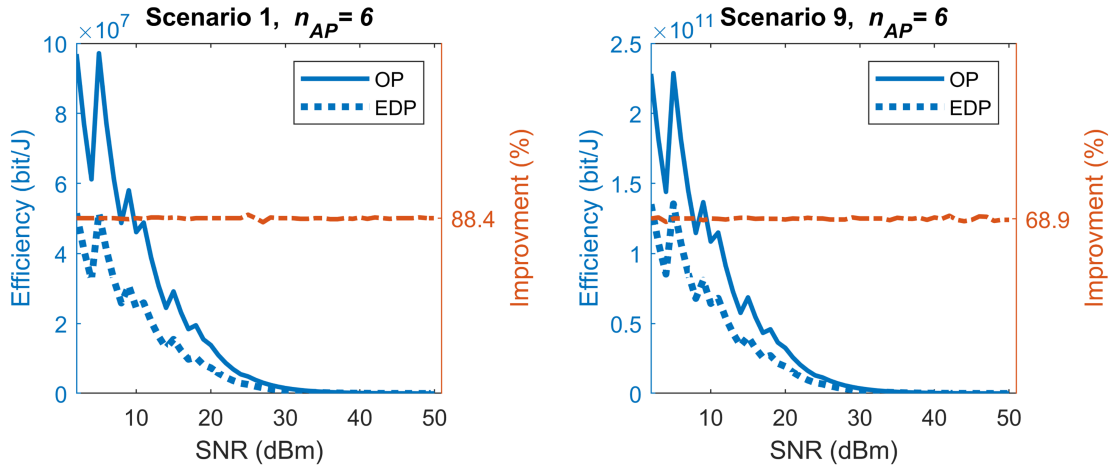
Figure 6.10: Efficiencies obtained by EDP and OP and the related improvement for $n_{AP} = 6$ and communication mode 1, for scenarios 1 and 9.

not change during train movement. This is because the rate of change in efficiency over SNR is the same for both EDP and OP algorithms at every scenario. For more clarification, we illustrate the efficiencies of both EDP and OP algorithms separately in Figure 6.10 for $n_{AP} = 6$ and communication mode 1 for scenarios 1 and 9 as examples. Furthermore, in Figure 6.10, we show the related improvements, which as can be seen, are constant over SNR. Due to similarity, we ignored such results for the other scenarios, other communication mode and the other number of APs. However, the numeric values of all the results are represented in Table 6.5.

Table 6.5: Improvement of efficiency obtained OP over EDP for $n_{AP} = 3 - 6$ and two different communication modes at every scenario

Scenario No.	Improvement [%]							
	Communication Mode 1				Communication Mode 2			
	$n_{AP} = 3$	$n_{AP} = 4$	$n_{AP} = 5$	$n_{AP} = 6$	$n_{AP} = 3$	$n_{AP} = 4$	$n_{AP} = 5$	$n_{AP} = 6$
1	144.5	88.6	186.7	88.4	145.2	121.6	155.4	80.5
2	104.9	217.9	105.6	58.4	104.9	61.2	118.4	60.4
3	115.9	71.4	97.7	51.3	107.2	46.2	96.9	51.4
4	96.4	55.6	72.2	42.4	95.3	55.4	70.4	41.8
5	107.6	153.5	83.2	45.8	108.6	61.2	92.4	45.5
6	70.9	97.4	57.0	31.7	69.2	30.7	57.8	33.2
7	96.4	55.6	72.2	42.4	95.3	55.4	70.4	41.8
8	104.9	217.9	105.7	58.4	104.9	61.2	118.4	60.4
9	104.7	207.3	102.1	68.9	103.9	207.1	105.1	76.1

6.3.5 HP Algorithm

As described, by using HP algorithm, we can determine both the required number and the optimal places of APs in a rail network. To evaluate the performance of HP algorithm, the two criteria that described in Section 6.2.5 must be met as follows. Firstly, the values of PL must not exceed the desired threshold at any point. Secondly, APs should be placed in a way that the average PL is minimum. To verify the second criteria, we show the amount of improvement in data transmission.

Through Table 6.4, we showed the ability of EDP algorithm to limit the PL to a desired threshold. HP algorithm uses the same method. Therefore, the first criteria has been already approved.

For the second criteria, we show the improvement in data transmission when using HP algorithm for AP placement compared to EDP algorithm in Table 6.6 through two

different communication modes for all the scenarios. As HP algorithm provides full solution for the AP placement, we add another scenario as scenario 10 to the previous test cases. Similar to scenario 9, scenario 10 is also a multi-environment scenario and has the same ECA. However, contrary to scenario 9 with uniform ECR, the weights of ECs in scenario 10 have a normal distribution as shown in (6.28). This ensures the correct performance of our proposed scheme for a multi-environment scenario with more complex ECR.

$$ECR_{Sc.10} =$$
$$[0.0001\ 0.0044\ 0.0540\ 0.2420\ 0.3989\ 0.2420\ 0.0540\ 0.0044\ 0.0001]$$

(6.28)

As shown in Table 6.6, by using HP algorithm, the efficiency of the network can be improved at least 21% in scenario 10 and up to 165% in scenario 5 compared to the baseline. Similar to the results of OP algorithm, the amounts of improvement for HP algorithm showed in Table 6.6, are also constant within all the possible values of SNR. To confirm this, we illustrate the efficiencies and the related improvements obtained by HP and EDP algorithms in a comparative manner in Figure 6.11 for scenarios 1 and 9 in communication mode 1. As seen, the efficiency of HP algorithm is higher than efficiency of EDP algorithm and the percentage of improvement is constant for all the values of SNR in the range. Due to similarity, we do not show the other scenarios and communication mode.

We have also used the model developed in Omnetpp version 5.4.1 in [9] to estimate the feasible capacity for data transmission through T2W communications via WiFi networks. The related results are shown in Figure 6.12 for scenario 9 for both communication modes and for a range of transmitter powers between $20\ mW - 50\ mW$. As shown, for a trip that takes less than five minutes in our case study, we can approximately transmit at least 6 gigabit and up to 254 gigabit data through T2W communications. Therefore, for a rail network including many trips that can take several hours, the amount of exchanged

Table 6.6: Results obtained by HP algorithm for every scenario

Scenario No.	n_{AP}	Improvement [%]	
		Communication Mode 1	Communication Mode 2
1	23	26.8	21.7
2	6	58.4	37.2
3	3	115.9	37.1
4	2	141.5	77.6
5	2	164.6	48.7
6	2	101.8	72.7
7	2	141.5	44.6
8	6	58.4	48.7
9	20	52.5	25.1
10	8	33	21.1

data through WiFi-based T2W communications will be over several hundreds of terabit.

6.3.6 The Effect of Different Scenarios on AP Placement

In this section, we show the AP placement for all the scenarios in two different cases: 1) based on a fixed desired PL threshold, and 2) based on a fixed AP number. For both the cases, we use EDP algorithm as well as OP and HP algorithms to see the differences between baseline and optimal placement. We especially focus on the differences between scenario 9, as a multi-environment scenario, and other single-environment scenarios, though, the differences among other scenarios can be seen as well.

