A machine learning based approach for the link-to-system mapping problem

XINYI LIN
A machine learning based approach for the link-to-system mapping problem

XINYI LIN

Master in machine learning
Date: September 12, 2017
Supervisor: Josephine Sullivan
Examiner: Danica Kragic Jensfelt
School of Computer Science and Communication
Abstract

The quality of mobile communication is related to signal transmissions. Early detection of the errors in transmissions may reduce the time delay of communications. The traditional error detection methods are not accurate enough. Therefore, in this report, a machine learning based approach is proposed for the link-to-system mapping problem, which can predict the outcomes (received correctly or not) of the link-level simulations without knowing the exact signals that are being transmitted. In this method, the transmission state is assumed to be a function of the features of a channel environment like the interference and the noise, the relative motion between the transmitter and the receiver and this function is obtained using a machine learning method. The training dataset is generated by simulations of the channel environment. Logistic regression, support vector machine and neural networks are the three algorithms implemented in this thesis. Experimental results show that all three algorithms work well compared to traditional methods. Neural networks provide the best results for this problem. Furthermore, the neural network model is tested with a dataset consisting of features of ten different channel environments, which verified the generalization ability of the model.
# Contents

## Contents

1 Introduction ........................................ 1
   1.1 Related work .................................. 2
   1.2 Research Question ............................. 3
   1.3 Societal aspect ................................. 4
   1.4 Outline ....................................... 4

2 Background ........................................ 5
   2.1 The Evolution of Wireless Technology ........ 5
   2.2 LTE basic concepts ............................. 6
      2.2.1 Multipath Fading Channel ................ 6
      2.2.2 Doppler Spread ............................ 9
      2.2.3 MIMO ..................................... 10
      2.2.4 OFDM .................................... 13
      2.2.5 MCS ..................................... 15

3 Method ........................................... 17
   3.1 Problem Definition and Environment Configuration .............................. 17
   3.2 Data preparation ............................... 18
      3.2.1 PCA ..................................... 19
      3.2.2 SVD ..................................... 20
   3.3 Algorithm selection ............................ 20
      3.3.1 Logistic Regression ......................... 21
      3.3.2 Support Vector Machine .................... 22
      3.3.3 Neural Network ............................ 23
   3.4 Analytical Method .............................. 24

4 Experiments and Discussion ....................... 26
   4.1 Dataset ........................................ 26
   4.2 The Input ...................................... 27
   4.3 The classification methods ..................... 28
      4.3.1 Logistic Regression ......................... 29
      4.3.2 Support Vector Machine .................... 30
      4.3.3 Neural Network ............................ 32
      4.3.4 Summary .................................. 34
   4.4 Further exploration ............................ 34
4.5 Generalization test ................................................. 36

5 Conclusions ......................................................... 39
  5.1 Conclusion ....................................................... 39
  5.2 Further works .................................................. 39
Chapter 1

Introduction

The wireless communication system was first used at the end of twentieth century and then quickly developed in the following decades. The term wireless communication refers to the information transmission from one device to the other through the air, without utilizing any cable, wire or other directly physical mediums. In the present day, two users can contact each other through wireless communication by many devices, even when they are in remote places. Among the various types of wireless communication, mobile communication is developing significantly and has an indisputable impact in everyone’s life and work.

Mobile communication systems deploy many base stations as the intermediates for the communication service. Each base station has a limited service range, which is called a cell. Thousands of cells make up the cellular network. Figure 1.1 illustrates the simplest situation of two users communicating in one cell where the hexagon represents the cell. To contact user B, user A needs to first send a message to the base station of this cell. Then, the base station receives the data and re-sends it to user B. Similarly, the information, which user B wants to send, also needs to pass through the base station. If the two users are not in the same cell, then the communication needs to pass two base stations which correspond to the cells where the two users are located.

Figure 1.1: A simple mobile communication example. The information sent by user A needs to go through the base station and then is received by user B. This information is not always successfully received. The communication quality will improve if the transmission state is known in advance. Therefore, a link-level simulator is used to make predictions about whether data will be successfully received or not.
The signal transmitted from user A to user B is easy to be destroyed due to the noise of the communication environment. To maintain the communication quality, researchers invented several algorithms to improve the transmissions reliability. In the development of these algorithms, it is commonly agreed that a simulator is necessary to optimize the algorithms as well as testing the network performance. In general, there are multiple senders and receivers at various locations sending messages via different base stations, which often refers to the communication system. There is a massive computational cost to accurately simulate the whole system, especially with all details of the physical layer such as communication environments, the speed of the sender and the receiver, etc. Therefore, there is a need to separate the simulator in link-level (physical layer) and system-level (network related) for a better performance evaluation.

A link-level simulator contains all components related to one transmission, which includes: a transmitter, a channel representing the communication environment and a receiver. The link-level simulator runs simulations to assess the physical layer performance, which facilitates the study of channel estimation, channel tracking and channel prediction algorithms. It is not possible to reflect the network related issues like cell planning and scheduling in physical layer simulations that lead to the system-level simulator. The system-level simulator targets optimizing the system performance with multiple cells and better understanding the user behaviors in various network deployments.

Due to the complexity, a system-level simulator should rely on a simplified version of the link-level model, which is accurate enough to capture the statistical performance of the physical layer with less computations. The interaction between link and system level simulators is known as link-to-system(L2S) mapping. In wireless communication, there is more than ten years of research history in the L2S mapping problem and many useful methods are proposed to tackle this issue.

In this report, a machine learning based approach is proposed for the link-to-system mapping problem, which can predict the outcomes (received correctly or not) of the link-level simulations without knowing the exact signals that are being transmitted. In particular, the link-level simulator mimics the scenario that a user transmits the information and the receiver in the base station receives the information. Then, the model predicts if the receiver will successfully receive the information or not. The method is expected to be used as a part of the system-level simulator with faster prediction time as well as more accurate results compared to traditional methods.

1.1 Related work

The mapping process connecting link-level simulators and system-level simulators is vital and has already been considered carefully in wireless communication. In the 1990s, Average Value Interface and Actual Value Interface [1] have been used to interface the link level with the system level. However, both methods are not accurate or efficient enough for the cellar network. Recently, several link-to-system mapping methods have been proposed, which are precise and useful for high data rate system by using effective Signal to Interference plus Noise Ratio (SINR). Exponential Effective SINR Mapping (EESM) and Mutual Information based Effective SINR Mapping (MI-ESM) are the two L2S methods commonly implemented in current wireless communication systems.

Although many scientists in wireless communication analyze the L2S mapping problem, as far as we know, there is no literature describing how to use machine learning al-
algorithms to solve this topic. However, the use of machine learning techniques in commu-
nication systems has a long history with an extensive range of applications, which
comprise channel modeling and prediction, equalization, quantization, modulation, and
demodulation. The combination of machine learning and communications often refers
to the term cognitive radio which was first proposed by Joseph Mitola [2]. Cognitive ra-
dio offers the idea of intelligent communications that study and adapt to their environ-
ment. Previously, most cognitive radio studies relied on developing hard-coded rules to
guide the behaviors of radios in given scenarios. Clancy et al.[3] propose a learning en-
gine to integrate with a reasoning engine so that radios can remember the knowledge
studied from the past and make decisions accurately and quickly in the future. With the
remarkable progress in computer vision area achieved by machine learning algorithms,
researchers recently make the preliminary investigations on the communication system
with deep learning techniques. The report [4] proposes a deep learning based approach
for channel decoding problems with pieces of evidence proving that neural networks can
learn a form of decoding algorithm, rather than only a simple classifier.

1.2 Research Question

The behavior of a receiver in the base station is realistically simulated, studied and im-
proved by link-level simulators. In the real world, receivers apply the Cyclic Redun-
dancy Check (CRC) to identify the transmission states, which is either received correctly
or not. Hence, in the link-level simulator, the CRC value represents the transmitted state.
If all the received bits are the same as the original bits, then the CRC value equals to one,
otherwise it is zero. However, there is no real signal running into the system-level sim-
ulator so that it is not possible to calculate the actual CRC value. Therefore, a predicted
CRC value is used to model the transmission performance, which should have the same
statistical characteristics of the real communications that are simulated in the link-level
simulations.

In this report, a machine learning model is proposed as a new approach for the L2S
mapping problem. Link-level simulations generate the data with the real CRC value as
the training dataset. For mimicking the system-level simulation environments, the input
of the model does not contain the signal information. Therefore, all input features are the
parameters representing the communication configurations, for instance, the noise and
interference of the environment and the relative motion between the base station and the
user. There are two basic assumptions in this problem. The first one is that the commu-

nication system environments control the transmitted outcomes and the results are not
related to the sent signals. The other assumption is that the link-level simulation can ac-
curately mimic the receiver in the real world and all the data generated by the link-level
simulator is precise and without noise. In conclusion, it is a binary classification problem
where the target is the state of the transmission, which either is 0 or 1.

It is not easy to adequately capture the statistical properties of complicated receivers
(e.g. iterative receivers with non-linear equalizers and channel estimation imperfections)
even when using a combination of analytical models (for example, to derive the post-
SINR) and link-to-system level curves. Moreover, whenever the receiver algorithm changes
the modeling work needs to be redone. Therefore, this report presents a learning model
based on machine learning method to replace conventional approaches. The learning
model is expected to adapt to different environments without retraining. The focus of
this study is to investigate the performance of the learning model in the link-to-system mapping problem and develop a framework to utilize machine learning techniques in the physical layer. Specifically, the research topics in this project are listed below.

- Test if machine learning methods can predict the transmission states with a high accuracy compared to conventional approaches.
- Identify the best machine learning algorithm with a compromise between the speed and accuracy.

The goal of this report is to offer a framework to classify the transmission states in different communication environments. Two datasets are used to evaluate the performance of different models. One dataset focuses on one scenario which is used to test the various model structures. The other dataset mimics the complex communication environments to check the generalization ability of the final model. The primary model proposed in the thesis is an artificial neural network, whose final architecture is decided by performing experiments and evaluating the effect of choosing different parameter settings.

