UKRAINIAN CATHOLIC UNIVERSITY

MASTER THESIS

BLE Mesh Reliability Optimization using Neural Networks

Author:
Oleksandr BRATUS

Supervisor:
Dr. Oleg FARENYUK

A thesis submitted in fulfillment of the requirements for the degree of Master of Science

in the

Department of Computer Sciences
Faculty of Applied Sciences

Lviv 2021

Declaration of Authorship

I, Oleksandr BRATUS, declare that this thesis titled, "BLE Mesh Reliability Optimization using Neural Networks" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:

UKRAINIAN CATHOLIC UNIVERSITY

Faculty of Applied Sciences

Master of Science

BLE Mesh Reliability Optimization using Neural Networks

by Oleksandr BRATUS

Abstract

The Bluetooth Low Energy (BLE) Mesh network technology is one of the newest technologies in the wireless communication domain. Due to low cost and low power consumption, it has already become widespread and has the potential for a wide range of applications.

However, the flooding algorithm on which based BLE Mesh data transmission process impacts strongly on networks reliability. Because improper network setup can be critical to ensuring sufficient network reliability, it is necessary to be able to predict the network reliability in order to be able to reconfigure the network to improve its reliability.

In this master thesis, we propose neural network approaches that predict the reliability of both the entire network and its individual nodes. Presented results demonstrate that trained neural networks are scalable by providing high accuracy of predictions on networks of different sizes.

Acknowledgements

I want to thank my supervisor, Oleg Farenyuk, who provided a lot of useful advice, ideas, and interesting discussions. I would like to express my appreciation to Igor Kravets for providing consultations and sharing his expertise in the BLE Mesh domain. Special gratitude goes out to Oleksandr Pryymak for his constant feedback and for bringing his experience in graph neural networks in particular. I would also like to thank the Ukrainian Catholic University and Oleksii Molchanovskyi for the Master Program that opened me up to the world of Data Science.

Finally, I must express my gratitude to my family for providing me with absolute support. Without all of you, this work would never be finished.

Contents

List of Figures

List of Tables

List of Abbreviations

BLE	**B**luetooth **L**ow **E**nergy
BER	**B**it **E**rror **R**ate
GCN	**G**raph **C**onvolutional **N**etwork
PER	**P**acket **E**rror **R**ate
PDR	**P**acket **D**elivery **R**atio
SINR	**S**ignal-to-**I**netrference-plus-**N**oise-**R**atio

Dedicated to my family

Chapter 1

Introduction

1.1 Motivation

In recent years, much attention has been paid to wireless communication networks [Zhang, Patras, and Haddadi, 2018]. Along with a wide range of wireless network technologies such as Bluetooth, WiFi, Near-Field Communication (NFC), mobile networks (e.g. 3G, 4G and 5G), Bluetooth Low Energy (BLE) technology [Bluetooth SIG, 2010] has become one of the most widely-adopted technologies due to high energy efficiency and widespread availability in user equipment. BLE technology is suitable for small portable and autonomous devices. For example, it makes possible to coordinate real-time networks of wearable devices, health care or home automation networks just by smartphone since almost every smartphone is equipped with this technology.

The BLE provides communication between directly connected devices located close to each other within the distance typically limited up to 50 meters, so it is used for short-range communication. However, this is one of the main limitations of BLE use. Including, for this reason, the technology fails to realize its full potential, despite the widespread. Therefore, in July 2017 was released Bluetooth Mesh [Bluetooth SIG, 2017], that wholly based on the BLE protocol stack. BLE Mesh neworks can connect thousands of devices providing transmission of only short messages between nodes with relatively small energy consumption.

Constructing such a large mesh network is important to ensure the high reliability of its operation, namely, a high packet delivery ratio. Furthermore, since building a network takes a lot of time and effort, it is desirable to model such a network to assess its reliability and optimize weaknesses. To our knowledge, some studies [Liu and Cerpa, 2014, Ateeq et al., 2020] predict the reliability of connections between specific nodes, but they do not consider the overall reliability of the entire network. To fill this gap, in this work, we propose an approach that can predict the reliability of each of the network nodes and the entire BLE Mesh network. With this approach, it will be possible to predict network reliability and decide on changing its settings to improve its performance.

1.2 The proposed method

Taking into consideration all of the above, we propose the following:

- applying neural network techniques for better evaluation of networks reliability. In particular, we will solve a node regression task and compare the performance of the regression model on networks, differ by sizes and number of nodes;

- checking the possibility of applying trained models.

The data used in this work are synthetic. To obtain it, a dynamic simulator was used, a detailed description of which is given in chapter 4.

1.3 Goals of the master thesis

1. To provide an overview of previous work related to the analysis and evaluation of BLE Mesh networks and machine learning methods to predict the reliability of data transmission in wireless networks.
2. To apply neural network model for a prediction of the reliability of BLE Mesh networks.

1.4 Thesis structure

The remainder of this thesis is structured as follows:

- Chapter 2 contains an overview of the wireless communications domain and a complete description of the BLE Mesh Standard.
- Chapter 3 presents work related to the BLE Mesh domain and applying machine learning methods in wireless communications.
- Chapter 4 introduces the data used throughout this work.
- Chapter 5 describes the basis of neural network methods and proposes our approach.
- Chapter 6 discusses solution experiments and corresponding results.
- Chapter 7 summarizes our contributions and list the directions for future work.

Chapter 2

Background

This chapter contains a wireless communication domain overview and a detailed description of BLE Mesh technology. Mainly, we focused on the importance of network topology and introduce the concept of network reliability. To provide a clear explanation of the features of the networks studied in this paper, the principles of BLE technology and the process of transmitting messages over the BLE Mesh network are fully explained.

2.1 Wireless communication networks

In recent years, wireless communication technologies have made significant progress. Many of them are involved in implementing the Internet of Things (IoT) concept that includes residential, industrial, commercial, healthcare, military and many others applications. By active involvement in research and exploration of industry and academia, IoT devices and applications have both increased and diversified exponentially.

Nowadays, wireless technologies are extremely heterogeneous in terms of protocols, performance, reliability, latency, cost and coverage [Cilfone et al., 2019]. Therefore, each of wireless technologies may be optimal depending on the properties of the protocol, the proposed network topology, environment and network operation scenarios and other factors [Chakkor et al., 2014].

2.1.1 Wireless network topologies

One of the basic characteristics of the network is its topology. Wireless networking topologies can be generally divided into four types: one-way, bi-directional, star and mesh networks [Silicon Labs, 2013]. Since the first two provide a connection only between the two nodes, they are fundamental for constructing the last two [Farej and Abdul-Hameed, 2015].

A star topology is a topology in which all nodes are individually connected to a central hub (e.g. Wi-Fi or mobile networks). This topology is attractive because of its simplicity, but at the same time, it has some disadvantages. For example, simultaneous transmission of data from a large number of nodes in the network can cause traffic overload through the central hub. As a result, central hub failure leads to the failure of the entire network.

Unlike star-based networks, mesh-based networks (e.g. ZigBee, Thread, BLE technologies) allow communication of any nodes with each other within their range of communication. The possibility to transmit data from one node to many other nodes at the same time provides a robust connection between all nodes, so the failure of one node has little effect on the entire network. However, redundant connections

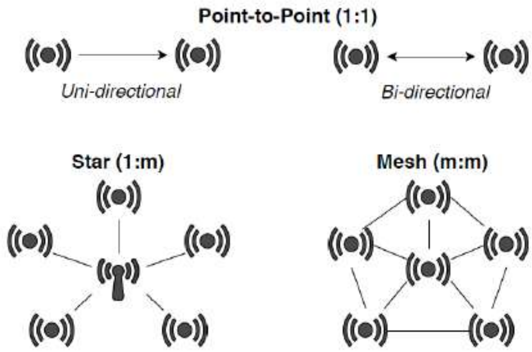


FIGURE 2.1: Network topologies

Leon and Nabi, 2020

between nodes carry the risk of uncontrolled data transmission, so setting up and monitoring such mesh networks is critical.

2.1.2 Reliability

Reliability is among the key performance requirements for many IoT applications. The reliability of a wireless network is formally defined as the network's ability to perform satisfactorily during its mission time when used under the stated conditions [Kuo and Zuo, 2002]. By stated conditions, we mean clearly defined environmental conditions in which the network is deployed and the protocol parameters under which the network operates.

Based on the definition, reliability can be considered from two different perspectives [Deif and Gadallah, 2017]. On the one hand, the network must be guaranteed to function for a certain period of time, so network reliability is related to the concept of network lifetime which is directly affected by the level of energy consumption of each node in networks with energy constraints. On the other hand, the network must ensure a sufficiently high rate of successful data transmissions (in other words, adhere to a certain low threshold for data packet loss).

Later in this work, we determine that each node of the network is guaranteed to operate for a certain period, and therefore the issue of network lifetime is not considered. That is why we focus on the second aspect of reliability and determine the ratio of successfully delivered packets as the main characteristic of the network.

2.1.3 Physical model

Although wireless technologies are very different, they are all equally subject to the physical processes of wireless transmission of information. Based on the wireless communication theory [Rappaport, 1996; Ahlin, Zander, and Slimane, 2006], we introduce a model of data transmission between nodes in the network. We consider wireless communication channel between two nodes. Firstly, to calculate how much of the transmitted power P_{Tx} actually ends up at the receiving side of the communication link (due to propagation) is used the simplest free-space path loss model (FSPL) [Vihlborg, 2011]:

$$L(d) = \left(\frac{4\pi d}{\lambda}\right)^2 \tag{2.1}$$

where d is distance between transmitter node and receiver node.