For the first case, we perform AP placement by EDP and HP algorithms assuming 100 dB for PL threshold. Figure 6.13 shows the results for all the scenarios. The most important
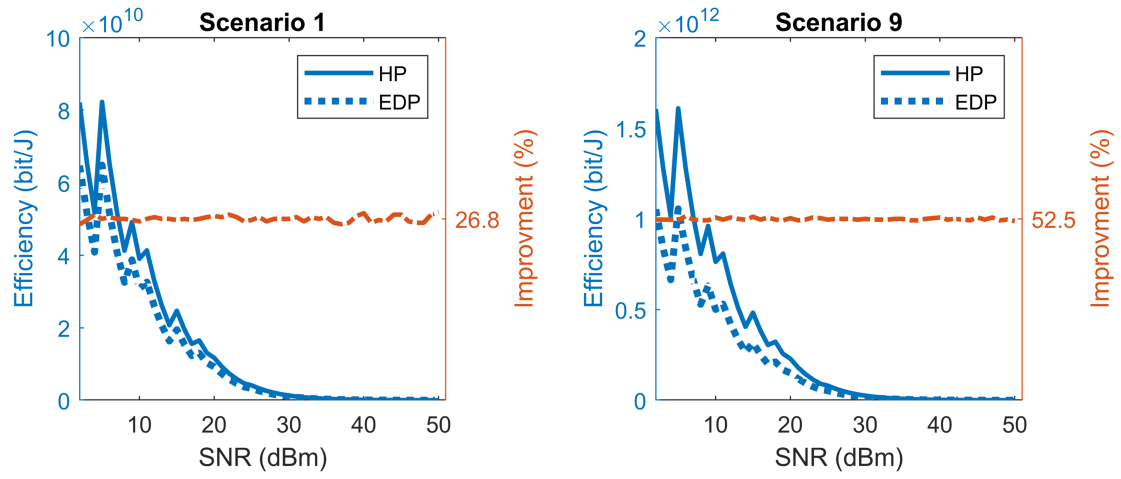
Figure 6.11: Efficiencies obtained by HP algorithm compared to EDP algorithm and the related improvements for scenarios 1 and 9
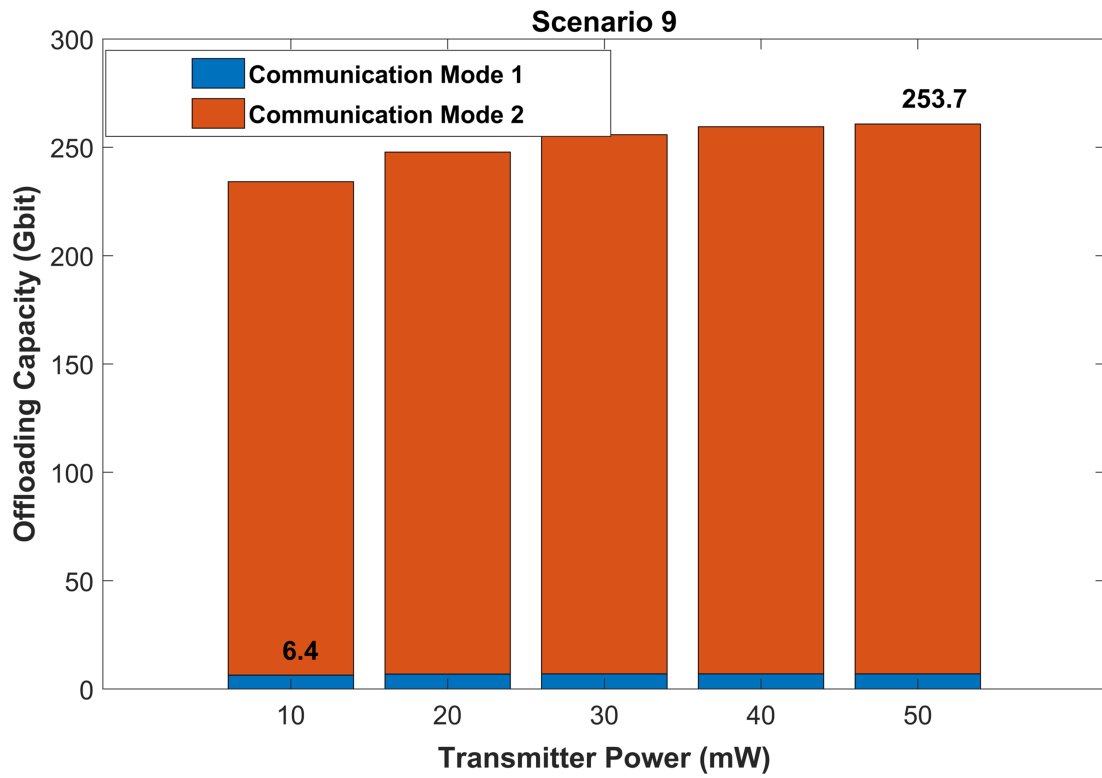
Figure 6.12: Low and high bounds of DEC for scenario 9
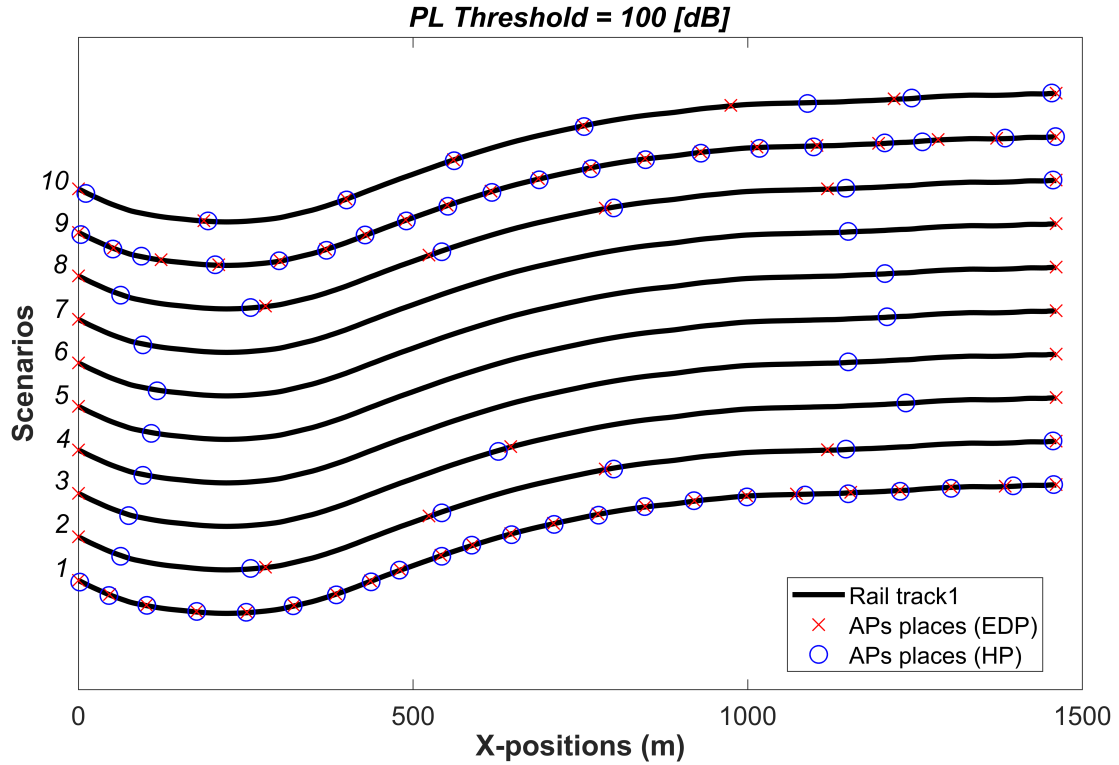
PL Threshold = 100 [dB]

Figure 6.13: AP Placement by EDP and HP algorithms for all the scenarios assuming a fixed PL threshold

observation we are seeking is the obvious differences between scenario 9, as a multi-environment scenario, and other single-environment scenarios. This clearly confirms that real rail lines, which pass through multiple environments, cannot be modeled by a single-environment scenario. Otherwise, the results will not be correct. Even for scenario 1 with the most similarity, the AP number and the efficiency are different, based on Table 6.6. Although, the difference between AP numbers of scenarios 1 and 9 is the lowest, i.e. 3, among others, the difference between the efficiencies is quite high around a hundred percent. Besides this, the differences between scenario 9 and other single-environment scenarios are significant.

For the second case, we run EDP and OP algorithms for $n_{AP} = 3$. The results are illustrated in Figure 6.14.a. Due to similarity, the results related to other number of

APs are not shown. As can be seen in Figure 6.14.a, while the results obtained by EDP algorithm are the same for all the scenarios, the results obtained by OP algorithm are different for every scenario despite the assumption of fixed AP numbers. Although, for the case of fixed AP number, the difference between some scenarios is not high, the comparison between Figure 6.14 and Table 6.5 for $n_{AP} = 3$ shows thateven slight changes in AP positions can have a significant impact on the efficiency of the network. On average, every 40-meter change (which is only 2% of the total trip length) in the position of every AP towards its optimal place approximately leads to a hundred percent increase in the network efficiency.

Additionally, the network average PL is illustrated in Figure 6.14.b to show that even for scenarios such as 1 and 9 with approximately similar placement, the difference between the average PL is quite high around 40 dB. This again confirms the difference between a multi-environment and a single-environment scenario.

Generally, both the above cases confirm the different results of AP placement for various scenarios, whether in terms of AP positions, average PL or network efficiency.

(a)

(b)
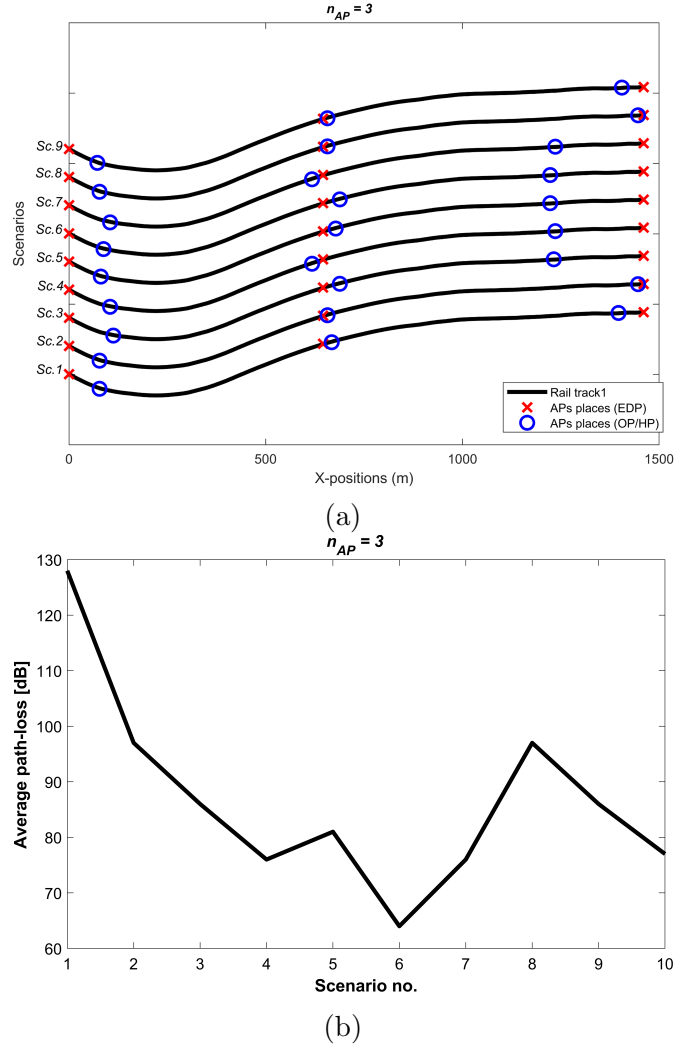
Figure 6.14: Results obtained by OP and EDP algorithms for a fixed number of APs:a) AP placement for every scenario, b) average PL at every scenario