### 1.3 Societal aspect

This report targets at the CRC prediction problem, which might potentially contribute to the society in several aspects. Improving the prediction accuracy in system-level simulations would have an impact on the wireless communication system deployment. Engineers could optimize the communication system design with a system-level simulator that is close to reality. An optimized system could reduce the possibility of deploying additional base stations, which avoids the drain on workforce and material resources. Moreover, the network related algorithms could be improved with a good system-level simulator so that all the system-level problems, such as cell planning and cell tracking, could find better solutions.

In addition, the communication quality might benefit from correct CRC predictions because early detection of errors reduces time delay. Nowadays, long-distance communication is no longer a problem, but people might uncomfortable or annoyed if the time delay is more than one second. Correctly predicting the CRC value in base stations helps to reduce this delay so that people can communicate without any problem. Furthermore, reducing the time delay means increasing the transmitting data rate. A communication system with high data rate could support more applications, such as virtual reality games.

### 1.4 Outline

The thesis is organized into individual chapters. Chapter 2 introduces some basic concepts of LTE physical layer. Chapter 3 formulates the link-to-system mapping problem and presents some classical machine learning approaches as well as the theoretical method used in wireless communication area. Chapter 4 describes the datasets, analyzes the performance of different learning models and discusses the experimental outcomes. Finally, Chapter 5 summarizes the experiments and provides some ideas of further studying.
Chapter 2

Background

Before jumping into the detailed description of the main problem, it is worth taking a look at the background of wireless communication and some fundamental technologies involved.

2.1 The Evolution of Wireless Technology

The beginnings of digital wireless technology start with second generation project in the 1990s that is called 2G. The key standard of 2G is Global System for Mobile (GSM). In the meantime, there are some competitive technologies such as CDMA, GPRS and EDGE. Table 2.1 shows the details about 2G evolution. At that time, the wireless system with data rate around 10-100 Kbps only supported voice call and basic Internet access.

<table>
<thead>
<tr>
<th>Network Name</th>
<th>Technology</th>
<th>Data Rate(per user)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2G</td>
<td>GSM(Global System for mobile)</td>
<td>10Kbps</td>
</tr>
<tr>
<td>2G</td>
<td>CDMA(Code Division Multiple Access)</td>
<td>~10Kbps</td>
</tr>
<tr>
<td>2.5G</td>
<td>GPRS(General Packet Radio Service)</td>
<td>~50Kbps</td>
</tr>
<tr>
<td>2.75G</td>
<td>EDGE(Enhanced Data rates for GSM Evolution)</td>
<td>~200Kbps</td>
</tr>
</tbody>
</table>

After ten years development, the 2G wireless generation system evolved to be the third Generation system. In the 2000s, several countries deployed the 3rd generation wireless system with different standards. Table 2.2 specifies the leading technologies and their data rates. It is remarkable that the data speed significantly increased in the third generation to a maximum of around 30Mbps, and many more applications were supported such as high-speed video callings.

The life cycle of the 3G system was around ten years. The first fourth-generation standard had been commercially deployed in Oslo, Norway, and Stockholm, Sweden since 2009. There were mainly two technologies used in 4G that are Long Term Evolution (LTE) and Worldwide Interoperability for Microwave Access (WiMAX). In the 4G era, the high quality of data rate as shown in Table 2.3 could meet the various demands of multimedia services such as online games, video conferences and high-definition televisions.
### Table 2.2: The evolution of 3rd generation

<table>
<thead>
<tr>
<th>Network Name</th>
<th>Technology</th>
<th>Data Rate (per user)</th>
</tr>
</thead>
<tbody>
<tr>
<td>3G</td>
<td>WCDMA/UMTS (Wideband CDMA / Universal Mobile Telecommunication standard)</td>
<td>∼ 384Kbps</td>
</tr>
<tr>
<td>3G</td>
<td>CDMA 2000</td>
<td>∼ 384Kbps</td>
</tr>
<tr>
<td>3.5G</td>
<td>HSDPA/HSUPA (High Speed Downlink/Uplink Packet Access)</td>
<td>5 ∼ 30Mbps</td>
</tr>
<tr>
<td>3.5G</td>
<td>1xEV-DO Rev A, B, C (Evolution-Data Optimized)</td>
<td>5 ∼ 30Mbps</td>
</tr>
</tbody>
</table>

### Table 2.3: The evolution of 4th generation

<table>
<thead>
<tr>
<th>Network Name</th>
<th>Technology</th>
<th>Data Rate (per user)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4G</td>
<td>LTE (Long Term Evolution)</td>
<td>100 ∼ 200 Mbps</td>
</tr>
<tr>
<td>4G</td>
<td>WiMAX (Worldwide Interoperability for Microwave Access)</td>
<td>∼ 100Mbps</td>
</tr>
</tbody>
</table>

## 2.2 LTE basic concepts

In this section, multipath fading channel model, Doppler spread, MIMO system as well as OFDM technology are addressed for better understanding.

### 2.2.1 Multipath Fading Channel

To better understand the wireless propagation problem, scientists and engineers develop the channel model to approximate the reality. The medium between the transmitting antenna and the receiving antenna is referred as channel [5], as shown in Figure 2.1. The wireless signal changes when it travels from transmitter (Tx) side to the receiver (Rx) side. These changes are caused by the different distance between two antennas, the number of paths chosen by the signal and the various types of environment around the path. Hence, the channel model is used to describe the complicated relationship between transmitted signals and received signals. In Figure 2.1, input signal $x$, propagates through channel $H$ and becomes output signal $y$. The propagation process can also be explained by the equation below.

$$y(f) = H(f)x(f) + n(f)$$

(2.1)

In Equation 2.1, $n(f)$ is the noise. There are some fundamental and accepted assumptions about the noise. Firstly, the noise is additive that means the noise and the signal...
are statistically independent. In other words, a received signal equals to the sum of the original signal and the noise. Then, it is a white noise which refers to the uniform power spectral density. The last assumption is that the sample of the noise has a Gaussian distribution in the time domain. The AWGN (Additive White Gaussian Noise) channel model is a simple and tractable mathematical model where the only factor affecting the communication quality is additive white Gaussian noise.

![The downlink multipath scenario](image1)

The multiple paths are caused by reflections of objects such as vehicles or buildings. This figure is copied from Jain [6]

The typical wireless communication scenario consists of a base station and a user equipment (UE) such as a mobile. The data transmitting from a base to a UE is called downlink transmission. On the contrary, the uplink transmission is the situation that the signal propagates from a mobile to a base station. However, there is no guarantee that there is no disturbance between a transmitter and a receiver in reality. The objects around the path reflect the wireless signals so that the receiver could also receive some of the reflected waves. The term line-of-sight (LOS) path depicts the straight line between a Tx and a Rx. Some other paths that arise because of the reflection from for instance objects such as vehicles or buildings, are called non-LOS paths. Figure 2.2 illustrates the multipath scenario. Commonly, the non-LOS signals are interference that is either constructive or destructive. Constructive interference enhances the received signal amplitude while destructive interference attenuates the amplitude. However, constructive interference is very rare indeed.

![Multipath Power Delay Profile](image2)

Each reflected signal takes a different path, so it has different amplitude and phase. Therefore, a single impulse sent from the transmitter results in multiple copies being received by the receiver with the time shift due to the differences in the distance that each reflected wave travels. Figure 2.3 is the delay profile of 4 tap-model. A delay profile represents a "tapped delay line" in the time domain that consists of some points at fixed positions on the delay line. The term delay spread describes the time differences between the earliest arrival multipath component and the latest arrival multipath component. A larger delay spread indicates a highly dispersive channel.
The impulse response of a multipath channel can be modelled by the following formula:

\[ h(t) = \sum_{i=0}^{L-1} a_i(t) \delta(t - \tau_i) \]  

(2.2)

There are L components in the model above, which mean a delay line with L taps. The coefficient \( a_i \) is the attenuation of the \( i^{th} \) path and \( \tau_i \) is the time delay of the \( i^{th} \) multipath component. Note that the channel response \( h \) varies with time \( t \) and the variance depends on the channel attenuation and the time delay of different multipath components. Depicting the relationship between the power of channel response \( h \) with the time \( t \), it will be similar to Figure 2.4. The power of a received signal fluctuates due to the phase variation and the angle variation of reflected signals. This phenomenon is known as multipath fading.

Figure 2.4: The fading phenomenon. Multipath fading channel response fluctuates in the time domain because each reflected signal has different phase and angle.

The channel mentioned in this report refers to the multipath fading channel. 3GPP standard [7] specifies three multipath fading channel models: Extended Pedestrian A model (EPA), Extended Vehicular A model (EVA) and Extended Typical Urban model (ETU), which are representative of low, medium, and high delay spread environments, respectively. The model parameters of each channel model are defined in Table 2.4 and Table 2.5 describes the tapped delay line models of EPA, EVA and ETU.