Then power of received signal P_{Rx} is defined as:

$$P_{Rx} = P_{Tx} \cdot L(d) = P_{Tx} \left(\frac{4\pi d}{\lambda}\right)^2 \tag{2.2}$$

where λ is wavelength of the wireless communication channel.

While describing a wireless communication network, it is crucial to take into account internal (from other nodes transmitting on the same channel) and external (from existing wireless networks at the same frequency) interference. Given impact of interference, it is necessary to define a way to measure the quality of a communication link, the so called *Signal-To-Interference-Plus-Noise-Ratio* (SINR) of the link [Iyer, Rosenberg, and Karnik, 2009]. The SINR is defined as:

$$SINR = \frac{P_{Rx}}{I_{int} + I_{ext} + N} \tag{2.3}$$

where I_{int}, I_{ext}, N are power of the internal interference, power of the external interference and power of noise at the receiver respectively.

Then theoretical probability of a bit error in any packet transmission is defined as *Bit Error Rate* (BER) as a function of SINR:

$$BER = \frac{1}{2} \cdot erfc\left(\sqrt{\frac{SINR}{2}}\right) \tag{2.4}$$

where $erfc(\cdot)$ is the complimentary error function.

Finally, we can measure theoretical probability of a failed packet transmission (probability that none of the bits of the packet will be damaged) as *Packet Error Rate* (PER):

$$PER = 1 - (1 - BER)^n \tag{2.5}$$

where n – size of the packet in bits.

2.2 BLE Mesh technology

The Bluetooth Low Energy (BLE) technology was officially introduced in 2010 with the v4.0 of the Bluetooth Core Specification [Bluetooth SIG, 2010] for short-range low-power wireless communication. Over the past few years, it has become one of the most widely-adopted short-range technologies thanks to its simplicity, low-power consumption, low-cost and robustness. Nevertheless, BLE topology has been restricted to point-to-point and one-to-many until the introduction of Bluetooth Mesh Profile in 2017 [Bluetooth SIG, 2017]. The Bluetooth Special Interest Group (SIG) has defined the mesh profile as a networking technology built on top of the BLE protocol. It allows many-to-many communication of up to a theoretical maximum of 32,767 nodes in 4,096 possible sub-networks.

2.2.1 BLE protocol

Bluetooth mesh is designed to be used without the need to establish connections among devices in the network. To achieve this connection-less approach, all data transmissions in a BLE Mesh network are broadcast under the advertising/scanning scheme of BLE. Basically, BLE protocol uses only three broadcasting channels (2402,

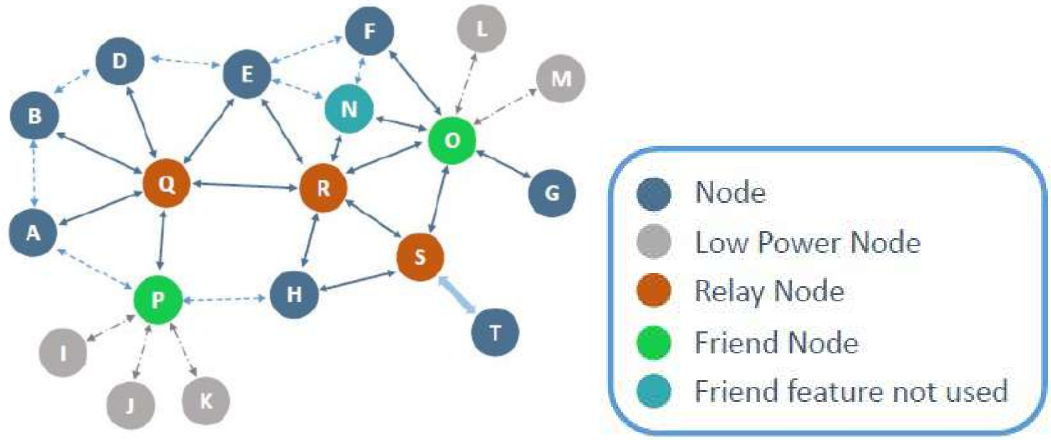


FIGURE 2.2: Bluetooth Mesh node types

Rondon et al., 2020

2426, 2480 MHz). Each broadcast transmission is sent using those three channels one after another during an advertisement event to have a higher probability of messages reaching a receiver. The scanner nodes scan a channel during a time window and then switch into another channel – it means they do not scan a channel all the time. BLE nodes are not synchronized to each other, so each node has random access to the channels. This avoids energy losses when the node is in a state of waiting for a long time. However, it is worth noting that such behaviour of nodes can cause frequent collisions, leading to the overlap of messages, hence losing information.

2.2.2 Bluetooth Mesh topology

Unlike other technologies such as WiFi or ZigBee, until 2017, the BLE protocol stack used a simple point-to-point network topology enabling one-to-one device communications. In contrast, a mesh network has a many-to-many topology and is an attractive alternative to traditional centralized star network topologies. Therefore, the release of BLE Mesh in 2017 was a necessary step that allowed data transmission between pairs of nodes in a dynamic and non-hierarchical way.

All nodes in BLE mesh can transmit and receive messages, but different features allow each node to play notable roles in the network. There are four types of nodes in BLE Mesh networks: proxy nodes, friend nodes, low power nodes, relay nodes. Proxy nodes can receive messages from a non-mesh-supported BLE device and retransmit them over the BLE Mesh network. The primary function of friendly nodes is storing and transmitting messages to their corresponding low power nodes. The friendship mechanism between these nodes is one of BLE Mesh Standard's latest innovations, which allows the low power node to be asleep, conserving energy, and being activated in a case when requested by a friendly node. Finally, relay nodes can receive and retransmit Bluetooth Mesh messages – they provide connectivity between non-directly connected nodes, so the end-to-end communication range is extended far beyond the radio range of each node. This is one of the most important features since the relay process makes Bluetooth mesh multi-hop communication possible.

2.2.3 Managed flooding

The Bluetooth Mesh transmission model is based on managed flooding, which defined by Mesh Standard. By this model, every node within the radio range of the transmitter receives all its messages. If a receiver is a relay node, it retransmits messages to nodes more distant from the transmitter. Compared to other wireless protocols, this model does not require complex routing tables or any routing protocols, reducing the required memory amount and defining the model as the simplest among any other protocols in the IoT context [Darroudi and Gomez, 2017]. As a result, using a flooding algorithm results in a highly resilient and reliable network due to messages' movement to their destination via multiple paths through the networks.

However, the uncontrolled flow of resenting messages through the relay nodes results in a potentially high risk of congestion and packet loss, also known as the broadcast storm problem [Ni et al., 1999]. Therefore, some measures optimize the way flooding works in Bluetooth mesh networking:

- Time-To-Live (TTL) limitation sets the maximum number of hops over which message is relayed before arriving at its destination node.
- Message cache contains all of the recently seen messages. The node compares the newly received message with the already processed ones and decides to ignore further or forward it. This method reduces message forwarding without reducing network capacity and reliability.

Chapter 3

Related Work

This chapter aims to inform a curious reader about previous research work. Firstly, we provided a review of researches on the BLE Mesh Standard. The vast majority of them explore the possibilities of BLE Mesh networks, focusing on strengths and weaknesses. In addition, there are works dedicated to the optimization of the BLE protocol parameters in mesh networking, and works analyzed the influence of the BLE mesh network topology on its reliability.

At the same time, numerous studies in the literature estimate and optimize the reliability of connections in wireless networks using machine learning methods, particularly neural networks, and show significant results.

3.1 BLE Mesh evaluation and optimization

Because the BLE Mesh technology is relatively new, the first studies were dedicated to its analysis and evaluation. Darroudi and Gomez, 2017 comprehensively reviewed the state of the art BLE mesh network solutions. Their conclusions contain a detailed description of the advantages and disadvantages of the BLE mesh networking. Baert et al., 2018 provided a detailed overview of how the BLE Mesh standard operates, performs, and tackles other BLE mesh networking issues. After evaluating the latency performance of both dense and sparse BLE-based mesh networks, the authors conclude requiring an advanced management mechanism for optimizing the performance of the mesh network.

As BLE Mesh built on top of BLE technology, some researchers have figured out how to configure BLE protocols parameters to optimize mesh network performance properly. Liendo et al., 2018 proposed an approach to obtain the appropriate configuration of the three time-parameters of the BLE protocol (scanning interval, scanning window and advertising interval). Their method increased up to 89 times the battery lifetime of low-power devices by optimizing energy consumption and guaranteeing a maximum acceptable critical latency. At the same time, Hernandez-Solana et al., 2020 showed the relationships and interaction between different parameters of the BLE stack and highlighted problems that arise when they are not properly configured.

Rondon et al., 2020 provided general guidelines to ensure the reliability and scalability of the mesh network. For this, the authors considered such network performance indicators as packet loss rate, the time delay, and packet errors per unit time. They conclude that the correct operation of the network protocol strongly depends on each of the time parameters' mutual settings, so the appropriate setting of both advertising and scanning processes has a decisive influence on transmitting messages. However, experiments have shown that scalability remains problematic for BLE Mesh networks due to message flooding.

Other comprehensive experiments to measure BLE Mesh protocol performance and capacity to deliver messages reliably were performed on a static network with fixed time parameters by Leon and Nabi, 2020. As a result, the paper described the data rate limitations of the technology. It was summarized that when nodes are oversaturated and transmitting messages with a high frequency, the technology cannot provide an acceptable packet delivery performance.