6.3.7 Comparison with Measurement-Based Method

In the previous sections, we validated the performance of the proposed optimization algorithms by comparison with the results of EDP algorithm as the baseline. Although, EDP algorithm is an effective method for evaluation of OP and HP algorithms, we still need to compare our method with an existing method for AP placement to complete

the-state-of-the-art (as EDP algorithm itself has been developed internally through this chapter). For this purpose, we select *Measurement-Based Placement (MBP)* method used in [60]. In short, MBP method works as follows.

1. The first AP is placed in one end of the path.

2. Then, PL at all sample points of the path is calculated (measured) and the position with maximum PL is determined.

3. The next AP is placed in the middle of the previous AP position and the position with maximum PL.

4. The steps 1 and 2 are continued until all available APs are placed along the line.

For the case that a PL threshold is desired, the addition of APs is continued until PL falls under the given threshold. Table 6.7 illustrates the results obtained by every AP placement method for 4 APs. Scenarios 1 and 9 are selected for the simulation as examples of single and multiple environment scenarios, respectively. Due to similarity, the results for other scenarios and different AP numbers are not shown. As seen, the MBP method has the worst average PL and the lowest efficiency compared to our proposed algorithms. On average, by using the proposed HP algorithm, we achieve over 10 dB decrease in the average PL and more than 10-times increase in the efficiency of the network, compared with the MBP method.

6.3.8 Discussion about Doppler Effect

Due to the train movement, Doppler effect should be also considered in the placement of APs. Therefore, we estimate the average amount of reference PL, i.e. PL_0, as part of PL model that can be affected by carrier frequency. As well as the amount of reference PL in the static mode, we also estimate it in other two situations related the mobility

Table 6.7: Comparison between our proposed algorithms with MBP for scenarios 1 and 9 and $n_{AP} = 4$

AP Placement Method	Average PL [dB]		Efficiency [bit/J]	
	Sc. 1	Sc. 9	Sc. 1 $(\times 10^5)$	Sc. 9 $(\times 10^9)$
MBP	131	91	0.2	0.3
EDP	124	85	1.3	1.1
HP	120	80	2.9	3.3

modes. One is relating to the time that a train is approaching (PL_0^{appr}) an AP. Another is pertinent to the moment that the train is leaving (PL_0^{leav}) an AP. Then, we calculate the differences between the values of PL_0 during standing and moving. The related results are shown in Table 6.8 (in last page) for three different train speeds assuming 2.4 GHz and 5 GHz as carrier frequencies. As illustrated, the change of PL due to Doppler effect is around a few micro dB between ± 0.8 and ± 2.4, which is really low and can be neglected.

Table 6.8: Impact of Doppler Effect on PL, for different frequencies and various speeds

v	PL_0 (dB)		PL_0^{appr} (dB)		PL_0^{leav} (dB)		$diff_{appr}$ (μdB)		$diff_{leav}$ (μdB)	
	$f_c = 2.4$ GHz	$f_c = 5$ GHz	$f_c = 2.4$ GHz	$f_c = 5$ GHz	$f_c = 2.4$ GHz	$f_c = 5$ GHz	$f_c = 2.4$ GHz	$f_c = 5$ GHz	$f_c = 2.4$ GHz	$f_c = 5$ GHz
100 Km/h	40.0460	46.4212	40.0460	46.4212	40.0460	46.4212	- 0.8043	- 0.8043	0.8043	0.8043
200 Km/h	40.0460	46.4212	40.0460	46.4212	40.0460	46.4212	- 1.6045	- 1.6045	1.6045	1.6045
300 Km/h	40.0460	46.4212	40.0460	46.4212	40.0460	46.4212	- 2.4127	- 2.4127	2.4127	2.4127

6.4 Conclusion and Future Works

We proposed three algorithms included EDP, OP and HP algorithms for placement of APs in a rail network. The least number of APs required for a network can be found by EDP algorithm so that the maximum PL does not exceed a desired threshold. We can also determine the initial places of APs through EDP algorithm that is required at the running stage of OP and HP algorithms. Additionally, EDP algorithm can be used for comparison purposes as the baseline algorithm to show the efficiency of OP and HP algorithms.

Through OP algorithm, the optimal places for a given number of APs can be found so that the average PL is minimum. OP algorithm is proper when we have a limited number of APs and seeks the best places for those APs. We showed that the efficiency of OP algorithm is at least 31% and up to 218% for different number of APs within all the defined scenarios.

By using HP algorithm, we can determine both required number and optimal places of APs in a rail network. We showed that the efficiency of the system can be improved at least 21% and up to 165% compared to the baseline by HP algorithm. In addition to use the EDP algorithm as the baseline, we also compared HP algorithm with MBP as an existing AP placement method. The results showed that HP algorithm is over 10 times more efficient than MBP method.

We also proposed REM as a method to model the changes of communication characteristics through different rail scenarios including urban, suburban, rural, viaduct, cutting, tunnel, river and station. We showed that the placement of APs is highly dependent on the REM and could have a massive impact on the efficiency of the system.

Additionally, we showed that we can theoretically transmit over 250 gigabit data through T2W communications when using the common WiFi technologies.

Finally, through extensive simulations over various speeds and different carrier frequencies, we proved that the Doppler effect does not affect the proposed placement algorithms.

In this chapter, we decreased the transmission power in T2W communications by reducing the average path-loss of the network. In the future work, we will go a step further by optimizing the power consumption of APs through limiting the time that every AP should stay on. This will provide higher efficiency for power-limited APs such as those that use combination of renewable and battery as power supply.

7 Conclusion and Future Works

In this thesis, we firstly developed a classifier that could realize critical data from delay-tolerant data in an online manner. Then, for transmission of delay-tolerant data, we developed three different models include T2S, T2T and T2W communication methods.

For T2S communication method, we developed an analytical model that could estimate the amount of offloaded data between trains and stations. We validated the performance of our model through simulation in various scenarios in Omnet network simulator. The simulation results showed over 98% accuracy. Additionally, by using our proposed scheme, we could theoretically offload up to 5.43 GB at every station.

For T2T communication method, we proposed a novel mobility model that could provide train traffic traces in real-time. As the proposed mobility model needed no GPS module, it could provide a practical solution when signal from GPS or Assisted-GPS is poor or unavailable such as in urban area or inside tunnels. Furthermore, as we used an energy optimization function, the proposed mobility model could provide a guidance trajectory for trains to have an energy-optimized operation. We also developed an algorithm that could determine the specifications of contacts between trains based on the traffic traces obtained from the mobility model. We validated our proposed model using data collected from Sydney Trains of Australia. The results obtained from our proposed model showed over 98 percent accuracy in comparison with the real data collected via a GPS module in Sydney Trains of Australia.

Finally, we proposed three algorithms for placement of access points (APs) along a rail line needed for the proposed T2W communication method. The first algorithm was proposed to find the minimum number of APs so that the path-loss (PL) did not exceed a desired threshold. Through the second algorithm, the most optimal places for a desired number of APs were determined so that the average PL was minimum. The goal of the third algorithm was to determine the required number and optimal places of APs in a rail network. Furthermore, we proposed a model to consider the effects of changes of communication characteristics on the placement of APs for rail lines in different scenarios. Through such model, the algorithms proposed for placement of APs could be used in different railway scenarios. The proposed algorithms were validated through extensive simulations in Sydney Trains of Australia. The simulation results showed that the proposed approach could improve the efficiency of the system at least 22% and up to 165% within the different defined scenarios. We also showed that we could approximately transmit over 250 Gigabit data through T2W communications over common WiFi networks.

Additionally, we highlight the key features of every section as follows.

7.1 Data Classification

- The classifier was successfully tested for three different data sets with over 98% accuracy.

- With the awareness of class of data, the classifier provides the possibility of using multi-modal communication methods. This enables us to significantly decrease the traffic from on-demand cellular networks and instead, by using cheap channels through existing WLANs, we can have a massive cost reduction (99%).

- Contrary to the regular ML-based model that should be trained in offline mode, our developed classifier can be trained online. This enables the model to be learned during operation and therefore it can be adapted to any rolling stock during running. It means we does not need a large amount of batched data from the history of the train to learn the classifier.

- By realizing critical data, the proposed classifier can be used as an on-board monitoring system. Additionally, it can help the train driver in early diagnosis of failures as a decision-support system.