<table>
<thead>
<tr>
<th>Model</th>
<th>Number of channel taps</th>
<th>Delay spread (r.m.s.)</th>
<th>Maximum excess tap delay (span)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extended Pedestrian A model (EPA)</td>
<td>7</td>
<td>45 ns</td>
<td>410 ns</td>
</tr>
<tr>
<td>Extended Vehicular A model (EVA)</td>
<td>9</td>
<td>357 ns</td>
<td>2510 ns</td>
</tr>
<tr>
<td>Extended Typical Urban model (ETU)</td>
<td>9</td>
<td>991 ns</td>
<td>5000 ns</td>
</tr>
</tbody>
</table>

The characteristics of a wireless channel changes when the transmitter, receiver or other objects in the environment move. The time interval that the channel characteristics are considered to be constant is called coherence time. Each phenomenon in the time domain has a corresponding phenomenon in the frequency domain. The Fourier transfer of the delay profile reflects the frequency dependency of channel characteristics. The term bandwidth is used to describe the range of frequency. Therefore, coherence bandwidth represents the frequency bandwidth that channel characteristics remain similar. Because the delay spread is the measurement of channel dispersion, the coherence bandwidth is inversely related to the delay spread. From table 2.4, it is evident that the delay spread of
Table 2.5: Tapped delay line models for EPA, EVA, ETU

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.0</td>
<td>0</td>
<td>0.0</td>
<td>0</td>
<td>-1.0</td>
</tr>
<tr>
<td>30</td>
<td>-1.0</td>
<td>30</td>
<td>-1.5</td>
<td>50</td>
<td>-1.0</td>
</tr>
<tr>
<td>70</td>
<td>-2.0</td>
<td>150</td>
<td>-1.4</td>
<td>120</td>
<td>-1.0</td>
</tr>
<tr>
<td>90</td>
<td>-3.0</td>
<td>310</td>
<td>-3.6</td>
<td>200</td>
<td>0.0</td>
</tr>
<tr>
<td>110</td>
<td>-8.0</td>
<td>370</td>
<td>-0.6</td>
<td>230</td>
<td>0.0</td>
</tr>
<tr>
<td>190</td>
<td>-17.2</td>
<td>710</td>
<td>-9.1</td>
<td>500</td>
<td>0.0</td>
</tr>
<tr>
<td>410</td>
<td>-20.8</td>
<td>1090</td>
<td>-7.0</td>
<td>1600</td>
<td>-3.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1730</td>
<td>-12.0</td>
<td>2300</td>
<td>-5.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2510</td>
<td>-16.9</td>
<td>5000</td>
<td>-7.0</td>
</tr>
</tbody>
</table>

EPA, EVA and ETU is increasing, then the coherence bandwidth of these three channel models should be correspondingly decreasing, as shown in Figure 2.5.

### 2.2.2 Doppler Spread

The delay profile is the statistical power distribution in the time domain for a signal transmitted at one time. Similarly, the statistical power distribution in the frequency domain for a signal transmitted at one frequency is called Doppler spectrum. The Doppler spectrum is caused by the change of the environments, for instance, the relative motion between a Tx and a Rx. It is quite common that a user equipment (UE) is moving when it transmits and receives the signal. The movement of the Tx leads to a change in the frequency of the received signal, which also called Doppler shift. Assuming Doppler shift is $f_d$ in the frequency domain, the frequency of a received signal becomes $f_c + f_d$, where $f_c$ is the carrier frequency. Since the user might move toward or away from the base, the frequency of the received signal is in the range of $[f_c - f_D, f_c + f_D]$. Here, $f_D$ is Doppler spread which equals the maximum of the Doppler shift $f_d$. Similar to the relationship between the coherence frequency with the delay spread, the coherence time is inversely related to the Doppler spread:

$$\text{Coherence Time} \approx \frac{1}{\text{Doppler Spread}}$$  \hspace{1cm} (2.3)

In conclusion, if a transmitter or a receiver moves very fast, the Doppler spread is large while the coherence time is small, which means the channel changes fast. 3GPP standard specification [7] shows the propagation conditions (Table 2.6) that are used for performance measurements in multipath fading environment for low, medium and high Doppler frequencies.

Table 2.6: Channel model parameters

<table>
<thead>
<tr>
<th>Model</th>
<th>Maximum Doppler frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>EPA 5Hz</td>
<td>5 Hz</td>
</tr>
<tr>
<td>EVA 5Hz</td>
<td>5 Hz</td>
</tr>
<tr>
<td>EVA 70Hz</td>
<td>70 Hz</td>
</tr>
<tr>
<td>ETU 70Hz</td>
<td>70 Hz</td>
</tr>
<tr>
<td>ETU 300Hz</td>
<td>300 Hz</td>
</tr>
</tbody>
</table>
2.2.3 MIMO

MIMO, Multiple Input Multiple Output, is the key technology used in the 3G/4G wireless system. It provides greater spectral efficiency, better reliability, and larger data rate compared with the traditional Single Input Single Output (SISO) system. Many theoretical studies [8, 9] and communication system design experiments [10] prove the great performance improvement by implementing MIMO scheme.

Including MIMO scheme, the communication system broadly divides into four classifications concerning the number of transmitting antennas and receiving antennas as...
listed below. Typically, the antennae configuration is noted as $N_T \times N_R$, where $N_T$ is the number of transmitting antennas and $N_R$ is the number of receiving antennas.

- SISO, Single Input and Single Output, with $1 \times 1$ antenna configuration
- SIMO, Single Input and Multiple Output, with $1 \times N_R$ antenna configuration ($N_R > 1$)
- MISO, Multiple Input and Single Output, with $N_T \times 1$ antenna configuration ($N_T > 1$)
- MIMO, Multiple Input and Multiple Output, with $N_T \times N_R$ antenna configuration ($N_T, N_R > 1$)

Figure 2.6 is a simple example of MIMO system. There are $N_T$ Tx antennas and $N_R$ Rx antennas. Each transmitting and receiving antenna pair has an independent fading channel. In other words, the MIMO system is a collection of fading channels consisting of each (Tx, Rx) antenna pair.

![Figure 2.6: An illustration of MIMO scheme. MIMO scheme consists of multiple transmitting antennas as well as multiple receiving antennas. Each transmitting antenna and receiving antenna pair is one fading channel.](image)

In addition to the antenna configurations, there are two remarkable techniques for improving the wireless communication system performance, spatial diversity and spatial multiplexing. Employing the transmitter and receiver diversity can eliminate fading effect [10] thereby increasing the reliability of transmissions. Spatial multiplexing techniques [11] increases the data rate in communication links so that it is more suitable for the applications that required high speed.

In diversity techniques, the same data is sent across independent channels to cancel the fading phenomenon. When a transmitter sends multiple copies of the information, the amount of fading suffered by each one is different. In this way, it guarantees that at least one copy suffers least fading compared with others. Therefore, the probability of recovering the correct signal increases thus enhancing the reliability of the system. The technique of sending duplicate signals is referred as spatial diversity or diversity in wireless communication system. The term diversity gain is the measurement metric for the amount of benefit obtained by employing diversity, which is equal to the maximum number of parallel paths $N_T \times N_R$. 

![Figure 2.6: An illustration of MIMO scheme. MIMO scheme consists of multiple transmitting antennas as well as multiple receiving antennas. Each transmitting antenna and receiving antenna pair is one fading channel.](image)
In SISO configuration listed below (Figure 2.7), the diversity gain equals to 0 because there is no parallel path in the system. Assume a data stream $[1 \ 0 \ 0 \ 1 \ 1]$ has been sent, the output of the system significantly depends on the channel quality. If the channel quality is poor, then the information might be lost or severely untrustworthy, which has significant side-effect to the communication system. The solution to overcoming the fast changing channel is to increase the transmitting and receiving antennas. In SIMO system as shown in Figure 2.8, there are two parallel paths for sending the data. Even of one of the paths fails, the chance of receiver to recover the information from the other one is high. Therefore, the more antennas the system deploys, the greater reliability the wireless communication system has.

![Figure 2.7: The diversity gain of a SISO scheme.](image)

A SISO scheme consists of one transmitting antenna with one receiving antenna, which gains zero diversity because no parallel channels is used.

![Figure 2.8: The diversity gain of a SIMO scheme.](image)

A SIMO scheme consists of one transmitting antenna with two receiving antenna, which achieves two diversity gains because the system has two parallel channels.

Compared with spatial diversity, spatial multiplexing transmits independent information over parallel channels with the same bandwidth and no additional power cost. In MIMO systems, the increasing in capacity linearly relates to the number of (Tx, Rx) antenna pair. In the ideal conditions, the throughput would be double if two independent streams are being transmitted at the same time [12]. Therefore, the data rate of the communication system is significantly boosted using a MIMO system with spatial multiplexing. In wireless communication, signals arriving from various directions provide degrees of freedom. The degree-of-freedom is the measurement of multiplexing gain, which equals to $\min(N_T, N_R)$.

Figure 2.9 is a toy example clarifying the difference between spatial diversity and spatial multiplexing. In the transmit diversity, same data is sent to different spatial dimensions as shown in Figure 2.9a. In the right, Figure 2.9b indicates that independent
CHAPTER 2. BACKGROUND

(a) A MIMO system with spatial diversity  
(b) A MIMO system with spatial multiplexing

Figure 2.9: The difference between spatial diversity and spatial multiplexing. The diversity technique sends same data in each antenna, while multiplexing technique sends different information in each antenna.

data is sent across the separate spatial channels. In conclusion, implementing MIMO using diversity techniques can provide diversity gain that is beneficial for improving the reliability, while applying MIMO using spatial multiplexing techniques provides multiplexing gain that aims to improve the data rate of the system.

2.2.4 OFDM

Orthogonal Frequency Division Multiplexing (OFDM) is a leading wireless broadband technology. The term broadband means a wireless system with broad bandwidth. OFDM operates over a wide range of frequency, therefore naturally the data rate is higher, which is suitable for the 3G/4G system. OFDM has been employed in 4G wireless systems like LTE. In the meantime, OFDM is practical because it has a low complexity of implementation.

Figure 2.10: A toy example of a broadband system. A broadband system has a large signal bandwidth and in the middle of bandwidth places a carrier.

The basic principal of OFDM can be explained using a toy example as shown in Figure 2.10. It is a typical broadband system with a bandwidth $B$ and a single carrier has placed in the center of bandwidth. The digital communication system transmits digital symbols and each symbol has a constant symbol time which determines the data rate of this network. The symbol time $T$ used to measure the performance of this broadband system is equal to $\frac{1}{B}$. Since it is a broadband system, the symbol time is tiny which is much smaller than the delay spread of the channel. As bandwidth $B$ increases, symbol time $\frac{1}{B}$ decreases leading to Inter-Symbol Interference (ISI).