As mentioned previously in subsection 2.2.3, uncontrolled message retransmissions produced by relay nodes can oversaturate the network and reduce its reliability. Therefore, algorithms for finding the optimal location of relays have been proposed. Hansen et al., 2018 considered relay selection approaches, where only a few nodes are relays and can retransmit messages. The authors compared three heuristic algorithms whose task to form a connected set of relays so that each network node has been connected to at least one relay node. The algorithms evaluated are Greedy Connect, K2 Pruning and Dominator!. Each algorithm outputs a fully connected network and can run distributed on each network node without centralized control. To compare the performance of algorithms is used Packet Delivery Ratio (PDR) metric. The authors concluded that each of the algorithms increases packet delivery probability experiencing the conditions of optimal use of each of them. Bęben, Bąk, and Sosnowski, 2020 introduced another relay node management method. To minimize the number of active relay nodes of the BLE Mesh network, the authors formulated the Minimum Relay Tree (MRT) problem and provided an exact solution based on integer linear programming. The authors emphasized the efficiency of the algorithm in dense networks. They concluded method reaches a compromise between the amount of energy consumed and the message delivery ratio.

3.2 Wireless network reliability evaluation using machine learning approaches

With the ever-increasing availability of performance data, data-driven techniques are becoming popular in prediction and optimization tasks. It is observed that many communication parameters influence different performance metrics characterized as essential requirements in wireless network applications.

In particular, some researches have been conducted to predict the reliability of wireless network connections with the Packet Delivery Ratio (PDR) as an apt metric. Liu and Cerpa, 2014 used the physical-layer information of a particular link, that is, Received Signal Strength Indicator (RSSI), Signal-to-Noise Ratio (SNR), and Link Quality Indicator (LQI) as input features to predict the success probability of delivering the next packet. They showed that logistic regression works better compared to naive Bayes classifier and artificial neural networks. At the same time, neural networks demonstrate excellent results in predicting Packet Loss Ratio (PLR) based on the inter-packet interval, the number of nodes, received packets and erroneous packets as inputs [Kulin et al., 2017]. One more paper proposed a deep learning solution for forecasting PDR [Ateeq et al., 2020]. Using a real dataset, contained thousands of configurations of preconfigured stack parameters from different layers, was achieved accuracy up to 98%.

Although the proposed neural network approaches to predict the reliability of wireless communications showed excellent results, they consider only individual connections without evaluating the reliability of the whole network.

One of the most critical parameters that affect network reliability is nodes' transmission power. On the one hand, transmission power minimization solutions aim to reduce energy consumption and, as a consequence, to increase the network's lifetime. On the other hand, the lowest possible transmission power can increase the network's vulnerability to the interference caused by bad SINR [Kim and Kwon, 2008]. Therefore, different algorithms can reduce energy consumption and improve channel capacity, where channel capacity is defined as the highest information rate that can be achieved with arbitrarily small error probability.

There are many transmission power control studies, which mainly focus on improving the sum rate channel capacities in a single-hop ad hoc interference network. Sun et al., 2018 used a deep learning network (following a supervised learning approach) to approximate the classical solution [Shi et al., 2011]. The main advantage of this paradigm is computational efficiency since, once trained, deep networks can run faster than the classical algorithms that they are imitating. Unfortunately, this method has a poor generalization to scenarios out of training examples, and thus, it cannot be scaled on a network of different topologies.

However, Chowdhury et al., 2020 proposed an unsupervised approach that directly employs the optimization objective as a loss function and uses graph neural networks to facilitate generalizability across network topologies. Numerical experiments proved generalizability to unseen scenarios such as different network densities and network sizes.

Chapter 4

Data

In this chapter, we substantiated the expediency of creating a simulator. The principles of its operation and the initial analysis of its performance are described.

In the second part of the chapter, we explain what data obtained from the simulations we use in our work and present the results of comparative analysis of different datasets.

4.1 BLE Mesh simulator

The research of wireless network technologies requires numerous practical experiments. During full-fledged real experiments, it is necessary to rebuild the investigated network each time and change the used data transmission protocol parameters. In most cases, it can be expensive both in financial terms and the time involved. Hence, different simulation tools are developed to check proposed solutions in some test environments.

After analyzing the available open-source simulators that implement Bluetooth mesh networking, we concluded that we could use none of the implementations found. The reasons for this are the lack of documentation or insufficient functionality for the planned experiments. For example, the most thoroughly described simulator is FruityMesh [1]. Although it is based on the BLE protocol, it does not use a BLE Mesh Standard having many distinctive features and cannot be used in our work. In addition, none of the studies reviewed in the section 3.1 and which uses its own simulator for experiments does not provide to its implementation. Thus, we use a BLE Mesh simulator implemented by the UCU Machine Learning Lab [2] in collaboration with Infineon Technologies [3] which can be easily modified according to our needs.

4.1.1 Simulator architecture

The simulation tool has been implemented in Python and belongs to the group of dynamic simulators.

Firstly, an advertising/scanning scheme was implemented in which each node is sequentially switched between two states. Being in one of two states, the node is alternately tuned to each of the three frequency channels, transmitting or receiving a packet. The principle of packet transmission between two nodes is completely based on the physical model described in the subsection 2.1.3. Thus, the packet will be delivered from the transmitter to the receiver if they operate on the same frequency channel simultaneously.

[1] https://www.bluerange.io/docs/fruitymesh/index.html

[2] https://apps.ucu.edu.ua/en/mllab/

[3] https://www.infineon.com/

The simulator allows creating a network of any topology determined by each node's features. Each node is characterized by its location, the ability to retransmit received packets (relay function) and transmission power, which determines the maximum range of packet transmission. Transmitting packets across the mesh network is based entirely on the managed flooding model (subsection 2.2.3).

Since this work examines the network's reliability, it is necessary to identify cases where the packet is scanned by the receiving node but will still be lost. So, packets in the simulator can be lost for the following reasons:

- overlapping packets - in case of simultaneous receipt of two or more packets to the receiver node, they are lost;
- insufficient transmission power - weak received signal strength increases the risk of packet loss.

4.1.2 Simulator setup

Implementation of the simulator allows us to set both the parameters of the BLE protocol and the BLE Mesh standard. However, the aim of this work is not to analyze the BLE protocol parameters. Therefore, these parameters are fixed and listed in the table 4.1.

TABLE 4.1: BLE protocol parameters

Random back-off delay (ms)	0-30
ScanInterval (ms)	20
$T_{interPDU}$ (ms)	1
Packet length (bits)	312

In this paper, we focused on the topology of the studied mesh networks. By simulating networks of different topologies, we can investigate how one or another characteristic of the network affects its reliability. Therefore, for each network simulation, the following network parameters are set:

- number of nodes in network;
- location, transmission power, role (relay/not relay) of each node;
- emission rate (number of randomly generated packets per second);
- simulation duration.

Besides these parameters, parameters such as sensitivity (minimum received signal strength to receive the packet), TTL and message cache are fixed (table 4.2).

Setting all these parameters completely determines the configuration of a particular network, and allows to simulate its operation (to transmit packets for a certain time). By performing simulations, we can obtain numerical indicators that characterize the various properties of this network. After each simulation experiment we get a set of statistical values:

TABLE 4.2: BLE Mesh Standard parameters

Transmission power (dBm)	-30 - -20
Sensitivity (dBm)	-95
TTL	4
Message cashe	32
Emission rate (packets/s)	50

- network PDR;
- PDR of every multi-hop connection between two nodes (as a matrix);
- average packet delivery delay;
- ratio of lost packets.

It is very important to note that in all experiments, we consider squared environments (without loss of generality). Furthermore, the location of each node is set randomly, as well as its transmission power (from a specific range). Thus, due to each node's different transmission power, there may be cases of one-way communication between nodes. Finally, in order for the studied networks to be homogeneous, all nodes are relays.

4.2 Data analysis

The behaviour of the network during the simulation is influenced by many parameters (which we set at the beginning). Some parameters characterize a specific node, which has little effect on the network's overall performance, but some parameters directly impact the network, which we will explore below.

The most important network indicator for us is the Packet Delivery Ratio (PDR) value which means the ratio of the number of packets that have reached the destination node to all created unique packets. It evaluates the reliability of the connection of the network and is a basic metric in this work.

4.2.1 Emission rate analysis

We first investigated the effect of the emission rate on network reliability. To ensure a uniform network load, each node in the network has the same probability of generating a packet at any given time, and we only specify the number of packets to be generated per second.

An experiment was conducted in four networks, different in size of the environment and the number of hosted nodes, and only the emission rate changed during the simulations. The figure 4.1 shows that an increase in the emission rate inevitably leads to a decrease in the PDR. Since the effect of this parameter is transparent and predictable, we fixed its value at level 50 packets/s for further experiments.

4.2.2 Network topology analysis

Depending on the size of the network environment and the number of nodes in the network, the network structure can change significantly, which affects the packet

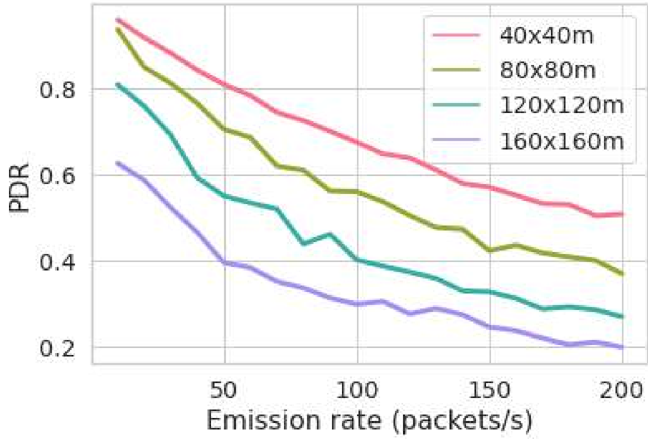


FIGURE 4.1: Emission rate analysis

transmission process. That is why such characteristics of complex network theory [Barabási and Pósfai, 2016] as the average shortest path and connectedness were considered.