- The proposed classification model and all the corresponding results were published in the high quality journal of IEEE Transaction on Intelligent Transportation Systems in 2020 with 5.7 impact factor. Please see the reference [8] for more detail.

7.2 T2S Communication

- The proposed T2S model was successfully tested for both stopping and passing stations with over 98% accuracy.

- We can offload over 5 GB data per station through the proposed approach. Additionally, as the model can be used with other communication methods, by using new emerging high capacity radios such as IEEE 802.11ay, the amount of exchange data will be massive.

- As the model uses the existing WLANs for data transmission, no upfront cost is required for infrastructure, which results in a significant cost reduction.

- By using a rate control algorithm, the proposed model can work even with very poor wireless signal.

- By proposing stations as offloading spots, trains can exchange IoT data as well as passengers during dwelling times.

- The proposed T2S model and the corresponding results were published in 2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring). Please refer to [9].

7.3 T2T Communication

- The proposed T2T communication method provides significant data offloading capacity in parallel with the T2S method. The simulation results in Line T7 of Sydney Trains of Australia showed that we can potentially offload over 20 GB data during the contact periods among trains in one operating day by using IEEE 802.11p-based DSRC method.

- By developing an internal mobility model, the T2T approach does not need GPS signal to determine the contact specifications. This can be highly beneficial when the GPS signal is poor or unavailable, e.g. in congested urban districts or remote regional areas.

- As we showed in Chapter 5, Doppler Effect has no impact on the proposed T2T communication channel. This makes the model robust in highly mobile scenarios such as vehicular networks.

- The proposed T2T approach can accurately provide train mobility traces with less than 2% error (in terms of position) compared to GPS data. Additionally, as the internal mobility model works based on train timetables, we showed that for up to 20% change in a given timetable, the corresponding error will be less than 10%.

- The proposed T2T approach with the related results were submitted as a journal paper in IEEE Transaction on Intelligent Transportation Systems, which is currently under review.

7.4 T2W Communication

- We developed three novel AP placement algorithms and one environment model in rail networks that can comprehensively provide solutions for various design situations, different network conditions and 10 different railway scenarios.

- By using the proposed T2W communication approach, we can exchange data between running trains and wayside units in a continuous manner. In a case study in Sydney Trains of Australia, we showed that we can potentially exchange over 30 GB data between a train and wayside units through the existing WiFi networks.

- Through extensive simulations over various speeds and different carrier frequencies, we proved that the Doppler Effect has no impact on the proposed placement algorithms. This make our approach robust and highly appropriate for vehicular networks.

- The proposed T2W approach with the related results were submitted as a journal paper in IEEE Transactions on Vehicular Technology (TVT), which is currently under review.

7.5 Future Work

Other ideas that can be further discussed as the future works of this thesis are as follows.

- Antenna design for T2X communications: finding number of required antennas, optimal types of omni and directional antennas or switching between them, etc. to maximize the communication zone.

- A transmission energy optimization through finding the optimal switching time of communication devices. Such algorithm can significantly decrease the transmitter energy consumption by reducing the total active time of communication devices.

- Developing a multi-modal train-based data-offloading algorithm that can handle request-based data offloading tasks with restricted delay. For this purpose, we need to seek the most optimum offloading strategy among three different communication methods include T2S, T2T and T2W based on the situation.

- Prototyping an intelligent platform for rail networks as "Internet of Trains". Some features of that platform can be IoT data collection, online data classification, On-board decision making support to assist train drivers, an intelligent multi-modal data transmission support using different kinds of T2X communication methods.

- Optimizing the power consumption of APs through limiting the time that every AP should stay on while attempting to maximize the offloading capacity of the network is another research problem study in the future.

- Expanding the applicability of T2X communications models to broader areas such as V2X communications or more generally X2X communications.

Bibliography

[1] L. Zhu, F. R. Yu, B. Ning, and T. Tang, "Communication-based train control (cbtc) systems with cooperative relaying: Design and performance analysis," *IEEE Transactions on Vehicular Technology*, vol. 63, no. 5, pp. 2162–2172, 2014.

[2] N. Transport, "Routes and timetables," pp. https://transportnsw.info/routes/train (accessed Mar. 27, 2019). [Online]. Available: https://transportnsw.info/routes/train

[3] V. J. Hodge, S. O'Keefe, M. Weeks, and A. Moulds, "Wireless sensor networks for condition monitoring in the railway industry: A survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 3, pp. 1088–1106, 2015.

[4] M. T. S. Jovanovic, D. Bozovic, "Railway infrastructure condition-monitoring and analysis as a basis for maintenance management," *GRADEVINAR*, vol. 66, no. 4, pp. 347–358, 2014.

[5] E. Fumeo, L. Oneto, and D. Anguita, "Condition based maintenance in railway transportation systems based on big data streaming analysis," *Procedia Computer Science*, vol. 53, pp. 437–446, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877050915018244

[6] R. He, B. Ai, G. Wang, K. Guan, Z. Zhong, A. F. Molisch, C. Briso-Rodriguez, and C. P. Oestges, "High-speed railway communications: From gsm-r to lte-r," *IEEE Vehicular Technology Magazine*, vol. 11, no. 3, pp. 49–58, 2016.

[7] T. Wang, P. Li, X. Wang, Y. Wang, T. Guo, and Y. Cao, "A comprehensive survey on mobile data offloading in heterogeneous network," *Wireless Networks*, vol. 25, no. 2, pp. 573–584, 2019.

[8] M. Saki, M. Abolhasan, and J. Lipman, "A novel approach for big data classification and transportation in rail networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 3, pp. 1239–1249, March 2020.

[9] M. Saki, M. Abolhasan, and J. Lipman, "A big sensor data offloading scheme in rail networks," in *2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring)*. IEEE, 2019, pp. 1–6.

[10] M. Saki, M. Abolhasan, J. Lipman, and A. Jamalipour, "Mobility model for contact-aware data offloading through train-to-train communications in rail networks," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–13, 2020.

[11] ——, "A comprehensive access point placement for iot data transmission through train-wayside communications in multi-environment based rail networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 10, pp. 11 937–11 949, 2020.

[12] L. Zhu, F. R. Yu, Y. Wang, B. Ning, and T. Tang, "Big data analytics in intelligent transportation systems: a survey," *IEEE Transactions on Intelligent Transportation Systems*, no. 99, pp. 1–16, 2018.

[13] H. Li, D. Parikh, Q. He, B. Qian, Z. Li, D. Fang, and A. Hampapur, "Improving rail network velocity: A machine learning approach to predictive maintenance," *Transportation Research Part C: Emerging Technologies*, vol. 45,

pp. 17–26, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0968090X14001107

[14] L. Oneto, I. Buselli, A. Lulli, R. Canepa, S. Petralli, and D. Anguita, "A dynamic, interpretable, and robust hybrid data analytics system for train movements in large-scale railway networks," *International Journal of Data Science and Analytics*, 2019. [Online]. Available: https://doi.org/10.1007/s41060-018-00171-z

[15] A. Lulli, L. Oneto, R. Canepa, S. Petralli, and D. Anguita, "Large-scale railway networks train movements: A dynamic, interpretable, and robust hybrid data analytics system," in *2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA)*, Conference Proceedings, pp. 371–380.

[16] H. Parkinson and G. Bamford, "The potential for using big data analytics to predict safety risks by analyzing rail accidents," in *3rd International Conference on Railway Technology: Research, Development and Maintenance, Cagliari, Sardinia, Italy*, 2016, pp. 5–8.

[17] J. Liu, X. Wang, A. J. Khattak, J. Hu, J. Cui, and J. Ma, "How big data serves for freight safety management at highway-rail grade crossings? a spatial approach fused with path analysis," *Neurocomputing*, vol. 181, pp. 38–52, 2016.

[18] F. Ghofrani, Q. He, R. M. P. Goverde, and X. Liu, "Recent applications of big data analytics in railway transportation systems: A survey," *Transportation Research Part C: Emerging Technologies*, vol. 90, pp. 226–246, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0968090X18303395

[19] A. De Mauro, M. Greco, and M. Grimaldi, "A formal definition of big data based on its essential features," *Library Review*, 2016.

[20] O. Eker, F. Camci, and U. Kumar, "Svm based diagnostics on railway turnouts," *International Journal of Performability Engineering*, vol. 8, no. 3, pp. 289–298, 2012.

[21] M. Macucci, S. D. Pascoli, P. Marconcini, and B. Tellini, "Derailment detection and data collection in freight trains, based on a wireless sensor network," *IEEE Transactions on Instrumentation and Measurement*, vol. 65, no. 9, pp. 1977–1987, 2016.

[22] H. C. Lee, Y. C. Chang, and Y. S. Huang, "A reliable wireless sensor system for monitoring mechanical wear-out of parts," *IEEE Transactions on Instrumentation and Measurement*, vol. 63, no. 10, pp. 2488–2497, 2014.

[23] A. L. Schiavo, "Fully autonomous wireless sensor network for freight wagon monitoring," *IEEE Sensors Journal*, vol. PP, no. 99, pp. 1–1, 2016.