Inter-Symbol Interference is a phenomenon where one symbol interferes with following symbols caused by multipath. ISI is unwanted because the effect of previous symbols is considered as noise to subsequent symbols that leads to a system that is less reliable.
The received signal is the sum of two multipath components with the different delay time. The different symbols of two paths are aligning which results in the deformation of the original signal. $S_0, S_1, S_2 \cdots$ in Figure 2.11 are the transmitted symbols and the symbol time is $T$. There are two paths in this communication system with the delay time $\tau_0$ and $\tau_1$ respectively. The received signal is the sum of these two copies of the original signal. Since the delay spread $\tau_1 - \tau_0$ is greater than the symbol time $T$, symbols $S_0$ in path 2 add with symbols $S_1$ in path 1 in the receiver. Different symbols are aligning in the receiver that causes the ISI. In other words, ISI occurs when the delay spread is larger than the symbol time. Hence, ideally, ISI can be eliminated if symbol time is longer than the delay spread.

Figure 2.12: The basic idea of OFDM. The OFDM technique divides the broad bandwidth into multiple narrow bands to avoid the ISI phenomenon.

ISI causes a degradation of the performance of the wireless communication system and therefore it is a significant challenge in broadband wireless systems. The practical method to overcome ISI is splitting the broadband into smaller bands which are called subbands. A subcarrier is placed in the center of each subband in Figure 2.12. If the bandwidth is $B$ and there are $N$ subbands, the bandwidth of each subband is $\frac{B}{N}$. By dividing the broadband, the bandwidth of each subcarrier is smaller and it is easy to make the new symbol time much greater than the delay spread by choosing a proper $N$. Therefore, there is no Inter-Symbol Interference in the system with multiple subbands and subcarriers. The idea of using a lot of parallel narrow band subcarriers instead of a single wide band carrier is the primary idea of OFDM. In LTE, the physical resource block (PRB) is the set of subcarriers, and this enables the system to be able to compartmentalize the data across different numbers of subcarriers. A PRB or resource block (RB), comprises 12 subcarriers regardless of the overall LTE signal bandwidth and each RB stands for
one subframe in the time domain. Hence, different signal bandwidths will have different numbers of RBs.

2.2.5 MCS

Modulation is the method of adjusting one or more properties of the carrier signal. The information coming out of a transmitter need to be modulated first then sent. On the receiving terminal, a receiver must do the reverse process, demodulation, to recover the original information. The properties of the signal modulation are frequency, amplitude and phase. All the modulation techniques alter at least one parameter of the sine wave to represent the transmitted data.

The LTE signal can choose between three types of modulation methods: QPSK (4QAM), 16QAM and 64QAM. Quadrature Amplitude Modulation (QAM) is a technique which is widely used for modulating data signals in radio communications. Quadratic Phase Shift Keying modulation (QPSK) or 4QAM uses a sinusoidal signal and changes its phase to transmit the information. In QPSK, two bits can be encoded using four different stages with $90^\circ$ phase shift and a constellation diagram represents the QPSK modulation scheme as shown in Figure 2.13a. The 16QAM technique encodes 4 bits symbol and uses both phase and amplitude. For instance, the data 0111 in Figure 2.13b is $225^\circ$ phase with 25% of the amplitude. The magnitude of 0000 is 75%, and the phase is $135^\circ$. To sum up, changing the amplitude and the phase generates new symbols. As for 64QAM, it is straightforward to calculate that it is 6 bits per symbol and outlines 64 different signal levels. The constellation diagram of 64QAM is too complicated to show in this report.

![Constellation diagram of QPSK modulation](image1.png)

![Constellation diagram of 16QAM modulation](image2.png)

Figure 2.13: The constellation diagram of different modulation techniques. The constellation diagram is a representation of the signal modulation scheme and the symbol position shown in the figure encodes the phase and the amplitude used to modulate.

Modulation and coding schemes (MCS) are used to determine the throughput and the data rate of a wireless communication system. Each MCS is based on a combination of the channel parameters such as modulation techniques, transport block sizes, resource block numbers. In general, MCS is linearly related to the code rate. Figure 2.14 gives an example of the relation between the MCS and the code rate under 5 resource blocks. Using the same modulation method, the code rate grows when the MCS index increases.
Figure 2.14: The relation between the MCS and code rate. In each modulation method, the code rate grows as the MCS index increases.
Chapter 3

Method

This chapter first gives a formal definition of the problem. Section two is about the data preparation, which is vital in the machine learning modeling process. Some classic machine learning algorithms are presented in Section three. The last part of this chapter introduces the traditional method used in wireless communication.

3.1 Problem Definition and Environment Configuration

In the previous Chapter 1.2, a preliminary version of the problem is described without going deep into the particulars. The following presents a formal problem definition and specifies wireless communication configurations.

Tom Mitchell [13] provides a short formalism of machine learning which has been widely used since. "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E." According to this formalism, the definition of the problem is as follows:

- **Task(T):**
  Classifying whether or not a transmission that has not been received will be correctly decoded.

- **Experience(E):**
  A corpus of transmissions for a specified receiver given in different environments where some transmissions fail and some do not.

- **Performance(P):**
  **Classification accuracy**, the number of transmissions predicted correctly out of all transmissions, considered as a percentage.

The performance of a classifier is measured based on the confusion matrix, which is built after each prediction made and compared with the real outcome. Table 3.1 shows all the possible situations. True Negatives($C_{00}$), False Positives($C_{01}$), False Negatives($C_{10}$) and True Positives($C_{11}$) denote the count of each possible state, respectively.

Equation 3.1 illustrates the mathematical computation of the performance metric.

$$Accuracy = \frac{C_{00} + C_{11}}{C_{00} + C_{01} + C_{10} + C_{11}}$$  \hspace{1cm} (3.1)
Table 3.1: Confusion Matrix

<table>
<thead>
<tr>
<th>real</th>
<th>predict</th>
<th>$CRC_p = 0$</th>
<th>$CRC_p = 1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$CRC_r = 0$</td>
<td>$C_{00}$</td>
<td>$C_{01}$</td>
<td></td>
</tr>
<tr>
<td>$CRC_r = 1$</td>
<td>$C_{10}$</td>
<td>$C_{11}$</td>
<td></td>
</tr>
</tbody>
</table>

According to the definition, Figure 3.1 shows a draft of the machine learning model. The inputs are the features of channel environments and the output is the CRC value which is either 0 or 1.

Figure 3.1: The draft of the machine learning model. The model contains the possible input features, a blackbox model and the output.

The error detection problem targets at the uplink transmission in LTE physical layer. As introduced in Chapter 2.2.4, LTE uses OFDM techniques to increase the data rate. In OFDM, six resource blocks (RBs) are chosen to simplify the problem. In total, the system has 72 subcarriers and 14 OFDM symbols of each subcarrier. In the mathematical view, the channel $H$ is a matrix with size $72 \times 14$ and each entry is a complex number. The uplink communication system has the SIMO schema with two receiving antennas. Therefore, there are two parallel paths from the Tx to the Rx that corresponds to two channels. To sum up, the channel $H$ of the system is a three-dimensional matrix of the size $72 \times 14 \times 2$, where the first dimension is the number of subcarriers, the second dimension corresponds to the number of OFDM symbols and the third dimension represents the number of receiving antennas. The features of the channel environments are summarized below.

- SIMO configuration: $1 \times 2$
- RB number: 6, corresponding to 72 OFDM subcarriers.
- Channel $H$: $72(\text{subcarriers}) \times 14(\text{symbols}) \times 2(\text{antenna})$ complex number matrix
- MCS value: 0, 5, 10, 15, 20
- Doppler spread: 5, 70

3.2 Data preparation

There is always a strong passion to include all the data that is available, so that the "more is better" statement will hold. However, this might not be suitable for the case with limited time and computational power. Since all the data is coming from link-level simula-
tions, which are easy to access, all features generated in the simulation process are relevant and available. Hence, it is troublesome to find a way to eliminate unnecessary features.

As mentioned before in Chapter 3.1, the channel $H$ is a $72 \times 14 \times 2$ complex number matrix that is equal to a 4032-dimensional vector, if the real part and imaginary part of the complex number are counted separately. Therefore, the input of the model is a vector with more than 4000 dimensions, which is hard to calculate and easy to cause the curse of dimensionality problem. The curse of dimensionality refers to the phenomenon that problems occur when the data space is in high dimensionality, while these problems do not appear in low dimensional data space. The data becomes sparse when the dimension expands. The growth of data needs to be exponential to catch up with the increase of dimensionality, which is hard to implement. In the meantime, sparsity destroys the common data organization so that grouping data with the same properties is almost impossible. In conclusion, there is a need to reduce high dimensionality while still preserving most information.

Dimension reduction consists of two parts: feature selection and feature extraction. Feature selection picks a subset of original features that is based on the prior knowledge from wireless communications. On the other hand, feature extraction transforms the high dimensional data into another space with lower dimensions and the transformation might be linear or non-linear. The most popular linear feature extraction algorithms are Principal Components Analysis (PCA) and Singular Value Decomposition (SVD).

### 3.2.1 PCA

Principal Components Analysis finds a linear combination of high dimensional data into a lower dimensional space. The goal of PCA is to maximize the retained variance while minimizing the least square reconstruction error. Using eigenvectors and eigenvalues of the covariance matrix, PCA finds the principal components which have highest eigenvalues. Those principal components are the new basis of the data space. Algorithm 1 is the pseudo-code of the PCA implementation.

**Algorithm 1: Principal Components Analysis**

<table>
<thead>
<tr>
<th>Input</th>
<th>Original $N \times d$ matrix $X$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output</td>
<td>New $N \times k$ matrix $Y$</td>
</tr>
</tbody>
</table>

1. Centralized the data (subtract the mean for each dimension)
2. Calculate the $d \times d$ covariance matrix: $C = \frac{1}{N-1}X^TX$
3. Calculate the eigenvectors and eigenvalues of the covariance matrix.
4. Sort the eigenvalues by descending order.
5. Select $k$ eigenvectors that correspond to the largest $k$ eigenvalues to form the $d \times k$ projecting matrix $P$.
6. Project the original data to the new space: $Y = XP$
3.2.2 SVD

Singular Value Decomposition is a matrix factorization approach commonly used in recommendation systems. In mathematics, any $N \times d$ matrix $X$ can be expressed as:

$$X = U \times \Sigma \times V^T$$  \hspace{1cm} (3.2)

Figure 3.2: An illustration of SVD. Singular Value Decomposition factorizes the matrix into three pieces: a left singular matrix, a diagonal matrix and a right singular matrix.