In networks, a path is a route that runs along the links of the network. A path's length represents the number of links the path contains. The shortest path d_{ij} between nodes i and j is the path with the fewest number of links. In a directed network (our case), the existence of a path from node i to node j does not guarantee the existence of a path from j to i. Then the average path length is the average distance between all pairs of connected nodes in the network. Because the studied mesh networks provide multi-hop communication, the average shortest path is an informative indicator that gives intuition about the size of the network.

Another essential utility of networks is to ensure connectedness. A network is connected if all pairs of nodes in the network are connected. Conversely, if a network is disconnected, it consists of some components (connected subnetworks). The concept of network connectedness is fundamental by mesh network examination. Since the nodes are placed randomly (which is typical for real networks), on the one hand, with a small number of nodes, cases of disconnected networks are possible. However, on the other hand, too many nodes in the network can be excessive and lead to network congestion in crowded places. Therefore, the influence of the number of nodes on the connectedness of the network in different environments was studied. We calculated the ratio of disconnected nodes in the network that do not belong to the largest connected component of the network.

We conducted some experiments to study the impact of the number of nodes in networks of different sizes ($40 \times 40m^2, 80 \times 80m^2, 120 \times 120m^2, 160 \times 160m^2$) on networks characteristics: ratio of disconnected nodes, average shortest path, PDR.

The figure 4.2 shows that in a disconnected network, increasing the number of nodes reduces the number of disconnected nodes, which contributes to an increase in PDR. However, as soon as the network becomes connected, further addition of new nodes reduces the PDR. As for the average shortest path, with the increase in the number of nodes, it reaches a specific stable value, which does not change with the subsequent addition of nodes, and therefore depends only on the size of the environment and the transmission ranges of nodes.

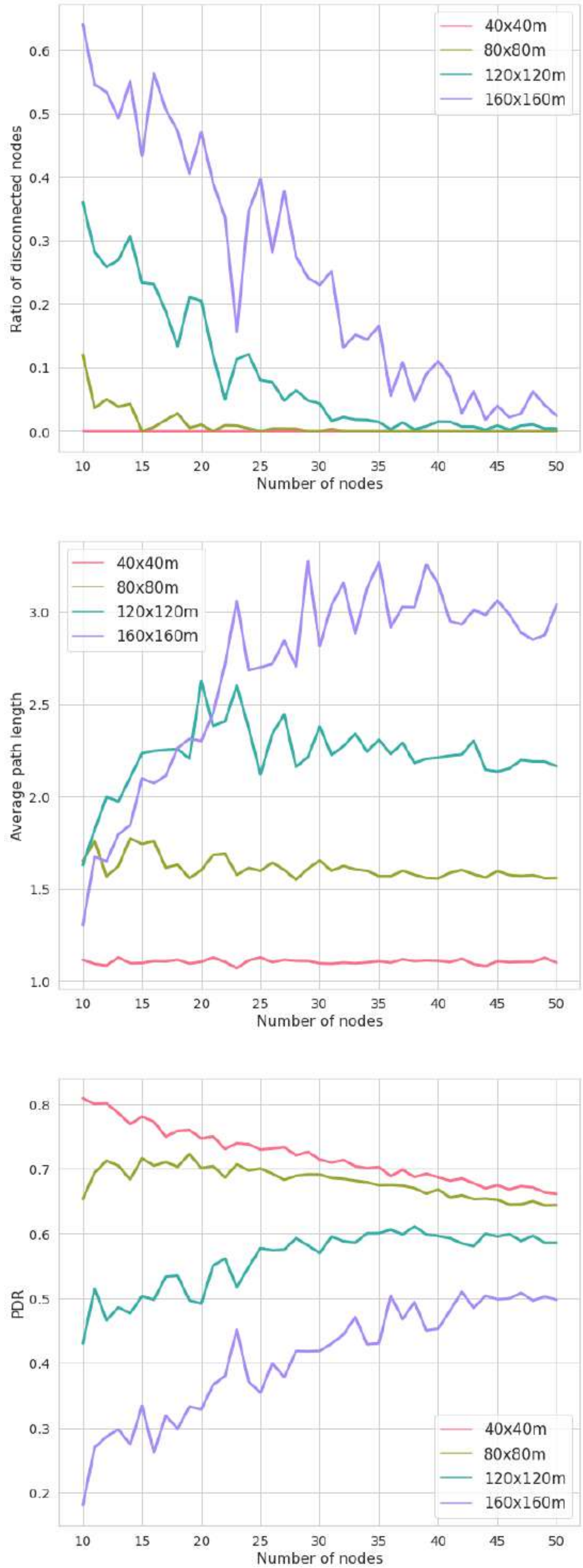


FIGURE 4.2: Network size analysis

4.3 Datasets

Based on the analysis results, we decided to investigate networks that differ in size, number of nodes, and hence, in the average shortest path. Thus, four datasets were simulated. Each dataset consists of a different number of simulated samples (networks), but within one dataset, the size of the networks and the number of nodes are constant. The comparative characteristics of these datasets are given in the table 4.3.

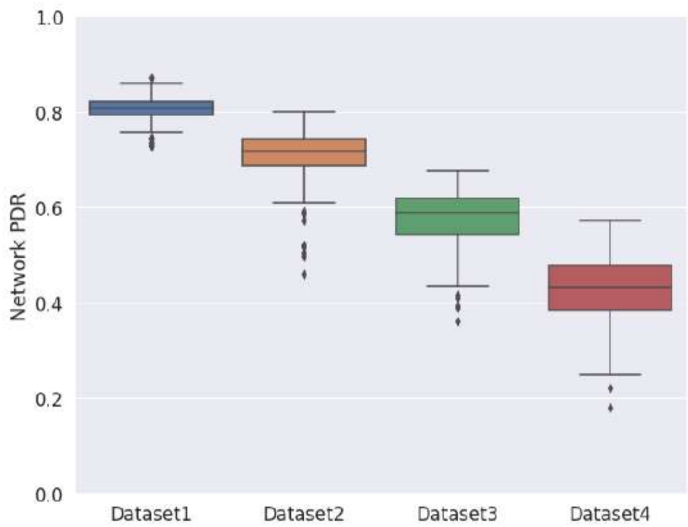
TABLE 4.3: Datasets description

Specification	Dataset 1	Dataset 2	Dataset 3	Dataset 4
Samples (networks)	2000	1000	600	400
Nodes per sample	10	15	25	30
Sizes (m^2)	40×40	80×80	120×120	160×160
Average path length	1.11	1.51	2.15	2.72
Average PDR	0.808	0.710	0.573	0.431

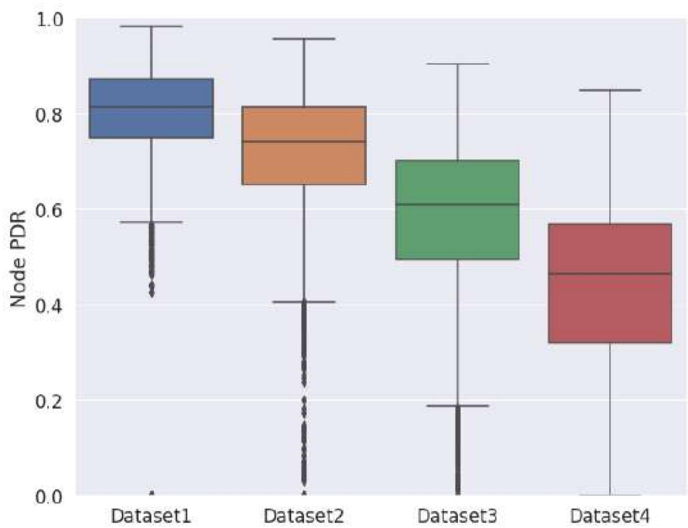
Special attention needs to be paid to the analysis of PDR metric values. The table 4.3 shows that the average PDR values for each dataset are significantly different. After plotting visualizations of the PDR distributions for each dataset using boxplots (figure 4.3a), we saw that the range of PDR values is relatively small, and the number of outliers is small. This means that these PDR values have low variance, so little variability within each of the datasets.

Since after each simulation, we have statistics of each of the broadcast packets, it is possible to aggregate these indicators, not at the whole network level but each of the network nodes level. Moreover, since packets are generated uniformly in a random manner, we can decompose the PDR of the network as the average PDR of all its nodes, where PDR of the node is defined as the ratio of successfully delivered packets generated by this node. Calculating the relationship between the PDR of the network and the average PDR of all nodes of this network, we obtained a correlation coefficient greater than 0.99, which indicates the absolute validity of this approach.

Having plotted similar boxplots for the distribution of PDR nodes (figure 4.3b), we observe distributions with much larger variance, which means nodes in the mesh network of similar sizes can have essentially different PDRs. This is a positive result for further use in our work.



(A) Network PDR



(B) Node PDR

FIGURE 4.3: PDR distributions in each dataset

Chapter 5

Approach

In this chapter, we described the basis of neural network methods, especially deep neural networks and graph neural networks.

We also introduced our approach to solving node regression task. Finally, we described data preprocessing techniques of graph data and proposed deep neural network models.

5.1 Neural networks

The usage of *artificial neural networks* has risen sharply over the past decade. They have recently gained high popularity due to their high-efficiency performance in weather forecasting, speech to text, handwriting recognition, face recognition, and many other problems [Deng and Yu, 2014; Liu et al., 2017]. Therefore, most of the recent state-of-the-art papers include some neural network in their model or pipeline.