[24] G. M. Shafiullah, S. A. Azad, and A. B. M. S. Ali, "Energy-efficient wireless mac protocols for railway monitoring applications," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 2, pp. 649–659, 2013.

[25] T. M, S. Sharma, R. Singh, K. Shubham, and R. Kumar, "Two-layer optimized railway monitoring system using wi-fi and zigbee interfaced wireless sensor network," *IEEE Sensors Journal*, vol. PP, no. 99, pp. 1–1, 2017.

[26] S. M. Rakshit, M. Hempel, and H. Sharif, "Study of a dual radio sensor platform for effective on-board real-time monitoring of freight trains," in *2016 International Wireless Communications and Mobile Computing Conference (IWCMC)*, Conference Proceedings, pp. 812–817.

[27] I. Val, A. Arriola, C. Cruces, R. Torrego, E. Gomez, and X. Arizkorreta, "Time-synchronized wireless sensor network for structural health monitoring applications

in railway environments," in *2015 IEEE World Conference on Factory Communication Systems (WFCS)*, Conference Proceedings, pp. 1–9.

[28] G. A. Barone, J. O. Breitfeller, D. A. Jacobs, C. W. Nyquist, and R. E. Rose, "Self-assembling wireless network, vehicle communications system, railroad wheel and bearing monitoring system and methods therefor," *US Patent*, 2010. [Online]. Available: https://www.google.com/patents/US7705743

[29] S. Chiocchio, A. Persia, F. Santucci, V. D. Claudio, D. D. Grande, P. Giugliano, and G. Guidotti, "A cloud-based heterogeneous wireless platform for monitoring and management of freight trains," in *2016 8th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, Conference Proceedings, pp. 263–268.

[30] M. Gruden, M. Hinnemo, D. Dancila, F. Zherdev, N. Edvinsson, K. Brunberg, L. Andersson, R. Bystrom, and A. Rydberg, "Field operational testing for safety improvement of freight trains using wireless monitoring by sensor network," *IET Wireless Sensor Systems*, vol. 4, no. 2, pp. 54–60, 2014.

[31] M. Zarafshan-Araki and K.-W. Chin, "Trainnet: A transport system for delivering non real-time data," *Computer Communications*, vol. 33, no. 15, pp. 1850–1863, 2010.

[32] B. Baron, P. Spathis, H. Rivano, and M. D. de Amorim, "Offloading massive data onto passenger vehicles: Topology simplification and traffic assignment," *IEEE/ACM Transactions on Networking*, vol. 24, no. 6, pp. 3248–3261, 2016.

[33] E. M. Husni and A. R. Sumarmo, "Delay tolerant network utilizing train for news portal and email services," in *Proceeding of the 3rd International Conference on Information and Communication Technology for the Moslem World (ICT4M) 2010*. IEEE, 2010, pp. G6–G10.

[34] S. Paul and S. Kumar, "Railnet infrastructure generation using sumo," 2016.

[35] D. Suh, H. Ko, and S. Pack, "Efficiency analysis of wifi offloading techniques," *IEEE Trans. Vehicular Technology*, vol. 65, no. 5, pp. 3813–3817, 2016.

[36] H. Deng and I. Hou, "On the capacity-performance trade-off of online policy in delayed mobile offloading," *IEEE Transactions on Wireless Communications*, vol. 16, no. 1, pp. 526–537, Jan 2017.

[37] S. Kashihara, "Transmission scheduling method for delivery of large-sized data object based on delay constrained request over scheduled transportation vehicles," *EPI International Journal of Engineering (EPIIJE)*, vol. 1, no. 1, pp. 60–68, 2018.

[38] C.-M. Huang, M.-S. Chiang, D.-T. Dao, H.-M. Pai, S. Xu, and H. Zhou, "Vehicle-to-infrastructure (v2i) offloading from cellular network to 802.11p wi-fi network based on the software-defined network (sdn) architecture," *Vehicular Communications*, vol. 9, pp. 288 – 300, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2214209616301942

[39] M. Saki, M. Abolhasan, and J. Lipman, "A big sensor data offloading scheme in rail networks," in *2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring)*, April 2019, pp. 1–6.

[40] B. Baron, P. Spathis, H. Rivano, M. D. de Amorim, Y. Viniotis, and M. Ammar, "Centrally-controlled mass data offloading using vehicular traffic," *IEEE Transactions on Network and Service Management*, vol. 14, no. 2, pp. 401–415, 2017.

[41] X. Zhu, Y. Li, D. Jin, and J. Lu, "Contact-aware optimal resource allocation for mobile data offloading in opportunistic vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 8, pp. 7384–7399, 2017.

[42] A. Bazzi, B. M. Masini, A. Zanella, and G. Pasolini, "Ieee 802.11p for cellular offloading in vehicular sensor networks," *Computer Communications*, vol. 60, pp. 97–108, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0140366415000274https://ac.els-cdn.com/S0140366415000274/1-s2.0-S0140366415000274-main.pdf?_tid=c53149f9-20ee-4ae9-a5ee-4b152730bee2&acdnat=1550094176_a3e665ab3adede9bcd26027b937fb785

[43] M. Taheri, N. Ansari, J. Feng, R. Rojas-Cessa, and M. Zhou, "Provisioning internet access using fso in high-speed rail networks," *IEEE Network*, vol. 31, no. 4, pp. 96–101, 2017.

[44] Y. Kaymak, R. Rojas-Cessa, J. Feng, N. Ansari, M. Zhou, and T. Zhang, "A survey on acquisition, tracking, and pointing mechanisms for mobile free-space optical communications," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 2, pp. 1104–1123, 2018.

[45] P. Unterhuber, I. Rashdan, M. Walter, and T. Kürner, "Path loss models and large scale fading statistics for c-band train-to-train communication," in *2020 14th European Conference on Antennas and Propagation (EuCAP)*. IEEE, 2020, pp. 1–5.

[46] C. Briso-Rodríguez, P. Fratilescu, and Y. Xu, "Path loss modeling for train-to-train communications in subway tunnels at 900/2400 mhz," *IEEE Antennas and Wireless Propagation Letters*, vol. 18, no. 6, pp. 1164–1168, 2019.

[47] J. Zhao, Y. Zhang, Y. Nie, and J. Liu, "Intelligent resource allocation for train-to-train communication: A multi-agent deep reinforcement learning approach," *IEEE Access*, vol. 8, pp. 8032–8040, 2020.

[48] X. Wang, L. Liu, L. Zhu, and T. Tang, "Train-centric cbtc meets age of information

in train-to-train communications," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–14, 2019.

[49] P. Fraga-Lamas, T. Fernandez-Caram, and L. Castedo, "Towards the internet of smart trains: A review on industrial iot-connected railways," *Sensors*, vol. 17, no. 6, p. 1457, 2017.

[50] A. Festag, "Cooperative intelligent transport systems standards in europe," *IEEE Communications Magazine*, vol. 52, no. 12, pp. 166–172, December 2014.

[51] A. Bazzi, B. M. Masini, A. Zanella, and I. Thibault, "On the performance of ieee 802.11p and lte-v2v for the cooperative awareness of connected vehicles," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 11, pp. 10 419–10 432, 2017.

[52] Jian He, A. A. Verstak, L. T. Watson, C. A. Stinson, N. Ramakrishnan, C. A. Shaffer, T. S. Rappaport, C. R. Anderson, K. K. Bae, Jing Jiang, and W. H. Tranter, "Globally optimal transmitter placement for indoor wireless communication systems," *IEEE Transactions on Wireless Communications*, vol. 3, no. 6, pp. 1906–1911, Nov 2004.

[53] Xiang Ling and Kwan Lawrence Yeung, "Joint access point placement and channel assignment for 802.11 wireless lans," *IEEE Transactions on Wireless Communications*, vol. 5, no. 10, pp. 2705–2711, Oct 2006.

[54] C. Ting, C. Lee, H. Chang, and J. Wu, "Wireless heterogeneous transmitter placement using multiobjective variable-length genetic algorithm," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 39, no. 4, pp. 945–958, Aug 2009.

[55] H. Liang, B. Wang, W. Liu, and H. Xu, "A novel transmitter placement scheme based on hierarchical simplex search for indoor wireless coverage optimization,"

IEEE Transactions on Antennas and Propagation, vol. 60, no. 8, pp. 3921–3932, Aug 2012.

[56] A. So and B. Liang, "Efficient wireless extension point placement algorithm in urban rectilineal wlans," *IEEE Transactions on Vehicular Technology*, vol. 57, no. 1, pp. 532–547, Jan 2008.

[57] P. Li, X. Huang, Y. Fang, and P. Lin, "Optimal placement of gateways in vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. 56, no. 6, pp. 3421–3430, Nov 2007.

[58] B. Zhang, X. Jia, K. Yang, and R. Xie, "Design of analytical model and algorithm for optimal roadside ap placement in vanets," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 9, pp. 7708–7718, Sep. 2016.

[59] T. Wen, C. Constantinou, L. Chen, Z. Tian, and C. Roberts, "Access point deployment optimization in cbtc data communication system," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 6, pp. 1985–1995, 2018.