Figure 3.2 illustrates the dimension of each matrix and $r$ is the rank of the matrix $X$. $U$ is the left singular matrix with size $N \times d$ and $V$ is the right singular matrix with size $d \times r$. Note that $\Sigma$ is a diagonal $r \times r$ matrix where the singular values $\sigma_i$ are sorted in descending order. The pseudo-code of SVD is presented in Algorithm 2.

**Algorithm 2: Singular Value Decomposition**

- **Input**: Original $N \times d$ matrix $X$
- **Output**: New $N \times k$ matrix $Y$

1. Centralized the data (subtract the mean for each dimension)
2. Decompose the matrix: $X = U\Sigma V^T$
3. Project the original data to the new space: $Y = XV$

PCA and SVD look similar when comparing Algorithm 1 and Algorithm 2. It is evident that both algorithms are intimately related. Equation 3.3 proves that the eigenvectors of the covariance matrix $C$ are the same as the right singular matrix $V$ of $X$.

$$X^TX = V\Sigma U^T U\Sigma V^T = V\Sigma^2 V^T$$

$$\Rightarrow C = \frac{1}{N-1}X^TX = V\frac{\Sigma^2}{N-1}V^T = V\Lambda V^T$$  \hspace{1cm} (3.3)

In fact, using the SVD to perform PCA makes much better sense numerically than forming the covariance matrix to start with. This is because of the accuracy that might be lost when calculating $X^TX$. Therefore, PCA mentioned in the following chapters is, in fact, using SVD in implementation.

3.3 Algorithm selection

Since the report focuses on supervised learning, the selected algorithms are inspired by the popular supervised classification methods that include Logistic Regression (LR), Support Vector Machine (SVM) and Neural Network (NN). The experiments on these algorithms evaluate the performance of different ML algorithms for the CRC prediction task. The best algorithm can be chosen by comparing the performance.
3.3.1 Logistic Regression

Logistic regression is a bunch of regression analyses that is used as a classification method in machine learning today. The model is often used in studies where a dichotomous outcome variable is related to a number of independent variables [14]. In this report, the form of the logistic regression model is as follows: let \( Y \in \{0, 1\} \) denote a binary outcome variable that equals to 1 if the transmission is successful and 0 if failed. Let \( x = \{x_1, x_2, \ldots, x_N\} \) be the vector of \( N \) known factors. The logistic regression model tries to find a hypothesis \( h_\theta(x) \) that has the property \( 0 \leq h_\theta(x) \leq 1 \) since all the classification labels are either 0 or 1. The LR model then hypothesizes,

\[
h_\theta(x) = g(\theta^T x)
\]

where \( \theta \) is the set of weights corresponding to \( N \) features and the bias.

\[
g(z) = \frac{1}{1 + e^{-z}}
\]

is a sigmoid function as shown in Figure 3.3. The purpose of using the sigmoid function here is to limit the output to the range of 0 to 1 and to introduce the non-linearity to the model.

![Figure 3.3: Sigmoid function](image)

To put everything together, the hypothesis of logistic regression is as follows,

\[
h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}}
\]

In order to interpret the model, the hypothesis \( h_\theta(x) \) is assumed to be the probability that \( Y = 1 \) given the input \( x \) and weights \( \theta \), as shown in Equation 3.7.

\[
h_\theta(x) = P(Y = 1|x, \theta) = \frac{1}{1 + e^{-\theta^T x}}
\]

Because \( Y \) is a binomial variable, it is simple to deduce the probability of \( Y = 0 \).

\[
P(Y = 0|x, \theta) = 1 - P(Y = 1|x, \theta) = \frac{e^{-\theta^T x}}{1 + e^{-\theta^T x}}
\]

In order to automatically calculate the weights \( \theta \), the cost function \( J(\theta) \) is needed for optimization, which is described in Equation 3.9. Note that \((x^i, y^i)\) is one pair of training samples and \( m \) is the total number of training samples.
\[
\min_{\theta} J(\theta) = \min_{\theta} \left(-\frac{1}{m} \sum_{i=1}^{m} y_i \log h_\theta(x_i) + (1 - y_i) \log (1 - h_\theta(x_i)) \right)
\] (3.9)

Gradient descent is the most popular approach to minimize the cost function \( J(\theta) \) iteratively and Equation 3.10 is the update equation. \( \alpha \) here is the learning rate.

\[
\theta_j = \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta) = \theta_j - \alpha \sum_{i=1}^{m} (y_i - h_\theta(x_i)) x^i_j
\] (3.10)

After fixing the \( \theta \), the new label of an unseen sample can be estimated by using the formula 3.7. Finally, a threshold \( s \) is given to determine the new label. The output is 1 if \( h_\theta(x) \geq s \) otherwise the output is 0.

### 3.3.2 Support Vector Machine

Support Vector Machine is a supervised learning technique often used for binary classification and regression analysis. It is also known as the large margin classifier because SVM aims at finding two hyperplanes that represent the biggest separation between the two classes and this separation refers to the margin.

Logistic regression hypothesizes \( h_{w,b}(x) = g(wx + b) \) is the probability of \( Y = 1 \). Instead of using the sigmoid function, SVM model simplifies the hypothesis and directly maps it to 1 and -1. For optimization reasons, the labels used in SVM are \( Y \in \{-1, 1\} \).

The mapping function is shown in Equation 3.11.

\[
h_{w,b}(x) = \begin{cases} 
1, & wx + b \geq 1 \\
-1, & wx + b \leq -1 
\end{cases}
\] (3.11)

Figure 3.4: A toy example of SVM. The dataset contains two linearly separable classes. The black points in the figure belong to class \( Y = -1 \) while the white points are the class \( Y = 1 \). The dotted lines represent the best separation of the dataset because the two lines have the largest distance.

Figure 3.4 describes a two-dimensional data space with two types of data. Since the data is linearly separable, two parallel hyperplanes are chosen to split the data and the
distance between them is required to be as large as possible. The points in these hyper-
planes are called support vectors and the margin is the region bounded by these two hy-
perplanes, which can be expressed mathematically by:

\[
\begin{align*}
wx + b &= 1 \\
wx + b &= -1
\end{align*}
\]  

(3.12)

The distance between these two hyperplanes is \(\frac{2}{||w||}\). Intuitively, maximizing this range
would result in a good separation of the data. In other words, a desirable separation hy-
perplane \(wx + b = 0\) need to have the largest distance to the nearest data point from
any class. Maximizing the distance equals to minimizing \(||w||\) and some constraints are
needed to guarantee that there is no data inside the margin.

\[
\begin{align*}
wx_i + b &\geq 1, \quad \text{if } y_i = 1 \\
wx_i + b &\leq -1, \quad \text{if } y_i = -1 
\end{align*}
\]  

\(\implies y_i(wx_i + b) \geq 1
\)

(3.13)

Equation 3.14 summaries the cost function of the linear SVM.

\[
\begin{align*}
\min \quad ||w|| \\
\text{subject to } \quad y_i(wx_i + b) \geq 1, \text{ for all } i
\end{align*}
\]  

(3.14)

3.3.3 Neural Network

The neural network is a state-of-the-art technique used today for machine learning prob-
lems. In fact, it is an old algorithm that is originally motivated by the goal of mimicking
the brain. The neural network is developed as simulating the network of neurons in the
brain, which means the artificial neural network also consists of a bunch of neurons and
and the neural model is represented by the logistic unit depicted in Figure 3.5. A neuron re-
ceives several inputs \(x_1, x_2, x_3\) and it outputs the value \(h_\theta(x)\) computed like logistic re-
gression. Normally, there is a bias unit \(x_0\) in the neurons, which is always equal to 1 and
is sometimes ignored in the figure. If a network consists of logistic units, the sigmoid
function \(g(z)\) used in the calculation is known as the activation function that introduces
non-linearity to the model.

A neural network is a group of different neurons strung together. Figure 3.6 is a con-
crete example of an artificial neural network. It is a three-layer neural network: the first
layer is the input layer, the hidden layer is in the middle and the last layer is the out-
put layer. Terminologically, all layers except the input layer and the output layer are
called hidden layers. The yellow circles shown in the figure represent the neurons and
each neuron is computed just like logistic regression. Mathematically, the definition of
a neuron’s function \(h(x)\) is a composition of other functions \(a_i(x)\), which can be further
calculated as a form of some other functions. A widely used composition is non-linear
weighted sum:

\[
h(x) = g(\sum_i w_i a_i(x))
\]  

(3.15)

where \(g(z)\) is a predefined activation function, \(a_i\) is the value of \(i^{th}\) input neuron and
\(w_i\) is the weight of corresponding neuron. Compared with logistic regression, the neu-
ral network can learn its features such as \(a_1, a_2, a_3\) and these features can be interesting
Figure 3.5: **The neuron model.** A neuron is a logistic unit that first calculates the weighted sum of all inputs and then applies a non-linear function to the sum. This function is often called the activation function in the neural network.

Figure 3.6: **A artificial neural network example.** This neural network model consists of the input layer, one hidden layer and the output layer. Except the input layer, each node in the other layers is a neuron model. The number of layers and the number of neurons per layer are the hyperparameters of the model.

and complex depending on what parameters have been chosen. Therefore, the model can present a better hypothesis.

In the state-of-the-art neural network structures, the activation function usually uses the rectified linear unit (ReLu). The rectifier is defined as the maximum of zero and the output value, which eliminates the negative values in the network. The main benefits of ReLu are avoiding the vanishing gradient and introducing sparsity on the network. Vanishing gradient is a phenomenon caused by training the deep neural networks with gradient-based learning methods. The weights in the first several layers are difficult to update because the gradient is decreasing during the back propagation and the update is almost close to zero when it reaches the beginning of the network. Compared with the sigmoid function, relu does not have the vanishing gradient problem. In the meantime, setting all negative outputs of the hidden units to zero allows the network to have a more sparse representation that accelerates the learning time.

### 3.4 Analytical Method

In wireless communications, the Effective SINR Mapping (ESM) approach [15, 16] is widely used to solve the link-to-system mapping problem, which uses post-equalization SINR value to predict the transmission state. Post-equalization SINR, also known as post-SINR, is the intermediate product of the receiving process. Similar to SNR, a small value of post-SINR indicates lots of noise in the channel that usually means failed transmissions. On the contrary, a large post-SINR value represents good channel quality and thereby the probability of receiving correct information is high.

Conventional ESM-based methods like Exponential Effective SINR Mapping (EESM) [17] and Mutual Information based Effective SINR Mapping (MI-ESM)[18], calculate effective post-SINR and then maps it with a Block Error Rate (BLER) curve under the AWGN channel. The Block Error Rate curve depicts the relationship between the post-SINR and the block error probability, which is a ratio of the number of total erroneous blocks to the total number of blocks. In particular, BLER is a function of the Modulation and Coding Scheme (MCS), the number of allocated Resource Blocks (RBs), the post-SINR and the
channel states. Therefore, each MCS or each RB number corresponds to a single BLER curve.

Figure 3.7: BLER curves under the AWGN channel. ESM-based methods calculate the effective post-SINR of a transmission and then maps it with one BLER curve under the AWGN channel depending on the MCS index. The corresponding point in the BLER curve is the probability of receiving error information.

Figure 3.7 illustrates several BLER curves under the AWGN channel and each curve corresponds to one MCS index value. The MCS1 curve represents the scenario where the MCS index value equals to 1 and different MCS index values are located in various positions in the figure. ESM-based approaches first compute the effective post-SINR value and then maps it with the BLER curve given the MCS index. The corresponding point in the curve is the probability of receiving wrong information. It is apparent that a small post-SINR value implies a high error probability while a large post-SINR value indicates a low error probability. After knowing the error probability, a random number is generated from the uniform distribution in the range $[0, 1]$ and this number is compared with the error probability to estimate if this transmission is wrong or not.
Chapter 4

Experiments and Discussion

This chapter focuses on the experiments of the transmission prediction problem. The first section introduces how to choose a proper dataset. Section 2 describes all available features and lists several strategies for selecting the input. The next section analyzes three supervised learning algorithms and each algorithm is evaluated with the different feature choices. The algorithm with the best performance is further tested in section 4. Finally, the last part of this section estimates the generalization ability of the best model.

4.1 Dataset

The data is generated based on prior knowledge from the L2S mapping problem in wireless communications. ESM-based approaches calibrate the post-SINR so that it can be mapped with a BLER curve under the AWGN channel, where the post-SINR is a human-made feature occurred in link-level simulations that has similar properties as the SNR. Therefore, a proper set of parameters is chosen to generate the dataset so that the Block Error Rate (BLER) curve is similar to Figure 4.1a. A BLER curve is drawn by first sorting the samples in the increasing order of the post-SINR and then computing the error probability of samples with the same post-SINR value. The BLER curve of a valuable dataset contains a horizontal line BLER = 1 on the left, a steep declining part in the center and a straight line BLER = 0 on the right. The line BLER = 1 represents that all the samples are wrongly transmitted, while the BLER = 0 line illustrates the situation that all the transmissions are recovered successfully. In the middle, the curve undergoes a downward slide from 1 to 0 as shown in the area between two red lines of Figure 4.1a. This area is called descending section, which is approximately 5 dB depending on different scenarios. By putting the descending section in the middle, it guarantees that the distribution of two classes is even. Figure 4.1b shows the distribution of the post-SINR and the precision of the figure is 0.1 dB. The samples in the descending section are approximately 30% of the whole dataset.

The link-level simulator generates two datasets to test the performance of a model and Table 4.1 lists the details of the dataset. The EVA dataset is a toy example representing one specified scenario in wireless communication that is used in all experiments except the generalization test. As for the generalization test, the general dataset mimics the environment in the real world so that it can evaluate the generalization ability of the final model. The difference of two datasets is the MCS index and Doppler spread. In other words, the EVA dataset is a particular case of the general dataset and it contains more
(a) **The BLER curve of a suitable dataset.** The BLER curve of the dataset contains three sections: a horizontal line $x=1$ on the left, a steep decline in the middle and a horizontal line $x=0$ on the right. In this way, the number of successful and failing transmissions are even.

(b) **The distribution of the post-SINR.** The distribution is similar to a Gaussian. Most samples locate in the descending section and the number of samples decreases as the post-SINR away from the center.

![Figure 4.1: The dataset properties](image)

Table 4.1: Dataset Configuration

<table>
<thead>
<tr>
<th>Name</th>
<th>Size</th>
<th>Channel type</th>
<th>MCS</th>
<th>Doppler spread</th>
<th>RBs</th>
<th>Antennas</th>
</tr>
</thead>
<tbody>
<tr>
<td>EVA</td>
<td>60,000</td>
<td>EVA</td>
<td>5</td>
<td>5</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>General</td>
<td>120,000</td>
<td>EVA</td>
<td>0,5,10,15,20</td>
<td>5,70</td>
<td>6</td>
<td>2</td>
</tr>
</tbody>
</table>

Before jumping into the experiments, there are some basic settings need to be mentioned. Cross-validation is the evaluation approach for judging the performance of different models and the training and validation dataset split is 70% and 30%, respectively. To standardize the model performance, the performance metric uses two types of accuracy. One is the general prediction accuracy, which is computed based on the whole validation dataset. From Figure 4.1a, it is evident that the BLER curve is steep in the descending section, which makes this region vital in the prediction problem. Therefore, the descending accuracy is used to judge the outcomes in the dropping part, which is more precise to assess the model performance.

### 4.2 The Input

All the raw features are listed below:

- **post-SINR:** scalar variable. It is an intermediate product of the receiving process, which is a human-made feature that does not always exist in link-level simulations. The large post-SINR value shows the high probability of receiving the correct data.
• **Doppler spread**: scalar variable. The value of Doppler spread represents the relative motion between the transmitter and the receiver. The high Doppler spread indicates fast speed.

• **MCS Index**: scalar variable. The Modulation and Coding Scheme represents the combinations of modulation techniques and transportation block sizes. The MCS index is linearly related to the code rate in most cases.

• **Modulation order**: scalar variable. The different modulation orders mean the different modulation techniques.

• **TBS Index**: scalar variable. The TBS index represents the transport block size of a given resource block number. The large value of TBS index means the big transport block size.

• **The channel H**: a 4032-dimensional vector. The feature is obtained by converting the channel matrix to the vector.

The raw features include five scalar variables and a 4032-dimensional vector. For the different experiments, there are different strategies to select the inputs. If the EVA dataset is used, the input features only need the post-SINR and the channel H. Because the EVA dataset focuses on one scenario, Doppler spread, MCS Index, modulation order and TBS Index are the same value for all transmissions in this dataset. The following describes three different feature designs in the EVA dataset.

1. **Baseline Strategy**: All features, the post-SINR and the channel H, are directly used in training the model.

2. **Feature Selected Strategy**: Based on prior knowledge in wireless communication, the first OFDM symbol of the first antenna is extracted to represent the channel that is a 144-dimensional vector. Therefore, total inputs are 145 dimensions.

3. **PCA Strategy**: PCA is used to reduce the dimensionality of the channel in this strategy. There are two ways to implement PCA. The first one is to directly extract K principal components as the inputs. The number of K using in this strategy is 16, 32, 64, 128 and 256. 16 principal components already contain more than 99% of the variance in the data. The second way is to extract K-1 principal components and add the post-SINR as the inputs.

As for the generalization test, the general dataset that consists of 10 different scenarios is used. Hence, Doppler spread, MCS Index, modulation order and TBS Index are also important features. The best combination of the algorithm and the feature design explored in the EVA dataset is used to train the model for generalization test.

### 4.3 The classification methods

As mentioned in Chapter 3.3, three supervised learning methods constitute the algorithm pool of this report and each of them is tested with three feature designs respectively. Then, the calibration method, which is an ESM-based approach, is used as a comparison to evaluate the performance of each machine learning method.
4.3.1 Logistic Regression

The logistic regression algorithm implemented in this part uses the default setting from the Python sklearn library.

A. Baseline

Figure 4.2 shows the accuracy of the baseline logistic regression model with a 97.53% general accuracy. The green line represents the logistic regression while the blue line is the calibration method. It is evident that both approaches perform perfectly outside the descending section and the most prediction errors appear in the descending area. The descending accuracy of logistic regression is 91.86%, while the calibration method only has a 74.22% descending accuracy. Apparently, the logistic regression outperforms the calibration method in the EVA dataset.

Figure 4.2: Baseline of Logistic Regression. The green line represents the logistic regression model with baseline feature design, while the blue line represents the calibration method. The green line outperforms the blue line with a 91.86% descending accuracy and a 74.22% descending accuracy, respectively.

B. Feature selection

In order to reduce the feature dimensionality, feature selection strategy picks a subset of features based on human knowledge. Figure 4.3 demonstrates the accuracy of applying this strategy with logistic regression. The general accuracy of this model is 97.47% and the descending accuracy is 91.81%. Compared with the baseline strategy, choosing fewer features from channel H does not affect the accuracy a lot. Moreover, both methods, baseline and feature selection strategy, perform much better than the calibration method.

C. PCA strategy

As described in Chapter 4.2, two different strategies are analyzed with PCA. Table 4.2 are the results of extracting K principal components as the input features. Obviously, the performance is worse compared to the calibration method which has a 74.22% descending
Figure 4.3: The performance of Logistic Regression model with feature selection design. The green line represents logistic regression with the feature selection design, while the blue line represents the calibration method. The descending accuracy of the green line is 91.81% that surpasses the blue line.

accuracy. There is two hypothesis about the reasons causing this poor performance. The first hypothesis is that it is because the post-SINR is not used in this feature setting. In the baseline and feature selection strategy, the post-SINR is one of the inputs and both models perform well. Therefore, the post-SINR may be a major factor that affects the model performance. The other hypothesis is that the logistic regression algorithm is too simple to capture the hidden patterns inside the data.

Table 4.2: Accuracy of applying PCA on the channel H as the input

<table>
<thead>
<tr>
<th>Numbers of principal components</th>
<th>16</th>
<th>32</th>
<th>64</th>
<th>128</th>
<th>256</th>
</tr>
</thead>
<tbody>
<tr>
<td>general accuracy</td>
<td>65.81%</td>
<td>69.39%</td>
<td>71.81%</td>
<td>71.45%</td>
<td>71.54%</td>
</tr>
<tr>
<td>descending accuracy</td>
<td>53.03%</td>
<td>54.48%</td>
<td>56.13%</td>
<td>55.89%</td>
<td>55.75%</td>
</tr>
</tbody>
</table>

After replacing the last principal component with the post-SINR, Table 4.3 shows good results with PCA. The accuracy slightly improves and reaches the maximum point when using 127 principal components. This test proves the hypothesis concerning the importance of the post-SINR. The 92.74% descending accuracy in Table 4.3 is the best performance of logistic regression that is obtained by using 127 principal components and the post-SINR as the input features.

Table 4.3: Accuracy of using principal components and post-SINR as the input

<table>
<thead>
<tr>
<th>Numbers of principal components</th>
<th>15</th>
<th>31</th>
<th>63</th>
<th>127</th>
<th>255</th>
</tr>
</thead>
<tbody>
<tr>
<td>general accuracy</td>
<td>97.58%</td>
<td>97.61%</td>
<td>97.49%</td>
<td>97.77%</td>
<td>97.49%</td>
</tr>
<tr>
<td>descending accuracy</td>
<td>92.05%</td>
<td>92.13%</td>
<td>91.75%</td>
<td>92.74%</td>
<td>91.72%</td>
</tr>
</tbody>
</table>

4.3.2 Support Vector Machine

The Support Vector Machine in this section is also implemented using the sklearn library. Instead of directly using SVC function, the experiments use SGDClassifier function as it
takes less time to train. Furthermore, the iteration time for SGD is 300 epochs.

A. Baseline

Similar to logistic regression, the first analysis tests the baseline SVM model. Figure 4.4 shows the accuracy of SVM (the green line) in the EVA dataset and the general accuracy of SVM is 97.28%. In the descending section, the accuracy of SVM is 91.10%, while the accuracy of the calibration method is only 74.22%. Apparently, SVM outperforms the calibration method but slightly less accurate than the logistic regression which has a 91.86% descending accuracy.

![Baseline of SVM](image)

Figure 4.4: Baseline of SVM. The green line represents SVM with baseline feature design, while the blue line is the calibration method. The green line surpasses the blue line with a 91.10% descending accuracy and a 74.22% descending accuracy, respectively.

B. Feature selection strategy

Figure 4.5 demonstrates the outcome of the manually selected partial features as the input of SVM model that has a 97.42% general accuracy and a 91.49% descending accuracy. The difference of this model with the baseline model is hardly noticeable. Hence, two algorithms, logistic regression and SVM, with baseline and feature selection strategy perform similarly, which mean simple classifiers could achieve a good accuracy as long as the post-SINR is used as one of the inputs.

C. PCA strategy

Table 4.4 and Table 4.5 illustrate the accuracy of two PCA strategies. Analogous to logistic regression, the SVM model behaves badly without using the post-SINR, while the model achieves good performance after adding the post-SINR. The best performance of SVM is also obtained by extracting 127 principal components and together with post-SINR as the input features. However, the best descending accuracy is only 91.96% that is approximately 0.8% less than the best descending accuracy of the logistic regression model. In conclusion, SVM performs slightly worse than logistic regression in general.
32 CHAPTER 4. EXPERIMENTS AND DISCUSSION

Figure 4.5: The performance of SVM model with feature selection design. The green line is the SVM model with feature selection design, while the blue line shows the accuracy of the calibration method. The descending accuracy of the green line is 91.49% that surpasses the blue line.

Table 4.4: Accuracy of the SVM model without using post-SINR

<table>
<thead>
<tr>
<th>Numbers of principal components</th>
<th>16</th>
<th>32</th>
<th>64</th>
<th>128</th>
<th>256</th>
</tr>
</thead>
<tbody>
<tr>
<td>general accuracy</td>
<td>66.29%</td>
<td>67.48%</td>
<td>67.97%</td>
<td>68.94%</td>
<td>66.55%</td>
</tr>
<tr>
<td>descending accuracy</td>
<td>50.47%</td>
<td>49.29%</td>
<td>49.61%</td>
<td>50.22%</td>
<td>47.74%</td>
</tr>
</tbody>
</table>

Table 4.5: Accuracy of the SVM model using post-SINR

<table>
<thead>
<tr>
<th>Numbers of principal components</th>
<th>15</th>
<th>31</th>
<th>63</th>
<th>127</th>
<th>255</th>
</tr>
</thead>
<tbody>
<tr>
<td>general accuracy</td>
<td>97.42%</td>
<td>97.47%</td>
<td>97.43%</td>
<td>97.53%</td>
<td>97.36%</td>
</tr>
<tr>
<td>descending accuracy</td>
<td>91.71%</td>
<td>91.70%</td>
<td>91.70%</td>
<td>91.96%</td>
<td>91.28%</td>
</tr>
</tbody>
</table>

4.3.3 Neural Network

The artificial neural network model is implemented on Python Keras library. It is a binary classification problem, so the activation function of the output layer is a sigmoid function, while the other layers use relu function. To simplify the experiments, the structure of the network is one hidden layer with 256 nodes.

A. Baseline

Figure 4.6 shows the accuracy of the baseline neural network model with a 97.91% general accuracy. The green line represents the neural network model with a 93.13% descending accuracy while the blue line is the calibration method with a 74.22% descending accuracy. The neural network model performs the best in the baseline strategy compared to a 91.86% and a 91.10% descending accuracy of the baseline LR model and the baseline SVM model respectively.

B. Feature selection strategy

Figure 4.7 is the performance of the neural network model using a subset features from the channel H and the post-SINR as the input. The general accuracy of the model is 97.66%
and the descending accuracy is 92.36%. Even though the performance of this model surpasses the calibration method, this feature selection strategy is less accurate than the baseline model. This might be because the selected features cannot represent the whole channel H and some valuable information is lost during the selection process. Apparently, the neural network model is a more complex model compared to LR and SVM that can utilize more input features and discover the hidden data structures.

Figure 4.7: **The Neural Network model with feature selection design.** The green line is the neural network model using a subset of features as the inputs, while the blue line shows the accuracy of the calibration method. The descending accuracy of the green line is 92.36%, which surpasses the blue line.

C. PCA strategy

In this experiment, two of the same PCA strategies are tested with the neural network model. However, the results are quite different compared with the same experiment on
LR and SVM. Table 4.6 shows a high accuracy of the neural network model when the principal components are the only inputs. This result proves the hypothesis that a complex classifier can discover the hidden patterns inside the channel H. The reason why logistic regression and SVM do not work is that the complexity of these two classifiers is low. Moreover, the improvement of the accuracy is slight after adding the post-SINR compared to Table 4.6 and Table 4.7, which means the post-SINR is not a major feature for the neural network model.

Table 4.6: The performance of the NN model only using principal components as inputs

<table>
<thead>
<tr>
<th>Numbers of principal components</th>
<th>16</th>
<th>32</th>
<th>64</th>
<th>128</th>
<th>256</th>
</tr>
</thead>
<tbody>
<tr>
<td>general accuracy</td>
<td>97.56%</td>
<td>97.57%</td>
<td>97.50%</td>
<td>97.61%</td>
<td>97.88%</td>
</tr>
<tr>
<td>descending accuracy</td>
<td>92.12%</td>
<td>92.12%</td>
<td>91.89%</td>
<td>92.26%</td>
<td>93.07%</td>
</tr>
</tbody>
</table>

Table 4.7: The performance of the NN model using principal components and post-SINR as inputs

<table>
<thead>
<tr>
<th>Numbers of principal components</th>
<th>15</th>
<th>31</th>
<th>63</th>
<th>127</th>
<th>255</th>
</tr>
</thead>
<tbody>
<tr>
<td>general accuracy</td>
<td>97.73%</td>
<td>97.81%</td>
<td>97.89%</td>
<td>97.91%</td>
<td>97.89%</td>
</tr>
<tr>
<td>descending accuracy</td>
<td>92.55%</td>
<td>92.86%</td>
<td>93.07%</td>
<td>93.19%</td>
<td>93.09%</td>
</tr>
</tbody>
</table>

4.3.4 Summary

In conclusion, all three machine learning algorithms outperform the calibration method in the EVA dataset, which proves that the machine learning methods work for this problem and can provide higher accuracy. Among the three algorithms, SVM performs the worst, which means it is not a suitable algorithm for this problem and the training time for the traditional SVM method is long. The advantage of logistic regression is the short learning time with an acceptable accuracy. However, logistic regression needs to use the post-SINR for prediction. Without using the post-SINR as the input, the performance of the logistic regression model is worse than the calibration method. The neural network model has the best prediction accuracy and performs well without the post-SINR. However, the neural network usually has a lot of parameters depending on the number of layers and the number of nodes in each layer, which requires a long training time.

For the transmission prediction problem, the final model is expected to avoid using human-made features like the post-SINR because it is not available for all types of receivers. Therefore, the neural network model is the best algorithm for this problem and the suitable feature design is to directly use principal components.

4.4 Further exploration

Previous experiments prove that the neural network is the best algorithm for the given problem. However, the structure employed in the above experiments is a random choice. The purpose of this section concentrates on the “art” of the network design that explores the fundamental network structure. The input strategy is directly extracting 256 principal components from channel H that makes the model more general for different kinds of receivers.
The most important decision about a neural network structure is to choose the number of hidden layers as well as the number of neurons per layer. Table 4.8 illustrates the usage of different numbers of hidden layers [19]. There is no theoretical evidence to support using more than two hidden layers in the neural network. The famous deep learning network structures like AlexNet[20], VGG[21] and GoogleNet[22] use at most two fully connected layers except the output layer and softmax layer. Therefore, the exploration of network structure only considers one and two hidden layers.

Table 4.8: The usage of different number of hidden layers [19]

<table>
<thead>
<tr>
<th>Number of hidden layers</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Only capable of representing linear separable functions or decisions</td>
</tr>
<tr>
<td>1</td>
<td>Can approximate any function that contains a continuous mapping from one finite space to another</td>
</tr>
<tr>
<td>2</td>
<td>Can represent an arbitrary decision boundary to arbitrary accuracy with rational activation functions and can approximate any smooth mapping to any accuracy</td>
</tr>
</tbody>
</table>

As for the number of neurons, some books and articles give suggestions on choosing the architecture:

- The size of the hidden layer should be somewhere between the input layer size and the output layer size [23].
- The number of hidden nodes should no more than the twice of the inputs [24].
- The number of hidden nodes is specified to capture $70 - 90\%$ of the variance of the input dataset [25].

However, some researchers believe these "rules of thumb" make no sense because they ignore the number of training examples, the noise in the dataset, the complexity of chosen algorithm and the regularization method. To date, there is no absolute rule about how to choose the number of hidden layers and hidden nodes. Therefore, several sets of experiments are implemented below to find a suitable structure of the neural network model.

Table 4.9 tests the performance of one hidden layer structure with 32, 64, 128 and 256 nodes, respectively. Both general accuracy and descending accuracy increases with the expansion of the node number. Moreover, it increases slowly after the number of nodes more than 128. Since the model with 256 nodes (the highlighted part in the Table 4.9) performs the best, one more hidden layer is added to this model and Table 4.10 shows the outcome of choosing different numbers of nodes in the second hidden layer. The accuracy increases slightly from 16 to 128 nodes and stops increasing from 128 nodes, which means, from this point, increasing the number of second hidden layer nodes does not equal to the improvement of accuracy. Besides the basic hidden neurons tests, Table 4.11 describes the results of some variant structures, which proves that the number of nodes in the first hidden layer should be at least larger than the second hidden layer.

To sum up, there are several conclusions observed from tables 4.9 to 4.11.

- The model has better performance when the number of hidden layers increases from one to two.
Table 4.9: Performance of different nodes with one hidden layer (Highlight part is the best model int this experiment)

<table>
<thead>
<tr>
<th>Name</th>
<th>Model Structure</th>
<th>Parameters</th>
<th>Training Time</th>
<th>general accuracy</th>
<th>descending accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>node_32</td>
<td>256<em>32</em>1</td>
<td>8257</td>
<td>62s</td>
<td>96.70%</td>
<td>89.37%</td>
</tr>
<tr>
<td>node_64</td>
<td>256<em>64</em>1</td>
<td>16513</td>
<td>82s</td>
<td>97.26%</td>
<td>91.02%</td>
</tr>
<tr>
<td>node_128</td>
<td>256<em>128</em>1</td>
<td>33025</td>
<td>123s</td>
<td>97.53%</td>
<td>91.92%</td>
</tr>
<tr>
<td>node_256</td>
<td>256<em>256</em>1</td>
<td>66049</td>
<td>216s</td>
<td>97.59%</td>
<td>92.10%</td>
</tr>
</tbody>
</table>

Table 4.10: Performance of different nodes with two hidden layers (Highlight part is the best model int this experiment)

<table>
<thead>
<tr>
<th>Name</th>
<th>Model Structure</th>
<th>Parameters</th>
<th>Training Time</th>
<th>general accuracy</th>
<th>descending accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>model_256_16</td>
<td>256<em>256</em>16*1</td>
<td>69921</td>
<td>202s</td>
<td>97.63%</td>
<td>92.23%</td>
</tr>
<tr>
<td>model_256_32</td>
<td>256<em>256</em>32*1</td>
<td>74049</td>
<td>220s</td>
<td>97.69%</td>
<td>92.44%</td>
</tr>
<tr>
<td>model_256_64</td>
<td>256<em>256</em>64*1</td>
<td>82305</td>
<td>255s</td>
<td>97.68%</td>
<td>92.39%</td>
</tr>
<tr>
<td>model_256_128</td>
<td>256<em>256</em>128*1</td>
<td>98817</td>
<td>281s</td>
<td>97.72%</td>
<td>92.54%</td>
</tr>
<tr>
<td>model_256_256</td>
<td>256<em>256</em>256*1</td>
<td>131841</td>
<td>289s</td>
<td>97.71%</td>
<td>92.50%</td>
</tr>
</tbody>
</table>

Table 4.11: Variants of two hidden layers network

<table>
<thead>
<tr>
<th>Name</th>
<th>Model Structure</th>
<th>Parameters</th>
<th>Training Time</th>
<th>general accuracy</th>
<th>descending accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>model_32_256</td>
<td>256<em>32</em>256*1</td>
<td>16929</td>
<td>155s</td>
<td>97.54%</td>
<td>91.95%</td>
</tr>
<tr>
<td>model_64_256</td>
<td>256<em>64</em>256*1</td>
<td>33345</td>
<td>197s</td>
<td>97.67%</td>
<td>92.37%</td>
</tr>
<tr>
<td>model_128_256</td>
<td>256<em>128</em>256*1</td>
<td>66177</td>
<td>238s</td>
<td>97.67%</td>
<td>92.37%</td>
</tr>
</tbody>
</table>

- In two hidden layers structure, the number of nodes in the second hidden layer should be smaller than the first hidden layer.
- The best network model has the structure of $256 \times 256 \times 128 \times 1$, which is used for the following experiments.

### 4.5 Generalization test

For verifying the generalization of the model, a two-hidden-layer structure with 256 nodes and 128 nodes respectively is chosen and the input features of this experiment consists of 256 principal components, Doppler spread, MCS Index, Modulation order and TBS Index. The accuracy and loss for both training set and validation set are shown in the Figure 4.8. The best training accuracy is 99% and the best validation accuracy is 96.5%. It is evident that the accuracy greatly increases before 50 epochs and then the model starts to overfit after 100 epochs.

In order to prevent overfitting, a dropout layer is added. The dropout is proposed as a method to avoid overfitting in the deep learning structure [26]. The dropout method plays a similar role to the ensemble learning. The dropout layer randomly shuts down some units, which means setting the output of those units to zero. Those units would not participate in the forward pass and the back-propagation on this iteration. The number of nodes, which will be dropped out, is controlled by a parameter representing a percentage of the output nodes in the previous layer. In each iteration, different units are dropped out, which is similar to using different models to predict. Figure 4.9 shows the
accuracy and loss of the Drop model, which adds a dropout layer after the first hidden layer with 30% probability that shuts down the units. The best training accuracy of this model is 97.27% and the best validation accuracy is 96.43%. The reason why the best accuracy of Drop model is slightly less than the model without using the dropout might be because the Drop model still does not converge. Using the dropout, the model needs more time to converge because after each iteration, it updates the remaining units.

![Figure 4.8: The accuracy and loss for the two hidden layers model. The figure shows the accuracy and loss for both training set and validation set, and 300 epochs are used for training the model. After 100 epochs, the model starts overfitting.](image1)

![Figure 4.9: The accuracy and loss for the Drop model. The Drop model adds a dropout layer after the first hidden layer, which prevents the model overfitting to the training set.](image2)

From Table 4.12, the NN model exceeds the calibration method when Doppler spread is equal to 5, while two methods perform similar when Doppler spread is equal to 70. In general, the descending accuracy is quite steady with more than 90% accuracy of 5 HZ Doppler spread and around 80% accuracy of 70 HZ Doppler spread. On the contrary, the calibration method performs much better when the MCS index is in the middle range, such as 10 or 15. The large precision variation of the calibration method is due to the variety of BLER curves. Figure 4.10 shows the accuracy of four different scenarios. Comparing Figure 4.10a and Figure 4.10b as well as Figure 4.10c and Figure 4.10d, the slope of BLER curve increases with the increase of the MCS index. In conclusion, the neural network model performs equally in all MCS index values, while the calibration method performs well in the mid range of the MCS index values and weak in others.
Table 4.12: Generalization of the Neural Network model (Highlight part is where the calibration method performs good)

<table>
<thead>
<tr>
<th>doppler spread</th>
<th>Calibration method</th>
<th>NN model</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>63.29% 74.17% 88.96% 87.30% 72.00%</td>
<td>91.97% 92.48% 94.70% 94.13% 92.82%</td>
</tr>
<tr>
<td>70</td>
<td>66.65% 77.25% 89.03% 90.53% 82.79%</td>
<td>82.18% 82.20% 84.49% 85.87% 78.93%</td>
</tr>
</tbody>
</table>

Figure 4.10: The examples of the performance under different scenarios. When Doppler spread equals to 5, the neural network model surpasses the calibration method, while two methods have similar performance when Doppler spread equals to 70.
Chapter 5

Conclusions

5.1 Conclusion

A machine learning based method is proposed to solve the error detection problem in system simulations. The algorithms used in this report are Logistic Regression, Support Vector Machine and Neural Network. Two datasets are used to analyze the performance of three algorithms. The first dataset contains the features of one channel environment, while the second dataset consists of the features of ten channel environments. The experimental results demonstrate that the three algorithms outperform the traditional approach and the neural network model is better than the other methods. Furthermore, different structures of neural networks are evaluated and the accuracy is measured. The final experiment investigates the generalization performance of the neural network model. The best model found in this thesis is the neural network model, with two hidden layers and the nodes of second hidden layer less than the first hidden layer, which performs better than others.

5.2 Further works

Further work should focus on solving the error detection problem using deep neural networks. The artificial neural network model evaluated in this report proves that the network structure can capture the hidden patterns in the channel. However, a two-layer network might not be sophisticated enough to capture all information inside the channel environments. Therefore, a deep neural network might improve the prediction accuracy.

On the other hand, the classification of transmission states exhibits a potential application of machine learning as well as a new approach for wireless communication fields. This report is a starting point for further investigations on the physical layer. Further work might involve utilizing the time dependency of the channel environment to predict the transmission states. For instance, the previous transmission data can be used to predict the current transmission state.
References