The artificial neural network normally consists of the input layer, a hidden layer, and the output layer consisting of neurons [Bishop, 1995]. A neuron is an object that is characterized by its input size, weight vector and activation function. Then each layer is just a collection of neurons that work on the same features of an object. One layer can be considered as a function; after applying it, we get some new space of features. Then we apply another such layer to this feature space. Layers can be of different types, and by combining several different layers according to certain principles, we can get a neural network with good performance. Thus, a neural network is a non-linear transforming function that consists of several functions' sequential application.

5.1.1 Deep neural networks

An artificial neural network that contains multiple hidden layers between the input and output layer is called Deep Neural Network (DNN) [Goodfellow, Bengio, and Courville, 2016]. The most straightforward deep neural network is a *multilayer perceptron* (MLP) [figure 5.1], where all neurons are fully pairwise connected between two adjacent layers. These layers are traditionally called dense or fully connected. Deep neural network propagates signal through layers modelling non-linear relationships between input and output.

The training of the neural network consists of forward and backward passes [Glorot and Bengio, 2010]. On forward pass, the input is propagated through layers, and the error function is calculated. On the backward pass, it figures out how each weight impacts total error computing the gradients via the chain rule, and modifies the network weights to decrease this error. Training continues until the error function converges to zero or training is stopped by other conditions in the perfect setting.

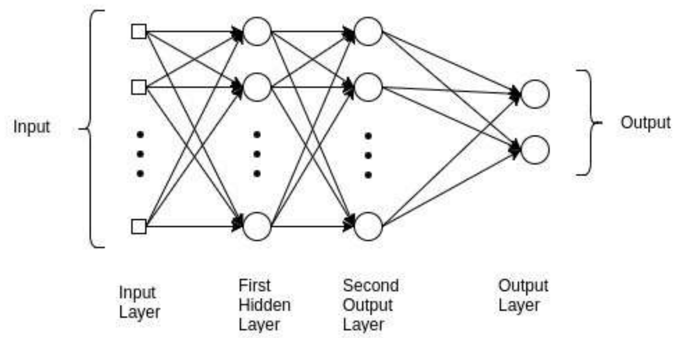
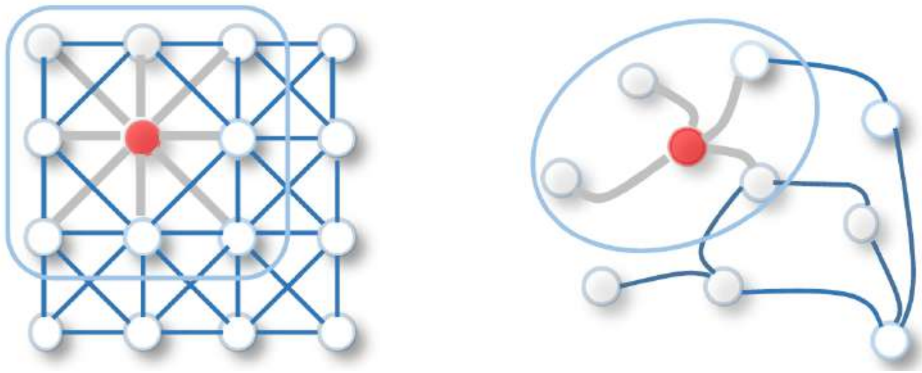


FIGURE 5.1: Multilayer Perceptron architecture

Kain, 2018



(A) 2D Convolution

(B) Graph Convolution

FIGURE 5.2: 2D Convolution vs. Graph Convolution

Wu et al., 2016

Deep neural networks allow investigating hidden patterns of sophisticated features, especially from high-dimensional spaces, which are hard to handcraft otherwise [Hinton and Salakhutdinov, 2006].

5.1.2 Graph neural networks

Recently, the number of applications where data are generated from non-Euclidean domains is dramatically increased. It turned out that traditional neural network methods are not able to effectively process such data [Bacciu et al., 2020]. Therefore, models have been proposed based on existing deep learning approaches but accept inputs in the form of graphs with complex relationships between objects. These methods are called Graph Neural Networks (GNN).

In particular, motivated by Convolution Neural Networks (CNNs) from deep learning, that showed state-of-the art performance in many Computer Vision tasks [LeCun, Kavukcuoglu, and Farabet, 2010], new generalizations and definitions of convolutions have been rapidly developed over the past few years to handle the complexity of graph data [Wu et al., 2016]. A graph convolution can be generalized from a 2D convolution. As illustrated in the figure 5.2, an image can be considered as a special case of graphs where pixels are connected by adjacent pixels. Like 2D convolution, one may perform graph convolutions by taking the weighted average of a node's neighbourhood information.

Each node of graph convolution layer collects features of its neighbours that were propagated through trainable filters (convolutions) - so called, *message passing*:

$$h_v^{l+1} = \phi\left(h_v^l, \Psi\left(\{\psi(h_u^l)|u \in N_v\}\right)\right) \tag{5.1}$$

where N_v - all node v neighbours, h_v^l - node v state at l-th layer, $\{\psi(\cdot)\}$ - *messages* from neighbours, Ψ - aggregation function which is permutation invariant, $\phi\left(h_v^l, \cdot\right)$ - combine function, which update node state.

Based on the equation 5.1, different graph convolution models were proposed depending on the aggregation and combine functions. Kipf and Welling, 2016 introduced Graph Convolution Network (GCN) that is a multilayer network, where each layer can be formulated as:

$$H^{(l+1)} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l+1)}\right) \tag{5.2}$$

where $\tilde{A} = A + I$ - is adjacency matrix with self-connections (I), $\tilde{D} = \sum_j \tilde{A}_{ij}$, W - trainable weights and H is output of previous layer or $H(0) = X$ is input, X - is node features. This model scales linearly in the number of graph edges and learns hidden layer representations that encode both local graph structure and features of nodes.

Such Convolutional GNNs can extract high-level node representations by graph convolution. With a multilayer perceptron as the output layer, GNNs are able to perform node-level tasks (node regression and node classification tasks) in an end-to-end manner [Wu et al., 2016].

5.2 Pipeline

When evaluating the reliability of a BLE Mesh network, it is difficult to specify a feature vector that would fully describe the entire network. Therefore, we propose to consider not the whole network, but each of its nodes separately, providing it with information about a particular local part of the network.

Therefore, we move from the problem of graph regression to node regression, and propose two approaches in its solution using neural networks.

5.2.1 Data preparation

As described in the chapter 4, we have four generated datasets that contain samples (networks), and samples of different datasets differ in size, number of nodes, and hence, in the average shortest path. Each network contains its nodes and existed edges between them. Therefore, we can obtain node features and edge features. Nodes are characterized only by their *transmission power*. However, it can be expressed in two units - in milliwatts (mW) and decibel-milliwatts (dBm) [1] with the following relationship between them:

$$P_{dBm} = 10 \cdot log_{10} P_{mW} \tag{5.3}$$

[1] https://en.wikipedia.org/wiki/DBm

P_{mW} are used in mathematical description of wireless transmissions while P_{dBm} are more informative when comparing different signals with each other due to the nature of the attenuation of the signals in space [*Decibel Tutorial: dB and dBm vs. Gain and Milliwatts*].

Most network information is contained in the edge features. Each edge has the following physical features:

- Received Signal Strength Indicator (*RSSI*) - the main property of wireless link which indicates the strength of the signal from transmitter node and directly affects the reliability of the connection.
- *distance* between transmitter and receiver.
- *BER* defines the error probability of any bit in the packet.

Features aggregation. Since we solve the node regression problem, and most of the information is contained in the edges, we must create feature vectors for each node. To do this, we aggregate the features of all output edges of the node, sort the RSSI values, and rearrange all other features in the order of the corresponding RSSIs. In this approach, the dimensions of the node feature vector will depend on the number of nodes in the network. Therefore, we decided to consider only the five nearest receiver nodes to the transmitter node to ensure equal conditions for each node. Considering transmission power (dBm) of a particular node and sorted edge features of its five nearest receiver nodes, we get a 16-dimensional feature vector of this node.

RSSI distribution. However, information about the five nearest nodes may not fully describe the node and the network topology around it. This becomes especially critical with increasing network size. Therefore, for each node, the cumulative distribution of the RSSI values greater than the sensitivity threshold (-95 dBm) was calculated.

This means that we calculate how many receiving nodes fall within a certain range of a particular node. The values -65, -70, -75, -80, -85, -90, -95 (dBm) fixed ranges were chosen, so we get the absolute distribution as a descending sequence of seven numbers. We normalized each of the sequences separately, and in order to be able to compare them with each other, we also normalized them by the total number of nodes in the network. Adding normalized RSSI distributions of a particular node's neighbours as features, we get a 30-dimensional feature vector of this node.

Train/validation/test split. After generating node features sets, we stack all the nodes of all dataset networks together, obtaining the final dataset for the regression task. To get valid experimental results, we should split each dataset into three non-intersecting subsets. We use the training set to fit the model, validation set to estimate prediction error for model selection and parameter tuning and test set for final evaluation.

There is no direct rule in what proportion should be done such splitting. According to Hastie, Tibshirani, and Friedman, 2001, the typical split might be 50% for training and 25% for validation and testing. Due to the specificity of our domain, we decided to keep 70% of data for training, 15% for validation, and 15% for the test set.

5.2.2 Training models

Multilayer perceptron. We chose the multilayer perceptron as the first proposed model. On each of the four datasets, we trained a separate MLP model, using the data preprocessing techniques from the previous subsection.

After achieving satisfactory accuracy on each of the datasets, we propose to combine all data into one dataset, dividing it into train, validation, test sets in proportion to each of the datasets. Then, after training one more MLP model on the generalized data, we want to test whether this model can show improvements on each of the initial datasets' test sets. Thus, we will test the scalability of this approach.

Graph Convolution Network. Another approach in solving the node regression task is the use of graph neural networks. There is no need to use data preprocessing techniques because we can represent the input data in the form of graphs, separating node features (1-dimensional vectors) and edge features (3-dimensional vectors), thus representing the network in its natural form.

To set the network topology, we use an adjacency matrix A, where nonzero a_{ij} determines the existence of a wireless communication channel from node i to node j.

Based on the vanilla GCN model, we need to improve it because it does not consider the edge features. For this, we aggregate neighbours' node features and edge features separately during the message passing process and stack their representation together into one feature vector. Then, this vector and current node state are combined and update node state.

Specifying the number of graph convolution layers, we indicate how deeply each of the nodes can explore the network. But too few layers will not provide enough information about the surrounding nodes, and too many will aggregate all the information so that each of the nodes will have approximately the same ideas about the network, which will not differ. Therefore, the choice of the number of layers should be made depending on the size of the network and its density.

Chapter 6

Experiments and evaluation

In this chapter, we described experiments to test neural proposed network models for node regression task. Furthermore, we compared how this model deals with data from different datasets.

Finally, we have used trained models to estimate network PDRs and optimize node PDRs.

6.1 Metrics

For evaluation of proposed regression models, we used two basic metrics:

- MSE (Mean Squared Error) [1] metric is used while training and validation processes as loss function and while comparing results of different models:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \tilde{y}_i)^2 \tag{6.1}$$

  where $\mathbf{y}$ – actual PDR values, $\tilde{\mathbf{y}}$ – predicted PDR values.

- MAE (Mean Absolute Error) [2] metric is used to intuitively understand the results of the models because we can naturally interpret this metric:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \tilde{y}_i| \tag{6.2}$$

Because we also want to investigate the strength of the relationship between actual PDR values and predicted PDR values, we also use two statistical metrics:

- Pearson correlation coefficient r_{xy} is a measure of linear correlation between two sets of data $\mathbf{x}$ and $\mathbf{y}$ [Pearson, 1895]:

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \overline{y})^2}} \tag{6.3}$$

  where $\overline{x}, \overline{y}$ – mean values of corresponding sets;

- Spearman's rank correlation coefficient [3] assesses how well the relationship between two datasets can be described using a monotonic function. It is equal to the Pearson correlation between the rank values of those two sets of data.

[1] https://en.wikipedia.org/wiki/Mean_squared_error
[2] https://en.wikipedia.org/wiki/Mean_absolute_error
[3] https://en.wikipedia.org/wiki/Spearman%27s_rank_correlation_coefficient

6.2 Training details

Both models proposed by us in this work are fully connected.

The hyperparameters such as hidden layers dimensions, activation function, learning rate were tuned on the cross-validation technique. Thus, a sigmoid function was used as a nonlinearity activation function. The training was conducted by the Adam optimizer [Kingma and Ba, 2014], with an unchanged learning rate of 0.0001. The ultimate optimization function is mean squared error function. The optimal MLP model had three hidden layers with 100 neurons per layer. Tuned GCN had two graph convolutional layers with 100 neurons per layer.

During training we group the data into the batches – node batches (with the batch size 256) in the first case and graph batches (combining n graphs into one huge graph with n components; with the batch size 16) in the second case.

6.3 Experiments

6.3.1 Node regression task

Experiments results described in this subsection below are presented in the table 6.1.

Baseline. To compare our neural network models with some baselines, we first used basic regression models, such as linear regression, LinearSVR, RandomForestRegressor and GradientBustingRegressor. As expected, ensemble models showed the best results. In particular, RandomForestRegressor, which demonstrated the best MSE and MAE metrics, was chosen as the baseline model. However, providing some additional experiments, we saw that trained on one dataset (on a specific type of networks), such a model can not achieve similar results on networks with other characteristics.

MLP. While implementing our first neural network method were trained four multilayer perceptrons (separately on each corresponding dataset). Also, their results on test sets were worse than baseline results; but after sequentially applying additional data preprocessing techniques (from subsection 5.2.1), better results were achieved.

To check the scalability of this method, we have stacked all train sets into one and trained a generalized MLP model. We trained network for 500 epochs with hyperparameters shown in section 6.2. It can be seen in the figure 6.1 that the model loss decreased on the train set during 500 epochs. However, validation loss decreased until 200 epochs and started to grow after. This indicates model overfitting after the 200-th step. Therefore, as final model we used model trained during 200 epochs.

Furthermore, when checking the model predictions on the each of four test set, slightly worse results were achieved. This means that this approach allows the node to successfully determine the surrounding network topology and issue accurate reliability predictions regardless of the network sizes (in particular those presented in our data).

GCN. Compared to the previous approach, where each node is a separate sample, in the second approach, each node is considered as part of the source network, storing information about its relationship with each neighbor.

By training the graph convolution network using the edge features during the message passing process, we obtained a neural network that converges well but not smoothly. The figure A.1 demonstrates train losses and validation losses on each dataset and indicates a sufficient number of epochs in each case (500, 300, 200, 500 epochs) before models overfit. Although results on the test data are inferior to the

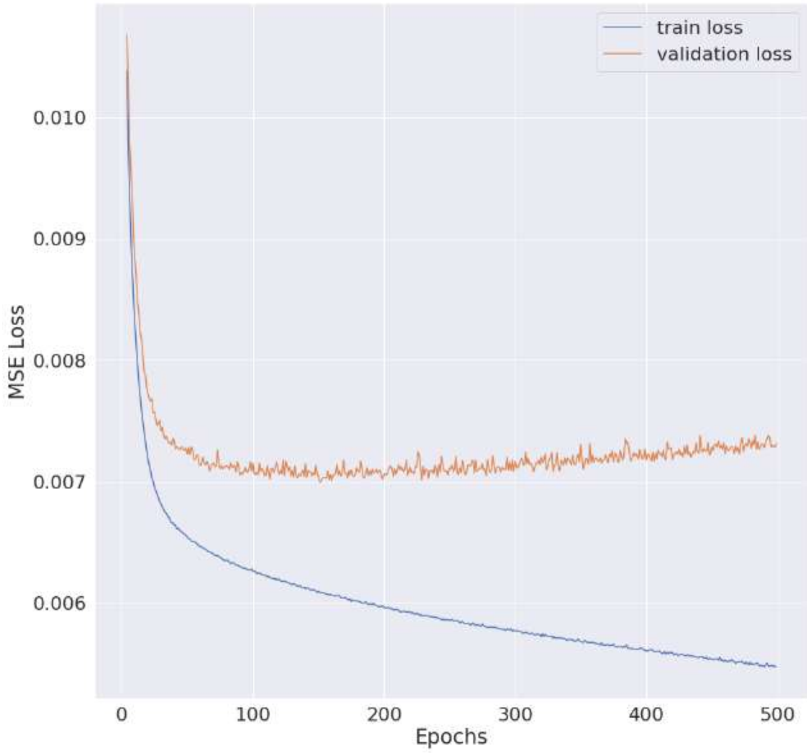


FIGURE 6.1: MLP model losses

previously obtained results, we are convinced that this approach can be applied to this problem. Furthermore, advantages such as invariance to the number of neighbours and scalability due to the increase in convolution layers leave room for further research.

6.3.2 Evaluation

Having obtained an MLP model that can predict the node' PDR, we considered two basic ways to apply this model.

Network PDR estimation. As described in the section 4.3, provided that packets are generated uniformly by each network node, the network' PDR is highly correlated with the average node' PDR. Therefore, having achieved results in predicting node' PDR, we apply these to estimate the PDR of the network.

We have split our data in a graph manner, so all nodes of a particular graph are used only in train, validation or test steps. Using the MLP model (trained on all train data), we averaged node PDRs by corresponding networks and got estimated network' PDR. Actual network' PDR is defined here as PDR of all packets generated in the network. Network' PDR predictions had low MSE and MAE metrics and sufficiently high correlation coefficients comparing to node' PDR predictions (table 6.2).

Node PDR optimization. The low reliability of the node can be caused by the improper placement of the node, which leads to its isolation from the entire network or insufficient transmission power. Because we consider the location of the nodes to be fixed, the only parameter we can change is the transmission power. Based on the simulated data, we are convinced that increasing the transmission power necessarily leads to an increase in the probability of delivering packets generated by

TABLE 6.1: Node regression task. Neural network approaches results

	Pearson	Spearman	MSE	MAE
Dataset 1				
RandomForestRegressor	0.80664	0.79498	0.00291	**0.04216**
MLP	0.79814	0.7923	0.00302	0.04295
MLP with CD features	**0.80673**	0.79235	**0.00291**	0.04241
MLP with CD features trained an all train data	0.8051	**0.79496**	0.00293	0.04255
GCN with edge features	0.79714	0.7939	0.0031	0.04272
Dataset 2				
RandomForestRegressor	0.87151	0.82997	0.00643	0.05677
MLP	0.86224	0.82609	0.00694	0.06044
MLP with CD features	0.87529	0.83422	0.00627	**0.05608**
MLP with CD features trained an all train data	**0.8776**	**0.83825**	**0.00615**	0.05632
GCN with edge features	0.85432	0.83219	0.00739	0.05943
Dataset 3				
RandomForestRegressor	0.84072	0.79795	0.00938	0.07101
MLP	0.83132	0.79575	0.0099	0.07435
MLP with CD features	**0.85168**	**0.81122**	**0.0088**	**0.06882**
MLP with CD features trained an all train data	0.85142	0.80767	0.00891	0.07099
GCN with edge features	0.84103	0.79508	0.00913	0.07198
Dataset 4				
RandomForestRegressor	0.80487	0.74233	0.01196	0.08539
MLP	0.77763	0.73087	0.01342	0.0917
MLP with CD features	**0.81244**	**0.75963**	**0.01154**	0.08351
MLP with CD features trained an all train data	0.80965	0.75679	0.01222	**0.08309**
GCN with edge features	0.80888	0.74635	0.01278	0.08588

the transmitter node. That is why we aim to see if our model can predict an increase in node PDR values with increasing transmission power.

To verify this, in each network of each of the four test datasets, we identified the nodes with the lowest predicted value. Step by step, increasing the transmission power (by 1 dBm) of these nodes to the maximum value (-20 dBm) and leaving all other nodes unchanged, we generated four extended test datasets to test final MLP model on it.

We have considered four cases of possible results with increasing transmission power:

TABLE 6.2: Node PDR predictions vs. Network PDR predictions

	Pearson	Spearman	MSE	MAE
Dataset 1				
Node PDR predictions	0.8051	0.79496	0.00293	0.04255
Network PDR predictions	0.79821	0.75955	0.00029	0.01349
Dataset 2				
Node PDR predictions	0.8776	0.83825	0.00615	0.05632
Network PDR predictions	0.86315	0.83028	0.00048	0.01752
Dataset 3				
Node PDR predictions	0.85142	0.80767	0.00891	0.07099
Network PDR predictions	0.87072	0.83193	0.00039	0.01625
Dataset 4				
Node PDR predictions	0.80965	0.75679	0.01222	0.0831
Network PDR predictions	0.72471	0.71837	0.00068	0.02062

- the node was isolated, but the increase in the transmission power left it isolated;
- the node was isolated, and the increase in the transmission power connected it to the nodes of the network;
- the node was not isolated, but the increase in the transmission power changed its PDR value within the MAE error;
- the node was not isolated, and the increase in the transmission power significantly increased its PDR value.

Table 6.3 showed ratio of each case in each of the datasets. As expected, increasing the transmission power of isolated nodes in sparse networks does not increase their reliability, so such nodes are redundant. However, if the increase in the transmission power connects the isolated node to the network, the decision on its need is made by comparing its maximum achieved reliability with the reliability of other network nodes. In addition, the increase in transmission power of non-isolated nodes has a more significant impact on the reliability of nodes in dense networks of smaller sizes because, in large networks, the increase in the transmission range of the node is less significant compared to the size of the entire network.

Thus, the proposed model can predict the change in the PDR value of the node when changing its transmission power, which can be used to determine the optimal transmission power.

TABLE 6.3: Node PDR predictions while optimization

	Zero PDR Unchanged	**Zero PDR Improved**	**Non-zero PDR Unchanged**	**Non-zero PDR Improved**
Dataset 1	0	0	0.69	0.31
Dataset 2	0	0	0.41	0.59
Dataset 3	0.067	0.144	0.422	0.367
Dataset 4	0.05	0.233	0.617	0.1

6.4 Implementation details

To implement our neural models, we used the framework of deep learning – *PyTorch* [4] with the integration of python 3.7. In addition, to represent the data in the form of graphs, we used a library *NetworkX* [5] that is compatible with the *Deep Graph Library* (DGL) [6] that was used to model graph neural networks.

All networks were trained on a GeForce GTX 1080 TI GPU.

[4]https://pytorch.org/
[5]https://networkx.org/
[6]https://docs.dgl.ai/index.html

Chapter 7

Conclusion

7.1 Contribution

In this work, we considered the task of reliability prediction of BLE Mesh networks in a supervised fashion and made the following contributions:

1. We made a comprehensive description of the BLE Mesh networks and provided an overview of related works dedicated to evaluating their performance.
2. To the best of our knowledge, our work is the first attempt to apply neural network models in the BLE Mesh domain, especially in reliability prediction task.
3. We collected four synthetic datasets that differ by sizes and analyzed how different network parameters affect its reliability. Based on this, we proposed some data preprocessing techniques to utilize local topology information at the node level.

Overall, we developed two neural network approaches for the node regression task. The first one is a multilayer perceptron that predicts the reliability of a particular node only based on its feature vector. The second one – graph convolutional network that uses whole network while training and maintains the natural connection between the nodes. Also, both models are suitable for solving the problem, MLP demonstrates better results on all datasets. We assume this because of the relative simplicity of the considered networks, which does not allow GCNs to demonstrate their full potential.

We also proved the feasibility of using the proposed models to estimate network reliability and find the optimal transmission power of the node to achieve its maximum reliability.

7.2 Future work

We have several directions for future work:

1. Since we use the simplest model of wireless communication in this work, it is worth noting to improve the simulator, given the various noises and interference that are always present in the natural environment.
2. In order to satisfy the connectedness of generated BLE Mesh networks, they are dense enough and have a relatively small average shortest path. However, in a natural environment, such networks are more sparse and have fewer receivers in their range. Therefore, it is advisable to consider sparse networks

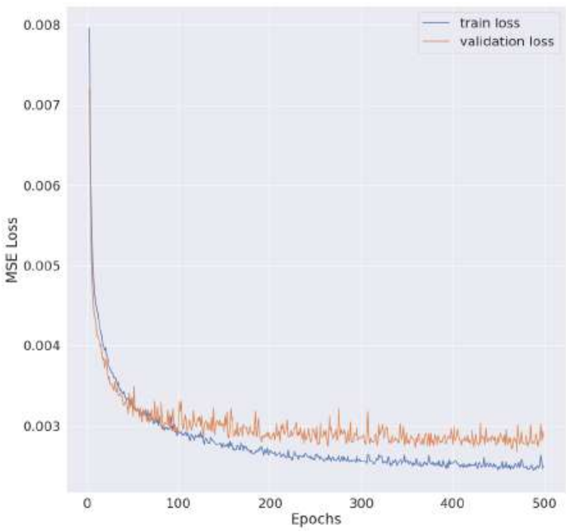
and propose an algorithm for constructing such networks. In addition, we should be considered heterogeneous networks.

3. It also makes sense to look for other data preprocessing techniques that can extract and create some more valuable features based on wireless communications' physical properties.

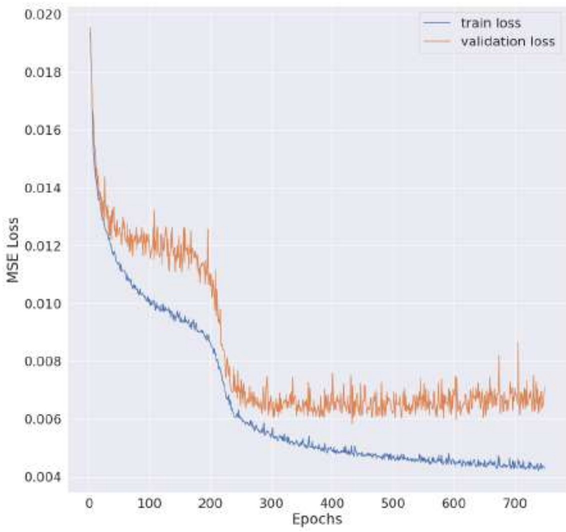
4. Taking the results of this thesis as a baseline, we can consider more complex neural networks, particularly other graph neural networks, which have proven their applicability in other areas of graph data.

Appendix A

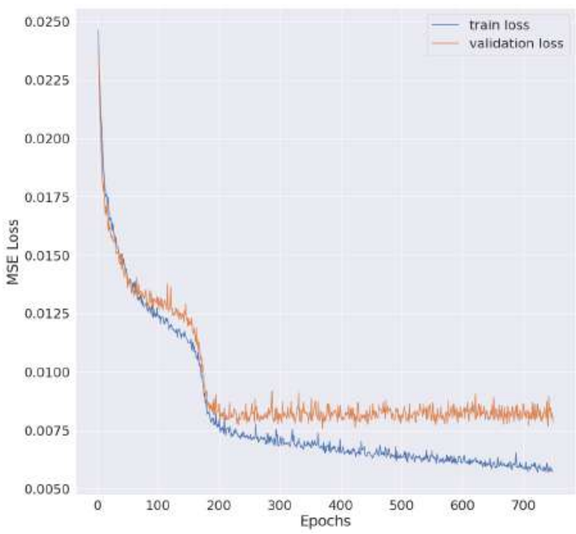
GCN training



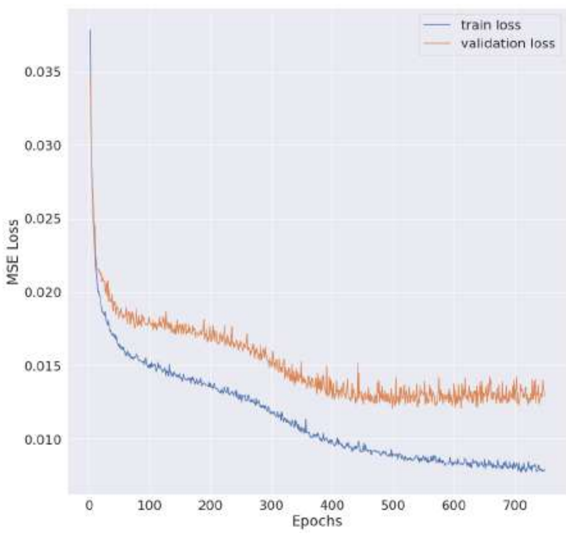
(A) Dataset 1



(B) Dataset 2



(C) Dataset 3



(D) Dataset 4

FIGURE A.1: GCN models losses (trained on different datasets)

Bibliography

Ahlin, Lars, Jens Zander, and Slimane Ben Slimane (2006). *Principles of Wireless Communications*. URL: http://kth.diva-portal.org/smash/record.jsf?pid=diva2%3A460175&dswid=1935.

Ateeq, Muhammad et al. (2020). "Deep Learning-Based Multiparametric Predictions for IoT". In: *Sustainability* 12.18, p. 7752. URL: https://www.researchgate.net/publication/345245078_Deep_Learning-Based_Multiparametric_Predictions_for_IoT.

Bacciu, Davide et al. (2020). "A Gentle Introduction to Deep Learning for Graphs". In: *Neural Networks* 129, pp. 203–221. URL: https://arxiv.org/abs/1912.12693.

Baert, Mathias et al. (2018). "The Bluetooth Mesh Standard: An Overview and Experimental Evaluation." In: *Sensors* 18.8, p. 2409.

Barabási, Albert-László and Márton Pósfai (2016). *Network science*. Cambridge: Cambridge University Press. ISBN: 9781107076266 1107076269. URL: http://barabasi.com/networksciencebook/.

Bishop, Christopher M. (1995). *Neural networks for pattern recognition*.

Bluetooth SIG (2010). "Bluetooth Core Specification 4.0". In: URL: https://www.bluetooth.com/specifications/specs/?status=all&keyword=&filter=.

— (2017). "Mesh Profile Bluetooth Specification v1.0.1". In: URL: https://www.bluetooth.com/specifications/specs/.

Bęben, Andrzej, Andrzej Bąk, and Maciej Sosnowski (2020). "Efficient relay node management in BLE MESH networks". In: *International Journal of Electronics and Telecommunications* 66.1, pp. 29–35.

Chakkor, Saad et al. (2014). "Comparative Performance Analysis of Wireless Communication Protocols for Intelligent Sensors and Their Applications". In: *International Journal of Advanced Computer Science and Applications* 5.4. URL: https://arxiv.org/abs/1409.6884.

Chowdhury, Arindam et al. (2020). "Unfolding WMMSE using Graph Neural Networks for Efficient Power Allocation". In: *arXiv preprint arXiv:2009.10812*. URL: https://arxiv.org/abs/2009.10812.

Cilfone, Antonio et al. (2019). "Wireless Mesh Networking: An IoT-Oriented Perspective Survey on Relevant Technologies". In: *Future Internet* 11.4, p. 99. URL: https://www.mdpi.com/1999-5903/11/4/99.

Darroudi, Seyed Mahdi and Carles Gomez (2017). "Bluetooth Low Energy Mesh Networks: A Survey." In: *Sensors* 17.7, p. 1467. URL: https://pubmed.ncbi.nlm.nih.gov/28640183/.

Decibel Tutorial: dB and dBm vs. Gain and Milliwatts. https://www.rfcafe.com/references/electrical/decibel-tutorial.htm.

Deif, Dina S. and Yasser Gadallah (2017). "A comprehensive wireless sensor network reliability metric for critical Internet of Things applications". In: *Eurasip Journal on Wireless Communications and Networking* 2017.1, pp. 1–18. URL: https://link.springer.com/article/10.1186/s13638-017-0930-3.

Deng, Li and Dong Yu (2014). *Deep Learning: Methods and Applications*. URL: https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/DeepLearning-NowPublishing-Vol7-SIG-039.pdf.

Farej, Ziyad Khalaf and Ali Maher Abdul-Hameed (2015). "Performance Comparison among (Star, Tree and Mesh) Topologies for Large Scale WSN based IEEE 802.15.4 Standard". In: *International Journal of Computer Applications* 124.6, pp. 41–44. URL: https://ui.adsabs.harvard.edu/abs/2015IJCA..124f..41K/abstract.

Glorot, Xavier and Yoshua Bengio (2010). "Understanding the difficulty of training deep feedforward neural networks". In: *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pp. 249–256. URL: http://proceedings.mlr.press/v9/glorot10a/glorot10a.pdf.

Goodfellow, Ian, Yoshua Bengio, and Aaron Courville (2016). *Deep Learning*. http://www.deeplearningbook.org. MIT Press.

Hansen, Emil A. J. et al. (2018). "On Relay Selection Approaches in Bluetooth Mesh Networks". In: *2018 10th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, p. 8631214.

Hastie, Trevor, Robert Tibshirani, and Jerome H. Friedman (2001). *The Elements of Statistical Learning*. Vol. 1. 10. Springer series in statistics New York, NY, USA.

Hernandez-Solana, angela et al. (2020). "Bluetooth Mesh Analysis, Issues, and Challenges". In: *IEEE Access* 8, pp. 53784–53800. URL: https://www.researchgate.net/publication/339923791_Bluetooth_Mesh_Analysis_Issues_and_Challenges.

Hinton, G. E. and R. R. Salakhutdinov (2006). "Reducing the Dimensionality of Data with Neural Networks". In: *Science* 313.5786, pp. 504–507. URL: http://www.cs.utoronto.ca/~rsalakhu/papers/science_som.pdf.

Iyer, A., C. Rosenberg, and A. Karnik (2009). "What is the right model for wireless channel interference". In: *IEEE Transactions on Wireless Communications* 8.5, pp. 2662–2671. URL: https://ieeexplore.ieee.org/document/4927481.

Kain, Nitin Kumar (2018). "Understanding of Multilayer perceptron (MLP)". In: URL: https://medium.com/@AI_with_Kain/understanding-of-multilayer-perceptron-mlp-8f179c4a135f.

Kim, Junseok and Younggoo Kwon (2008). "Interference-Aware Transmission Power Control for Wireless Sensor Networks". In: *IEICE Transactions on Communications* 91.11, pp. 3434–3441.

Kingma, Diederik P. and Jimmy Ba (2014). "Adam: A Method for Stochastic Optimization". In: *arXiv preprint arXiv:1412.6980*. URL: https://arxiv.org/abs/1412.6980.

Kipf, Thomas N. and Max Welling (2016). "Semi-Supervised Classification with Graph Convolutional Networks". In: *arXiv preprint arXiv:1609.02907*. URL: https://arxiv.org/abs/1609.02907.

Kulin, Merima et al. (2017). "Poster: Towards a Cognitive MAC Layer: Predicting the MAC-level Performance in Dynamic WSN using Machine Learning". In: *Proceedings of the 2017 International Conference on Embedded Wireless Systems and Networks*, pp. 214–215.

Kuo, Way and Ming J. Zuo (2002). *Optimal Reliability Modeling: Principles and Applications*, pp. 85–102. URL: http://www.esfandi.ir/files/ebook/ORM.pdf.

LeCun, Yann, Koray Kavukcuoglu, and Clement Farabet (2010). "Convolutional networks and applications in vision". In: *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*, pp. 253–256. URL: http://cs.williams.edu/~andrea/cs374/Articles/lecun-iscas-10.pdf.

Leon, Eduardo De and Majid Nabi (2020). "An Experimental Performance Evaluation of Bluetooth Mesh Technology for Monitoring Applications". In: *2020 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1–6.

Liendo, Andreina et al. (2018). "BLE Parameter Optimization for IoT Applications". In: *2018 IEEE International Conference on Communications (ICC)*, pp. 1–7. URL: https://hal.archives-ouvertes.fr/hal-01775056/file/ble_parameter.pdf.

Liu, Tao and Alberto E. Cerpa (2014). "Data-driven link quality prediction using link features". In: *ACM Transactions on Sensor Networks* 10.2, p. 37.

Liu, Weibo et al. (2017). "A survey of deep neural network architectures and their applications". In: *Neurocomputing* 234.234, pp. 11–26. URL: https://bura.brunel.ac.uk/bitstream/2438/14221/1/FullText.pdf.

Ni, Sze-Yao et al. (1999). "The broadcast storm problem in a mobile ad hoc network". In: *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, pp. 151–162. URL: http://cs.brown.edu/courses/cs295-1/storm.pdf.

Pearson, Karl (1895). "Note on Regression and Inheritance in the Case of Two Parents". In: *Proceedings of The Royal Society of London* 58.-1, pp. 240–242.

Rappaport, Theodore S. (1996). *Wireless Communications: Principles and Practice*.

Rondon, Raul et al. (2020). "Understanding the Performance of Bluetooth Mesh: Reliability, Delay, and Scalability Analysis". In: *IEEE Internet of Things Journal* 7.3, pp. 2089–2101. URL: https://arxiv.org/abs/1910.03345.

Shi, Qingjiang et al. (2011). "An Iteratively Weighted MMSE Approach to Distributed Sum-Utility Maximization for a MIMO Interfering Broadcast Channel". In: *IEEE Transactions on Signal Processing* 59.9, pp. 4331–4340.

Silicon Labs (2013). "The Evolution of Wireless Sensor Networks". In: URL: https://www.silabs.com/documents/public/white-papers/evolution-of-wireless-sensor-networks.pdf.

Sun, Haoran et al. (2018). "Learning to Optimize: Training Deep Neural Networks for Interference Management". In: *IEEE Transactions on Signal Processing* 66.20, pp. 5438–5453. URL: https://arxiv.org/abs/1705.09412.

Vihlborg, Fredrik (2011). "On Models for Interference Calculations between Radio Communication Systems". In: URL: http://www.diva-portal.org/smash/record.jsf?pid=diva2%3A467170&dswid=252.

Wu, Zonghan et al. (2016). "A Comprehensive Survey on Graph Neural Networks". In: URL: https://arxiv.org/abs/1901.00596.

Zhang, Chaoyun, Paul Patras, and Hamed Haddadi (2018). "Deep Learning in Mobile and Wireless Networking: A Survey". In: *arXiv preprint arXiv:1803.04311*. URL: https://arxiv.org/abs/1803.04311.