[60] X. Zhang, A. Ludwig, N. Sood, and C. D. Sarris, "Physics-based optimization of access point placement for train communication systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 9, pp. 3028–3038, 2018.

[61] A. R. S. Industry, "On track to 2040 roadmap: Preparing the australian rail supply industry for challenges and growth. 2012 p. 88," *Report No.: 0.*

[62] T. Bureau of Infrastructure and R. E. (BITRE), "Yearbook 2013: Australian infrastructure statistics, statistical report," Report, 2013.

[63] A. Gandomi and M. Haider, "Beyond the hype: Big data concepts, methods, and analytics," *International Journal of Information Management*, vol. 35, no. 2, pp. 137–144, 2015.

[64] P. Weston, C. Roberts, G. Yeo, and E. Stewart, "Perspectives on railway track geometry condition monitoring from in-service railway vehicles," *Vehicle System Dynamics*, vol. 53, no. 7, pp. 1063–1091, 2015. [Online]. Available: http://dx.doi.org/10.1080/00423114.2015.1034730

[65] Q. Li, Z. Zhong, Z. Liang, and Y. Liang, "Rail inspection meets big data: Methods and trends," in *2015 18th International Conference on Network-Based Information Systems*, Conference Proceedings, pp. 302–308.

[66] S. Bi, R. Zhang, Z. Ding, and S. Cui, "Wireless communications in the era of big data," *IEEE communications magazine*, vol. 53, no. 10, pp. 190–199, 2015.

[67] M. Khalil, J. Qadir, O. Onireti, M. A. Imran, and S. Younis, "Feasibility, architecture and cost considerations of using tvws for rural internet access in 5g," in *2017 20th Conference on Innovations in Clouds, Internet and Networks (ICIN)*. IEEE, 2017, pp. 23–30.

[68] E. Baccarelli, N. Cordeschi, A. Mei, M. Panella, M. Shojafar, and J. Stefa, "Energy-efficient dynamic traffic offloading and reconfiguration of networked data centers for big data stream mobile computing: review, challenges, and a case study," *IEEE Network*, vol. 30, no. 2, pp. 54–61, 2016.

[69] M. Gu, X. Li, and Y. Cao, "Optical storage arrays: a perspective for future big data storage," *Light: Science Applications*, vol. 3, no. 5, p. e177, 2014.

[70] W. Dargie, "Analysis of time and frequency domain features of accelerometer measurements," in *2009 Proceedings of 18th International Conference on Computer Communications and Networks*, Conference Proceedings, pp. 1–6.

[71] H. Saruhan, S. Sandemir, A. Cicek, and I. Uygur, "Vibration analysis of rolling element bearings defects," *Journal of applied research and technology*, vol. 12, no. 3, pp. 384–395, 2014.

[72] J. P. Amezquita-Sanchez and H. Adeli, "Signal processing techniques for vibration-based health monitoring of smart structures," *Archives of Computational Methods in Engineering*, vol. 23, no. 1, pp. 1–15, 2016.

[73] C. Sobie, C. Freitas, and M. Nicolai, "Simulation-driven machine learning: Bearing fault classification," *Mechanical Systems and Signal Processing*, vol. 99, pp. 403–419, 2018.

[74] A. Rai and S. Upadhyay, "A review on signal processing techniques utilized in the fault diagnosis of rolling element bearings," *Tribology International*, vol. 96, pp. 289–306, 2016.

[75] W. Qiao and D. Lu, "A survey on wind turbine condition monitoring and fault diagnosisï¿œpart i: Components and subsystems," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 10, pp. 6536–6545, 2015.

[76] B. Dolenc, P. Bovskoski, and D. Jurivic, "Distributed bearing fault diagnosis based on vibration analysis," *Mechanical Systems and Signal Processing*, vol. 66, pp. 521–532, 2016.

[77] J. Zarei, M. A. Tajeddini, and H. R. Karimi, "Vibration analysis for bearing fault detection and classification using an intelligent filter," *Mechatronics*, vol. 24, no. 2, pp. 151–157, 2014.

[78] P. Nectoux, R. Gouriveau, K. Medjaher, E. Ramasso, B. Morello, N. Zerhouni, and C. Varnier, "An experimental platform for bearings accelerated life test," in *Proceedings of the IEEE International Conference on Prognostics and Health Management*, 2012.

[79] T. Liu, M. Lyu, Z. Wang, and S. Yan, "An identification method of orbit responses rooting in vibration analysis of rotor during touchdowns of active magnetic bearings," *Journal of Sound and Vibration*, vol. 414, pp. 174–191, 2018.

[80] E. T. Esfahani, S. Wang, and V. Sundararajan, "Multisensor wireless system for eccentricity and bearing fault detection in induction motors," *IEEE/ASME Transactions on Mechatronics*, vol. 19, no. 3, pp. 818–826, June 2014.

[81] R. N. Khushaba, A. Al-Timemy, and S. Kodagoda, "Influence of multiple dynamic factors on the performance of myoelectric pattern recognition," in *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 2015, pp. 1679–1682.

[82] H. Wimmer and L. Powell, "Principle component analysis for feature reduction and data preprocessing in data science," in *Proceedings of the Conference on Information Systems Applied Research ISSN*, vol. 2167, 2016, p. 1508.

[83] S. M. Erfani, S. Rajasegarar, S. Karunasekera, and C. Leckie, "High-dimensional and large-scale anomaly detection using a linear one-class svm with deep learning," *Pattern Recognition*, vol. 58, pp. 121–134, 2016.

[84] A. Zivkovic, M. Zeljkovic, and S. Tabakovic, "Nonlinear dynamic behaviours do ball bearing due to raceways waviness," *International Conference on Manufacturing Science and Education-MSE*, 2013.

[85] A. Widodo and B.-S. Yang, "Support vector machine in machine condition monitoring and fault diagnosis," *Mechanical systems and signal processing*, vol. 21, no. 6, pp. 2560–2574, 2007.

[86] S. M. Ashley and A., "Official ieee 802.11 working group project timelines," *Online*, 2016. [Online]. Available: http://www.ieee802.org/11/Reports/802.11_Timelines.htm

[87] T. E. Bogale and L. B. Le, "Massive mimo and mmwave for 5g wireless hetnet: potential benefits and challenges," *IEEE Vehicular Technology Magazine*, vol. 11, no. 1, pp. 64–75, 2016.

[88] J. L. Hai Qiu, Jay Lee, "Wavelet filter-based weak signature detection method and its application on roller bearing prognostics," *Journal of Sound and Vibration*, vol. 289, pp. 1066–1090, 2006.

[89] R. V. Daniel, S. A. Siddhappa, S. B. Gajanan, S. V. Philip, and P. S. Paul, "Effect of bearings on vibration in rotating machinery," in *IOP Conference Series: Materials Science and Engineering*, vol. 225, no. 1. IOP Publishing, 2017, p. 012264.

[90] Telstra, "Telstra data packs," Report, 2017. [Online]. Available: https://www.telstra.com.au/mobile-phones/plans-and-rates/data-packs

[91] H. Zhou, H. Wang, X. Li, and V. C. M. Leung, "A survey on mobile data offloading technologies," *IEEE Access*, vol. 6, pp. 5101–5111, 2018.

[92] D. Li, W. Daamen, and R. M. Goverde, "Estimation of train dwell time at short stops based on track occupation event data: A study at a dutch railway station," *Journal of Advanced Transportation*, vol. 50, no. 5, pp. 877–896, 2016.

[93] M. Lacage, M. H. Manshaei, and T. Turletti, "Ieee 802.11 rate adaptation: a practical approach," in *Proceedings of the 7th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems*, 2004, pp. 126–134.

[94] "Ieee standard for information technology–telecommunications and information exchange between systems local and metropolitan area networks–specific requirements - part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications," *IEEE Std 802.11-2016 (Revision of IEEE Std 802.11-2012)*, pp. 1–3534, 2016.

[95] S. Fu, Y. Zhang, M. Ceriotti, Y. Jiang, M. Packeiser, and P. J. Marrï¿œn, "Modeling packet loss rate of ieee 802.15.4 links in diverse environmental conditions," in

2018 IEEE Wireless Communications and Networking Conference (WCNC), April 2018, pp. 1–6.

[96] J. Luomala and I. Hakala, "Effects of temperature and humidity on radio signal strength in outdoor wireless sensor networks," in *2015 Federated Conference on Computer Science and Information Systems (FedCSIS)*, Conference Proceedings, pp. 1247–1255.

[97] T. Vanhatupa, "Wi-fi capacity analysis for 802.11 ac and 802.11 n: Theory practice," *Ekahau Inc*, 2013.

[98] S. Wang, T. Lei, L. Zhang, C.-H. Hsu, and F. Yang, "Offloading mobile data traffic for qos-aware service provision in vehicular cyber-physical systems," *Future Generation Computer Systems*, vol. 61, pp. 118–127, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167739X15003179

[99] L. Zhu, F. R. Yu, Y. Wang, B. Ning, and T. Tang, "Big data analytics in intelligent transportation systems: A survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 1, pp. 383–398, Jan 2019.

[100] F. Ghofrani, Q. He, R. M. Goverde, and X. Liu, "Recent applications of big data analytics in railway transportation systems: A survey," *Transportation Research Part C: Emerging Technologies*, vol. 90, pp. 226–246, 2018.

[101] E. S. Bentley, J. Suprenant, and S. Reichhart, "Vehicular data offloading for resource-limited delay tolerant networks," in *2017 IEEE Vehicular Networking Conference (VNC)*, Nov 2017, pp. 243–246.

[102] A. Thaduri, D. Galar, and U. Kumar, "Railway assets: A potential domain for big data analytics," *Procedia Computer Science*, vol. 53, pp. 457–467, 2015.

[103] X. Feng, B. Mao, X. Feng, and J. Feng, "Study on the maximum operation speeds of metro trains for energy saving as well as transport efficiency improvement," *Energy*, vol. 36, no. 11, pp. 6577 – 6582, 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0360544211006013

[104] E. Zochmann, K. Guan, and M. Rupp, "Two-ray models in mmwave communications," in *2017 IEEE 18th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*. IEEE, 2017, pp. 1–5.

[105] M. Heddebaut, V. Deniau, J. Rioult, and C. Gransart, "Mitigation techniques to reduce the vulnerability of railway signaling to radiated intentional emi emitted from a train," *IEEE Transactions on Electromagnetic Compatibility*, vol. 59, no. 3, pp. 845–852, 2016.

[106] S. Trains, "Annual report," Sydney Trains, 477 Pitt Street Sydney NSW 2000, Tech. Rep. Vol. 1, Oct. 2018.

[107] Y. Yao, L. Rao, and X. Liu, "Performance and reliability analysis of ieee 802.11 p safety communication in a highway environment," *IEEE transactions on vehicular technology*, vol. 62, no. 9, pp. 4198–4212, 2013.

[108] T. Wang, P. Li, X. Wang, Y. Wang, T. Guo, and Y. Cao, "A comprehensive survey on mobile data offloading in heterogeneous network," *Wireless Networks*, vol. 25, no. 2, pp. 573–584, 2019. [Online]. Available: https://doi.org/10.1007/s11276-017-1576-0

[109] A. Goldsmith, *Wireless communications*. Cambridge university press, 2005.

[110] J. Nocedal and S. J. Wright, "Sequential quadratic programming," *Numerical optimization*, pp. 529–562, 2006.

[111] P. E. Gill and E. Wong, *Sequential quadratic programming methods.* Springer, 2012, pp. 147–224.

[112] B. Ai, R. He, Z. Zhong, K. Guan, B. Chen, P. Liu, and Y. Li, "Radio wave propagation scene partitioning for high-speed rails," *International Journal of Antennas and Propagation*, vol. 2012, p. 7, 2012. [Online]. Available: http://dx.doi.org/10.1155/2012/815232

[113] H. F. Ates, S. M. Hashir, T. Baykas, and B. K. Gunturk, "Path loss exponent and shadowing factor prediction from satellite images using deep learning," *IEEE Access*, vol. 7, pp. 101366–101375, 2019.

Appendix

A Novel Approach for Big Data Classification and Transportation in Rail Networks

Mahdi Saki, Mehran Abolhasan, *Senior Member, IEEE*, and Justin Lipman, *Senior Member, IEEE*

Abstract— This paper introduces a new framework into future data-driven railway condition monitoring systems (RCM). For this purpose, we have proposed an edge processing unit that includes two main parts: a data classification model that classifies Internet of Things (IoT) data into maintenance-critical data (MCD) and maintenance-non-critical data (MNCD) and a data transmission unit that, based on the class of data, employs appropriate communication methods to transmit data to railway control centers. For the transmission of MNCD, we propose a travel pattern method that employs train stations as points of data offloading so that trains can deliver data as well as passengers at stations. The performance of our proposed solution is successfully validated via three various data sets under different operating conditions.

Index Terms— Railway condition monitoring, big data, online data classification.

I. INTRODUCTION

THE development of condition-based monitoring (CBM) systems in the Australian railway industry with over 33 thousand kilometers operational rail track has received the highest investment priority till 2040 [1], [2]. As will be described in the following, RCM systems will deal with of Big Data problem in the future because these systems meet the three aspects of variety, velocity and volume [3]. Future data-driven RCM systems will be strongly reliant on the huge amount of data received from heterogeneous IoT devices (variety) that are widely distributed throughout the rail network [4]–[6]. In order to have a sense of the volume of available data that needs to be processed, Fumeo *et al.* [7] have presented an example showing that the volume of collected data from only one vibration sensor of a train bearing with sampling rate of 25.6 KHz (velocity) will be as big as 10 TB only for one train per eight hours of its operation. Consequently, the amount of gathered raw data from all sensors distributed throughout a long moving train can be as huge as several hundreds of Tera Bytes (volume). Therefore, finding appropriate solutions for storage or transmission of such massive data will be necessary.

Currently, there is not any cost-effective communication network to handle such massive data traffic [8]. Irrespective of future high capacity data transport methods such as 5G (expected after 2020), we will still have the challenge of network coverage as railway networks are widely distributed all around the country and providing full network coverage for all areas especially for remote areas will be very costly and difficult [9]. Furthermore, assuming that a high capacity data transmission network is available to send all data to data centers, real-time processing of such huge unclassified data sent from all trains will not be efficient [10]. On the other hand, long-time storage of large scale data on-board with current storing technologies will be costly and affected by limitation in storage space [11].

In this work, we will propose a new framework which will innovatively change the vision of Big Data in rail networks. The proposed framework, which can be implemented in an on-board IoT gateway, will consist of two main parts: data analysis and data transmission. In data analysis part, raw data collected from heterogeneous on-board sensors will be analyzed and classified. Then, based on the classification of data, appropriate transmission solutions including real-time or delayed protocols will be applied for the transportation of collected data to control centers. In fact, the reality of having different transmission methods is one of the advantages of data classification prior to its transportation (edge processing). Performing data classification before its transmission will allow data centers to receive data in a classified manner instead of having a huge amount of unclassified raw data which will be extremely computationally expensive to process. This also causes a significant reduction in the capacity and bandwidth of real-time communication networks with railway centers as only critical data needs to be transmitted in real-time.

The main contributions of this work are as follows:

- We propose an on-board edge processing scheme that will handle the whole data management process including pre-processing, classification and multi-mode transmission of data in one integrated unit. This means that the proposed scheme will not only be able to classify the sensor data, but it will also provide the method of data transmission based on the classification of data.
- We develop a classification model that can be shortly re-trained in an online manner. Due to this ability, the developed classification algorithm can be quickly implemented in any train with different specifications.

A Big Sensor Data Offloading Scheme in Rail Networks

1st Mahdi Saki
School of Electrical and Data Eng.
University of Technology Sydney
Sydney, Australia
Mahdi.Saki@student.uts.edu.au

2nd Mehran Abolhasan
School of Electrical and Data Eng.
University of Technology Sydney
Sydney, Australia
Mehran.Abolhasan@uts.edu.au

3rd Justin Lipman
School of Electrical and Data Eng.
University of Technology Sydney
Sydney, Australia
Justin.Lipman@uts.edu.au

Abstract—In this paper, we propose an offloading scheme to transfer massive stored sensor data from rolling stock to railway data centers. We apply a delayed offloading strategy for non-critical stored data assuming that the critical data has been already separated through an appropriate edge processing task and has been sent via a real-time communication such as cellular networks. We propose train stations as potential and feasible spots for data offloading via available wireless local area networks (WLAN) such as existing WiFi network at stations. Thus, stations will not only be the places of passenger exchange but also data exchange. We develop an analytical model customized for the proposed offloading strategy in rail applications. Then we validate the performance of our model through simulation in various scenarios in Omnet. The simulation results shows an accuracy of %98.67 for the proposed analytical model with reference to the simulation results in Omnetpp. Additionally, by using our proposed scheme, we can theoretically offload up to 5.43 GB per each stopping station.

Index Terms—big sensor data, delayed offloading, IEEE 802.11

I. Introduction

Future trains will be equipped with many sensors that continuously sense and generate massive IoT (internet of things) data [1]. According to [1], the amount of sensor data produced by only one sensor for sensing the vibration of just one wheel bearing in a train will be as huge as 10 TB during eight operating hours. Thus, for a train with many parts that will be sensed by wide variety of sensors [2], the created data amount will be extremely massive and transmission of such data into data centers will be a challenge.

Based on the risks for passengers and rail equipment, the collected sensor data is classified into two classes including critical data and non-critical data. The critical data can cause serious damages for both people or rolling stocks and should be declared immediately. However, the non-critical data is used for long time analysis and can be evaluated by delay. As trains operate in normal conditions for most of the time, the amount of critical data is tiny and the main part of sensor data is composed of non-critical data. Therefore, if we could classify the sensor data through an appropriate edge processing task (this is the subject of our another work

which is currently accepted and is under final review), we will be able to employ different communication strategies to transfer critical and non-critical data to railway data centers. In this case, it is feasible to send the tiny amount of critical data in a real-time manner (e.g. via cellular networks) while temporarily store the non-critical data and deliver it later via an appropriate offloading strategy [3]. In this way, we will significantly reduce the data traffic over expensive and infrastructure-based communication networks (such as cellular or satellite networks) by offloading the massive part of data through an available cheap WLAN's channel such as WiFi networks at stations (which approximately has no cost).

This is the idea behind our current work which based on that, we propose train stations as potential spots to offload the delay-tolerant non-critical sensor data. In this way, stations as grounded infrastructure, has the feasibility to provide more powerful computation and communication capabilities for our offloading task. Additionally, if we employ the available channels of WLAN in stations, this will cause large cost saving because we will no longer need to install any rail/road side units (RSU). The proposed offloading method will be a train-to-station (T2S) communication between on-board units (OBU) in trains and a data sink system in stations.

Therefore, the main contributions of this paper is:

- we propose a novel scheme for offloading of delay-tolerant part of IoT data in rail networks,
- we develop an analytical model for the proposed offloading scheme that can model the data offloading task for passing stations as well as stopping stations,
- we provide an integrated equation that can estimate the total offloading capacity for a given train during its trip between two terminals including stopping and passing stations
- we embedded the offloading model with a rate control algorithm that enables the data to be offloaded even with the minimum WiFi signal power and therefore, makes the offloading capacity maximum.

The rest of the paper is organized as follows. Firstly, a short literature review is presented in Section II. Then, we explain our proposed offloading scheme in Section III. In Section IV, we develop an analytical model for the proposed offloading

Mobility Model for Contact-Aware Data Offloading Through Train-to-Train Communications in Rail Networks

Mahdi Saki, *Member, IEEE*, Mehran Abolhasan, *Senior Member, IEEE*,
Justin Lipman, *Senior Member, IEEE*, and Abbas Jamalipour, *Fellow, IEEE*

Abstract—In this paper, we propose a novel mobility model providing train traffic traces essential for train-to-train communication models. As the proposed mobility model works only based on trip timetables and train timetables are currently available in real-time, the produced mobility traces will be also in real-time. Additionally, as no GPS module is used in this method, our proposed model can provide a practical solution when signal from GPS or Assisted GPS is poor or unavailable such as in urban area or inside tunnels. Furthermore, as we used an energy optimization function, the proposed mobility model will provide a guidance trajectory for trains to have an energy-optimized operation. We also develop an algorithm that can determine the specifications of contacts between trains based on the traffic traces obtained from the mobility model. Such specifications includes duration, rate and location of train contacts used for estimation of data exchange capacity between trains through train-to-train communications. We validate our proposed model using data collected from Sydney Trains of Australia. The results obtained from our proposed model show over 98 percent accuracy in comparison with the real data collected via a GPS module from Sydney Trains.

Index Terms—Train mobility model, train-to-train contact model, train-to-train communication, intelligent transportation system, vehicular delay tolerant data, sensor data offloading, train energy-optimized operation.

I. INTRODUCTION

INTELLIGENT transportation systems (ITS) can generate several Petabytes of data from various IoT devices such as condition monitoring sensors and surveillance cameras [1]–[3]. To give a picture of such data, [4] provides a calculation that shows the size of vibration data relating to only one bearing of a train can be over one terabyte per hour. Such massive data should be transferred to data centers for further analysis using big data analytics (BDA) to improve operation and maintenance services of a rail network [5]–[7]. Since the current cellular networks cannot transfer such huge amount of data in an efficient, cost-effective and reliable manner, transmission of that data from ITS to data centers will be a significant issue [8], [9]. Even with the new emerging cellular-based vehicular communication methods such as LTE-V2V or 5G-V2V [10]–[12], having an alternative method due to the rapidly growing demand for cellular networks specially during peak times can be highly beneficial [13]. Rail transportation systems (RTS) as one of the main type of ITS is not also an exception from such massive issue [14].

There are two types of IoT data in RTS include *'mission-critical'* data (MCD) and *'mission-non-critical'* data (MnCD). MCD is related to safety issues and must be immediately informed and transferred to control centers. However, MnCD will be used for improvement of future maintenance and operation services and therefore, can be delivered with delay. Assuming every train has an appropriate edge processing unit that can classify IoT data (collected from various sensors along a train) in a real-time manner, the MnCD can be distinguished and separated from MCD [15]. In this way, we can safely use delayed offloading methods for MnCD.

This work is part of a larger concept where the main idea is to use the trains as carriers of data in addition to passengers. Therefore, in our three previous works, we firstly proposed a classification algorithm that could determine the class of data [15]. Then, we developed models for transmission of delay-tolerant part of data using train-to-station (T2S) and train-to-wayside (T2W) communications in [16] and [17], respectively. Therefore, as a continuation of the previous two works, in this paper, we propose train-to-train (T2T) communications as well as T2S communications to increase the amount of data offloaded through moving trains. In this way, not only the amount of offloaded data will be increased, but it will also provide faster data transfer for applications such as a request-based data demand with restricted delay. Such data can be part of a video stream from a surveillance camera in a moving train requested by a police officer. In such situation, assuming lack of reliable and robust cellular network (that is quite common in the harsh environment of rail tracks), a T2T data exchange strategy can be a vital solution by forwarding

Mahdi Saki, Mehran Abolhasan, and Justin Lipman are with the School of Electrical and Data Engineering, University of Technology Sydney, Ultimo, NSW 2007, Australia (e-mail: mahdi.saki@student.uts.edu.au; mehran.abolhasan@uts.edu.au; justin.lipman@uts.edu.au).

Abbas Jamalipour is with the School of Electrical and Information Engineering, The University of Sydney, Sydney, NSW 2006, Australia (e-mail: a.jamalipour@ieee.org).

A Comprehensive Access Point Placement for IoT Data Transmission Through Train-Wayside Communications in Multi-Environment Based Rail Networks

Mahdi Saki ⬤, Mehran Abolhasan ⬤, *Senior Member, IEEE*, Justin Lipman ⬤, *Senior Member, IEEE*, and Abbas Jamalipour ⬤, *Fellow, IEEE*

Abstract—In this paper, we propose three algorithms for placement of access points (APs) for the purpose of data transportation via train-to-wayside (T2W) communications along a rail network. The first algorithm is proposed to find the minimum number of APs so that the path-loss (PL) does not exceed a desired threshold. Through the second algorithm, the most optimal places for a desired number of APs are determined so that the average PL is minimum. The goal of the third algorithm is to determine the required number and optimal places of APs in a rail network. Furthermore, we propose a model to consider the effects of changes of communication characteristics on the efficiency of the network in different environments. Through such model, the algorithms proposed for placement of APs can be used in different railway scenarios. The proposed algorithms are validated through extensive simulations in Sydney Trains of Australia. The simulation results show that the proposed approach can improve the efficiency of the system at least 21% and up to 165% within 10 different scenarios. We also show that we can approximately transmit over 250 Gigabit data through T2W communications over common WiFi networks.

Index Terms—Access point placement, train-to-wayside communications, data transportation, railway scenarios, rail networks, IEEE802.11, WiFi.

I. INTRODUCTION

WITH the advancement of data-driven intelligent transportation systems (ITS), the amount of data generated by IoT devices is significantly increasing. Therefore, it is essential to find a reliable and cost-effective communication method for exchange of such massive data between ITS and data centers. Cellular communications specially upcoming 5G as the latest revision, can be one of the main solution for such data exchange.
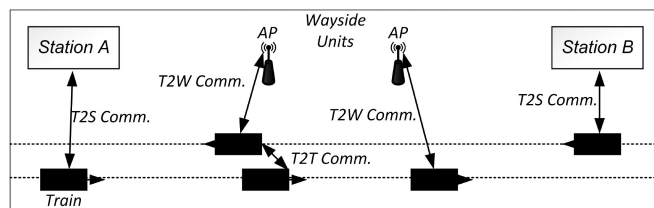
Fig. 1. Three different short-range communication methods in rail networks.

However, due to rising public demand for cellular networks specially during peak hours, exchange of data through other communication platforms in a reliable and cost-effective manner will be highly beneficial [1].

In our recent works, we proposed models for data transportation in rail networks through train-to-station (T2S) and train-to-train (T2T) communication methods [2]. We used T2S and T2T communication methods for transmission of delay-tolerant part of data assuming that the critical part of data had been already realized via an appropriate classification scheme [3]. Fig. 1 shows such different communication methods in rail networks.

In this work, we enhance the previous methods by proposing train-to-wayside (T2W) communications. In this way, the data exchange process can be performed not only partially within T2S or T2T communications but also along the whole rail track and in a continuous manner. To implement such method, we assume that there are wayside units along the rail that can exchange data with trains via an appropriate wireless local area network (WLAN) like WiFi. Then, to have an efficient data exchange, we develop three different algorithms to optimally place the APs of those wayside units in various conditions. All the three algorithms can accurately adapt to any rail track with different geometric paths by using the related GPS data. We also propose a method to model the change of communication characteristics when trains pass through various environment-based scenarios. Assuming several railway scenarios include urban, suburban, rural, cutting, viaduct, tunnel and river, such model cannot only simulate a rail line with single scenario but also a line with multiple scenarios. The main contributions of the current work are as follows: