Reducing Errors in Optical Data Transmission Using Trainable Machine Learning Methods

Submitted to the University of Hertfordshire in partial fulfilment of the requirements for the degree of

Doctor of Philosophy

Weam M. Binjumah

Department of Computer Science

September 2018

This thesis is dedicated

... to the memory of my father, **Engr. Mohammed Bin Jumah**, who could not see this thesis completed, but he always believed in my ability to be successful in the academic arena. He is gone, but his belief in me has made this journey possible.

... to my beloved mother, **Mrs. Maqboolah**, for her presence that always gives me strength and support to face the challenges, and urges me to strive to achieve my goals in life. She is a constant source of inspiration in my life.

... to my siblings for their endless love, support and encouragement.

– *Weam Binjumah*

Acknowledgements

I would like to express my special appreciation to my supervisors Dr. Yi Sun, Dr. Neil Davey, Prof. Rod Adams and Dr. Alexey Redyuk for their guidance, great support and kind advice throughout my Ph.D. research studies. It was a real privilege and an honor for me to share their exceptional scientific knowledge and extraordinary human qualities. They helped me keep things in perspective, and cared so much about my work. They also never hesitated to answer my questions and queries so immediately.

I want to specially thank my principal supervisor Dr. Yi Sun, for allowing me to grow as a research scientist. Her good guidance, her abundant knowledge and style quality had a deep impact on the completion of this project. A special thanks to Dr. Neil Davey, for the endless guidance and support he provided throughout my time as his student; Professor Rod Adams for the tremendous support and his enlightening suggestions that put me on the right way; and Dr. Alexey Redyuk for helping me with understanding the optical communication systems field.

I am very thankful to all colleagues and friends of the Biocomputation research group for their companionship and for providing a pleasurable and friendly working atmosphere during my studying at the University of Hertfordshire. The moments of leisure shared together helped me overcome hard times. I am also thankful for their immediate help, whenever I needed it.

I'm so grateful to my late father, who I miss every day; my beloved mother; my brothers: Dr. Sami and Mr. Fahad; and my sisters: Ms. Areej, Ms. Maissa, Ms. Hebah, Dr. Murooj and Dr. Ghadeer. I would not be where I am today without their unconditional love and support.

Finally, I would like to thank my home country the Kingdom of Saudi Arabia for providing the scholarship which allowed me to undertake this research, and for giving me this opportunity to complete my Ph.D. degree. I am thankful to Taibah University, Community College, for providing the financial support for my research study.

Abstract

Reducing Bit Error Ratio (BER) and improving performance of modern coherent optical communication system is a significant issue. As the distance travelled by the information signal increases, the bit error ratio will degrade. Machine learning techniques (ML) have been used in applications associated with optical communication systems. The most common machine learning techniques that have been used in applications of optical communication systems are artificial neural networks, Bayesian analysis, and support vector machines (SVMs). This thesis investigates how to improve the bit error ratio in optical data transmission using a trainable machine learning method (ML), that is, a Support Vector Machine (SVM). SVM is a successful machine learning method for pattern recognition, which outperformed the conventional threshold method based on measuring the phase value of each symbol's central sample. In order that the described system can be implemented in hardware, this thesis focuses on applications of SVM with a linear kernel due to the fact that the linear separator is easier to be built in hardware at the desired high speed required of the decoder.

In this thesis, using an SVM to reduce the bit error ratio of signals that travel over various distances has been investigated thoroughly. Especially, particular attention has been paid to using the neighbouring information of each symbol being decoded. To further improve the bit error ratio, the wavelet transforms (WT) technique has been employed to reduce the noise of distorted optical signals; however the method did not bring the sort of improvements that the proponents of wavelets led me to believe.

It has been found that the most significant improvement of bit error ratio over the current threshold method is to use a number of neighbours on either side of the symbol being decoded. This works much better than using more information from the symbol itself.

Keywords: Machine learning (ML), Support vector machine (SVM), Signal processing, Fiber optics, Wavelets, Bit errors ratio (BER), classification, Optical communication systems.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

In recent years, optical fibres have been widely used in transmitting information over long distances with a high bandwidth demand (that is, that maximum rate of data transfer across a given path[1]), for example, in Internet communication. As bandwidth demands increase and the tolerance for errors decreases, there has been an increasing interest in the need for improving the quality of transmission. Improving the Bit Error Ratio (BER)[2] in optical transmission systems is a challenging problem.

The basic idea of optical communication is carrying the information encoded into the light beam, to transmit the data between two points over different distances. Typical optical communication systems consist of three main components, which are illustrated in Figure 1.1: an optical transmitter (Tx [3]) that converts the electrical signal into an optical signal, an optical fibre as the propagation medium of the optical signal, and an optical receiver (Rx [4]) that converts the received optical signal into an electrical signal again.

To make the light carry the digital signals, the light is modulated using one of the modulation methods such as QPSK (more details, see Chapter 2). There are many different causes of transmitted signal degradation in optical communication systems (Bernstein et al., 2003). During the transmission, the optical signals are exposed to many kinds of impairments such as attenuation, dispersion broadening and nonlinear distortion (Kanprachar, 1999). By the end at the destination, these impairments can lead to the optical transmission signals being decoded incorrectly and generate some information bits being in error at the receiver of the fibre link. Increasing the distance travelled by the signal leads to a further loss in the quality of the signal and a larger Bit Error

[1] https://simple.wikipedia.org/wiki/Bandwidth$_($*computing*)

[2] Bit Error Ratio is defined by dividing the number of bit errors by the total number of bits, more details can be seen in Section 4.6, page 72.

[3] "The optical transmit (Tx) Power is the signal level leaving the transceiver, and it should fall within the transmitter output power range", quoted from (Rhoads, 2016)

[4] "The optical receive (Rx) power is the incoming signal level being received from the far end transceiver", quoted from (Rhoads, 2016)

[5] "Fibre" has been used in this thesis. However, since this figure (Figure 1.1) and the other figure (Figure 2.2) are cited from literature, "fiber" has been used.

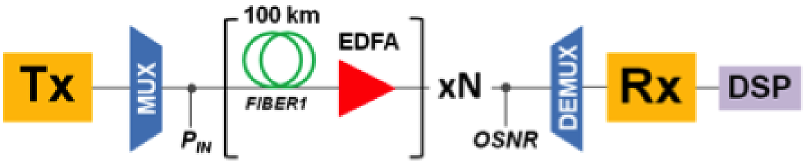


Figure 1.1: The optical fibre link configuration (Binjumah et al., 2015a). The transmission fibre has a length of 100 km. P_{in} is a function of the fibre launch power at each span (Chughtai, 2012) (Matsumoto, 2013). N denotes the number of spans, where the span is the cable length between two amplifiers (Shahi et al., 2014) (StackExchange, 2017). EDFA, MUX, DEMUX, OSNR and DSP are the short forms of erbium-doped fibre amplifier, multiplexer, de-multiplexer, optical signal to noise ratio and digital signal processor, respectively (for more details, see Chapter 2). My work is concerned with improving the performance of the decoding at the receiver (^{5}).

Ratio (BER) degradation (Binjumah et al., 2015b). Obviously the high-speed and long distance data transmission in optical systems needs to be accompanied with as low bit error ratio as possible (Metaxas et al., 2013). Therefore, the reduction of Bit Error Ratio (BER) in optical data transmission is a significant issue and is difficult to achieve. My work is concerned with improving the performance of the decoding at the receiver (de-modulator in Figure 1.2) using machine learning techniques (ML), and reducing the bit errors resulting from mis-classifying the optical transmission signals. Hence this work is not error correction, which is usually achieved using some method of redundancy in the data, but in reducing the actual number of errors that are made at the decoding receiver.

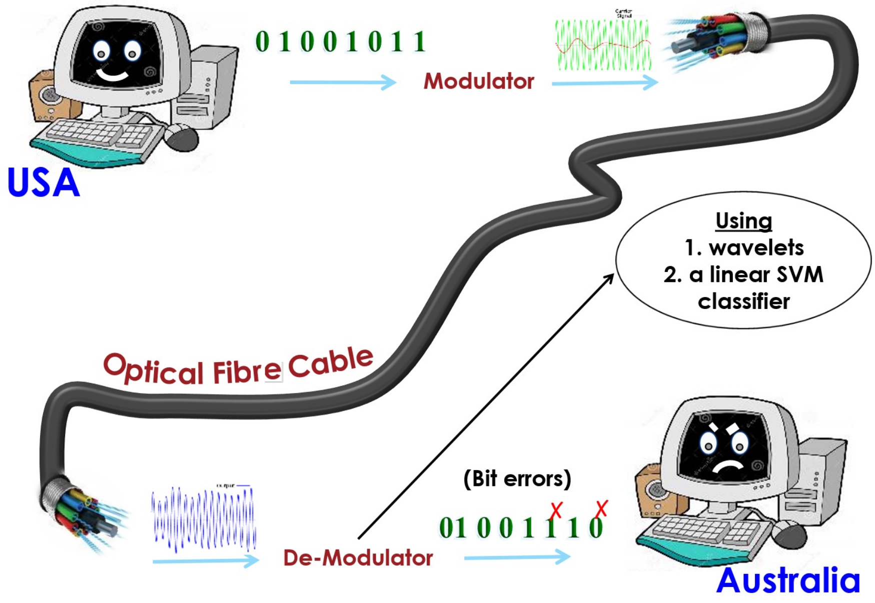


Figure 1.2: My work is concerned with improving the performance of the decoding at the receiver (de-modulator) using machine learning techniques (ML), and reducing the bit errors resulting from mis-classifying the optical transmission signals inaccurately.

Machine learning is ideally suited for reducing the bit error ratio in optical data transmission. Initial work in using ML techniques to improve bit error ratios in adaptive electrical signal post-processing in optical communication system has been analyzed in (Sun et al., 2008), (Hunt et al., 2009) and (Hunt et al., 2010). Sun et al. (2008) have proposed using neighbouring information to form different input (feature) vectors, and had a first look at examining its effectiveness for error reduction of the signal.

Hunt et al. (2009) have further evaluated the method shown in (Sun et al., 2008) by applying various feature extraction methods including wavelet transformation.

Thrane et al. (2017) show an overview of using different machine learning techniques and their application in optical communication. One of these techniques is the support vector machine that was used for modulation format classification (Thrane et al., 2017).

The Support Vector Machine (SVM) is a successful machine learning method for pattern recognition. The initial study in (Metaxas et al., 2013) has shown that linear Support Vector Machines (SVMs) outperformed other trainable classifiers for error reduction in optical data transmission and mentioned that it should be easier to be built in the hardware for real time use. However, none of these authors have investigated either how the linear SVM performs when signals travel over various distances or the full effect of using neighbouring information as part of the input to the classifier. The work in this thesis is the first systematic study of all these issues. It also uses the latest in simulated data, and therefore offers more robust performance values.

Building an SVM classifier into a real-world system offers a promising future for achieving high performance computation while keeping the power consumption and costs low (Afifi et al., 2015). Recently, it has been proposed that a non-linear SVM can be implemented using a Field-Programmable Gate Array (FPGA) (Afifi et al., 2015) and that this too may be fast enough for real-world applications in the future. This has led to further investigation on the performance of a non-linear SVM in this work.

1.2 Research Questions

My research deals with the fundamental question of how to improve the Bit Error Ratio (BER) in optical data transmission using a trainable machine learning method, that is, a Support Vector Machine (SVM). This is to be done by classifying symbols at the receiving end more accurately so that less errors are made when detecting them. The major challenge of this research is to provide accurate predictions especially at and beyond a long distance of 8,000 km. Figure 1.3 shows that the tolerable BER is 2×10^{-2} (the horizontal line). The green curve displays BER obtained using the traditional threshold method (see Chapter 4). It can be seen that the BER obtained using the threshold method exceeds the tolerable value after about 7,000km. Research questions in my study include:

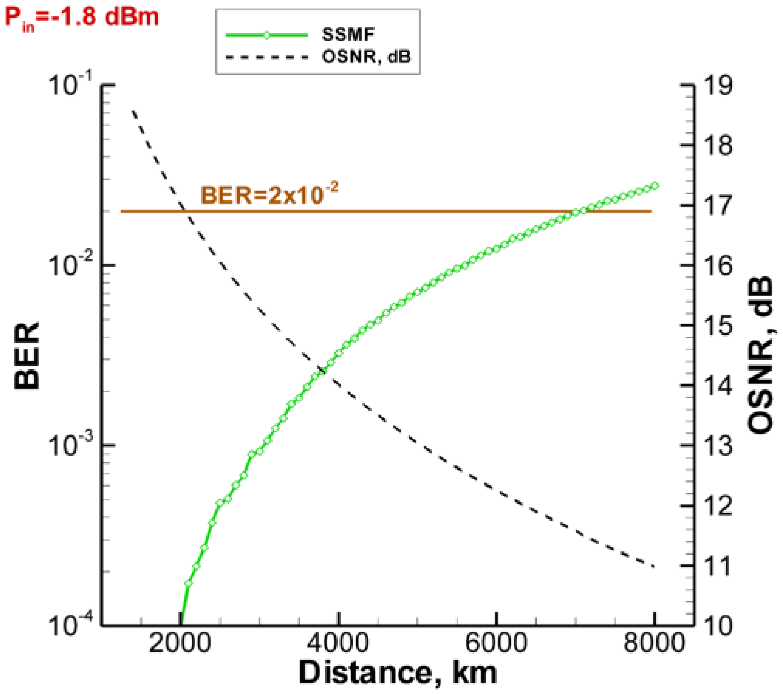


Figure 1.3: Using the traditional classifier (threshold method) for this data exceeded the tolerable bit error ratio (BER) after 7,000 km, which is 2×10^{-2} (Mizuochi et al., 2004). Note: BER stands for Bit Error Ratio, SSMF denotes Standard Single Mode Fibre, OSNR stands for Optical Signal-To-Noise Ratio that compares the level of a desired signal to the level of background noise (Wikipedia, 2018e) (it is measured by decibel (dB[6])), and P_{in} denotes the input power to the fibre generated by the transmitter used to launch the signal (Massa, 2000) (it is measured by decibel-milliwatts (dBm[7])).

1. Investigate thoroughly whether or not a linear SVM can be suitable for improving the bit error ratio in optical data transmission over distances from 1,000km to 10,000km.

2. What methods can be used with classifiers so that together they further improve BER?

   - Sun et al. (2008) have proposed the idea using neighbouring information. However, they did not validate the methods over different distances travelled by signals, nor did they investigate more distant neighbours, and the work did not have access to the latest simulated data. In this work, I shall investigate how many neighbours should be used over various distances.

   - Furthermore, I shall answer the question: to what degree, can wavelets be of help to assist in improving bit error ratio on the distorted optical signals? Especially, I look into whether or not wavelets can deal with noise in phase and/or frequency of optical signals.

[6] A decibel (dB) is a unit used to express relative differences in signal strength (CISCO, 2005).

[7] dBm is used in fibre-optical communication networks as a convenient measure of absolute power because of its capability to express both very large and very small values (Wikipedia, 2018c)

3. How reliable the proposed method is, that is, using an SVM together with feature selection methods, when working on the meaningful-text data sets?

1.3 Contributions

The novel contributions in this research includes:

- I have thoroughly investigated the effect of using the neighbouring symbols to inform training. My results show that it works well on most of my data over various distances traveled by signals. In fact it is the most effective addition to the current method, namely the threshold method. This is my most important contribution to knowledge.

- I have confirmed that the bit error ratio can be improved by using a trainable machine learning model, that is, an SVM-based classifier. Especially, the proposed method works well on the meaningful text data set, and I confirmed this result using much more comprehensive and realistic data than has been used before.

- I have empirically demonstrated that the linear SVM, which is a hardware realisable algorithm, can be of use in improving the bit error ratio, especially for simulated distances less than 8,000km.

- Despite the widespread use of wavelets in signal processing I have found little benefit in using them in the context of my work.

1.4 Publications on this Thesis

Conference paper, see Appendix D

- **Reducing Bit Error Rate of Optical Data Transmission with Neighboring Symbol Information Using a Linear Support Vector Machine** by Weam M. Binjumah, Alexey Redyuk, Neil Davey, Rod Adams and Yi Sun - presented at the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, from 7 to 11 September, 2015 in Porto, Portugal.

- **Error Correction over Optical Transmission** by Weam M. Binjumah, Alexey Redyuk, Rod Adams, Neil Davey and Yi Sun - presented at ICPRAM 2017; 6th International Conference on Pattern Recognition Applications and Methods, from 24 to 26 February, 2017 in Porto, Portugal.

- **Investigating Optical Transmission Error Correction using Wavelet Transforms** by Weam M. Binjumah, Alexey Redyuk, Rod Adams, Neil Davey and Yi Sun - presented at

ESANN 2017; Investigating Optical Transmission Error Correction using Wavelet Transforms, from 26 to 28 April, 2017 in Bruges, Belgium.

1.5 Typographical Conventions

In this thesis, I have used "10" in two different ways as follows:

- **Symbol "10":** 10 in quotes is a binary number. It reads as symbol one zero, which is a two-bit symbol. In this work, there are four unique symbols (classes), which are Symbol "10", Symbol "11", Symbol "01" and Symbol "00". Details can be found in Chapter 3.
- **10 symbols:** here, 10 is a decimal. It reads ten symbols.

1.6 Expression of Phase Values

In this thesis, I have some figures that show the phase value along the y-axis. For example, see Figure 3.18, where the phase value is ranged from zero to 2π. In some other figures, the phase value has been shown from zero to π and then from $-\pi$ to zero. I would clarify that generating plots in this thesis by using either way gives the same results, as illustrated in Figure 1.4.

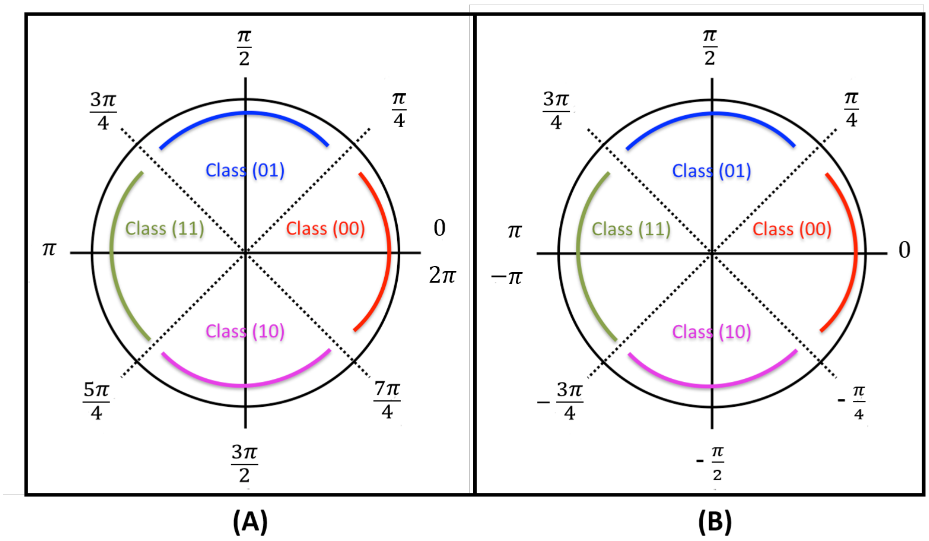


Figure 1.4: Four different variants of the light after QPSK modulation. Panel (A) shows the phase value in only positive angle. Panel (B) shows the phase value in both positive and negative angles.

1.7 The Computational Time in this Work

The computer used for carrying out experiments has an Intle (R) processor with Core (TM) i7 and a 2.5 GHz CPU. The installed memory (RAM) is 8.00 GB. The operating system is 64-bit Windows 7 Enterprise.

Most of the computational time was spent in training the classifier, when trying to find the optimal parameters. The time spent in running each experiment depends on mainly: 1) the number of features in the input vector; 2) the size of the parameter space (more details can be found in Section 4.3.6); and 3) the RAM's capability. In general, it took about 4 days to one week to obtain one result (that is one line) shown in those tables (for example, Table 8.2).

1.8 Thesis Outline

The rest of the thesis is organized as follows:

Chapter 2 introduces the background of data transmission including reviewing some questions about optical fibre links and the modulation of the light. It also gives a literature review of using machine learning techniques for signal processing applications, especially applications associated with the optical communication systems, and of classifying the optical transmission signals.

Chapter 3 is in two parts: a description of all the data, that I have used for the purpose of this study; and an introduction of data pre-processing including a brief description of wavelet transforms, used to represent the data as input vectors of classifiers in this study.

Chapter 4 shows all the methods that I have used in this work, including the benchmark method (the threshold method), which is currently used in the hardware. It also introduces principal component analysis (PCA), which I have used for visualizing data. In addition, Chapter 4 describes the support vector machine (SVM), which is the main method I have used for improving the classification in this work.

Chapter5 presents all the initial experimental results that have been obtained using an SVM on the first type of optical transmission data. It describes an initial investigation of the methodology that will be used in subsequent experiments.

Chapter6 presents the results for the second type of modulated data, which is the main data in my research. This data represents probably the most realistic simulated data. The main focus of my experiments in this chapter is to find out which set of features, including using neighbour information, gives the best results.

Chapter7 shows experiments and results using an SVM with wavelets. This chapter investigates the effect of applying the wavelet transforms (WT) prior to using an SVM.

Chapter8 presents results obtained using data based on a meaningful text dataset (the third type of data). It discusses the effect of using the SVM classifier and wavelet transforms on the optical transmission data that is simulated based on a meaningful text input.

Chapter 9 summarise all the findings in this thesis. Moreover, some future work is indicated and discussed.

Chapter 2

Background and Literature Review

This chapter consists of two parts. In the first part, I shall introduce background information on the basic knowledge of the transmission of optical data. The second part is a literature review, showing relevant research using machine learning methods in the problem domain.

2.1 Background

Various books and web resources were used as sources for this section. Such sources have been cited in each sub-section. In general, these resources are very good, and in some cases were so well written and said exactly what I wanted to say so I have quoted them verbatim. This also keeps the information as accurate as possible. Where the source is quoted verbatim I have given the corresponding references right under each topic heading. All this directly quoted text is then indented.

2.1.1 A brief history of data transmission systems

Data communication began with a simple wired link. Probably the first use of data transmission using copper wire as an electrical conductor was when Samuel Morse sent the first telegraphic message from Washington, D.C., to Baltimore (Library of Congress, n.d.). He invented Morse Code and so the telegraph was the beginning of the digital communication using these electrical pulses (Glesk, 2010).

In 1876, the telephone was invented by Alexander Graham Bell, which brought about a major change in the way people interacted with each other. This lead to the development of other types of cable including the coaxial cable used to communicate one television channel, that based on 3 MHz system. The coaxial cable has many limitations such as the limited distance that signals can travel before being 'refreshed' using a signal re-generator. Some help on this front was afforded by the use of microwave technology.

All modern long-distance cable systems though use fibre optic cables. Their use had to wait until the development of a coherent light source - the laser. Since then fibre optic technologies became the prevailing way to broadcast data. The combination of fibre optics, optical data multiplexing techniques, and advanced signal processing helped to realize the current data transmission system. Figure 2.1 shows a brief history of data broadcasting and communication.

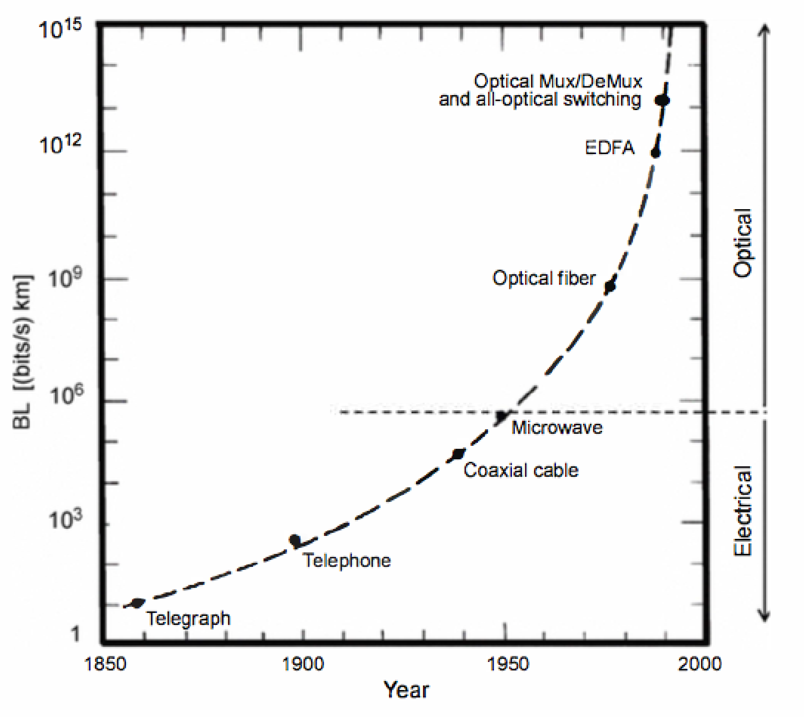


Figure 2.1: A brief history of data broadcasting and communication (Glesk, 2010). Note: in (BL), (B) indicates bit rate and (L) indicates the distance.

2.1.2 General characteristics of optical fibre

This section is quoted from (McCool, n.d.)

> An optical fibre is a strand of silica based glass. Its diameter is slightly thicker than that of a human hair, surrounded by a transparent cladding. Light can be transmitted along the fibre over great distances at very high data transfer rates providing an ideal medium for the transport of information is used. Figure 2.2 shows the typical structure of a basic optical fibre used for communication links. It has an inner glass core with an outer cladding. This is covered with a protective buffer and outer jacket. This design of fibre is light and has a very low loss making it ideal for the transmission of information over long distances. The light propagates along the fibre by the process of total internal reflection (for more information, see Section 2.1.6), and it is contained within the glass core and cladding by careful design of their refractive indices. The loss along the fibre is low and the signal is not subject to electromagnetic interference which plagues other methods of signal transmission, such as

radio or copper wire links. However, The signal is degraded by other means particular to the fibre such as dispersion and non linear effects (caused by a high power density in the fibre core).

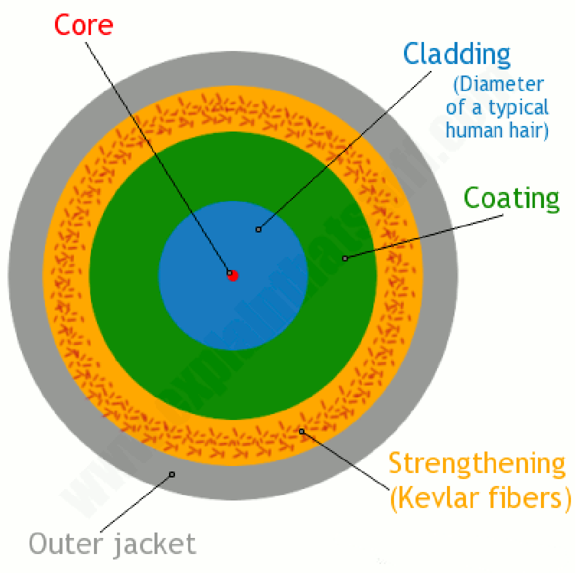

Core
Cladding
(Diameter of a typical human hair)
Coating
Strengthening (Kevlar fibers)
Outer jacket

Figure 2.2: Fibre-Optic cable construction. (Woodford, 2018). Commonly, the size of the optical fibre is specified by its core, cladding and coating (Belden Inc., 2018). The core is between 8 (a typical single-mode fibre) and 62.5 (a typical multi-mode fibre) microns (millionth of a meter (μm)) in diameter. The core is surrounded by the cladding, that is typically 125 microns in diameter and is a little bit thicker than a typical human hair, which is about 100 microns in diameter. The coating is typically half the diameter of cladding, and provides an extra protection to the fibre. The core and the cladding together with the coating can increase the overall size of the optical fibre to 250 microns (Woodward, 2014).

2.1.3 How thick is the optical fibre?

The simplest type of optical fibre is called *single-mode*. It has a very thin core of 5-10 microns (millionths of meter) in diameter. Cable TV, Internet, and telephone signals are generally carried by single-mode fibres, wrapped together into a huge bundle. Cables like this can send information over 100 km (60 miles).

Another type of fibre-optic cable is called *multi-mode*. Each optical fibre in a multi-mode cable is about 10 times bigger than one in a single-mode cable. Multi-mode cables can send information only over relatively short distances and are used (among other things) to link computer networks together.

Even thicker fibres are used in a medical tool called a *gastroscope* (a type of endoscope), which doctors use to see problems inside the patient's stomach. There is also an industrial version of the tool, called a fibre-scope, which can be used to examine things like inaccessible pieces of machinery in airplane engines (Woodford, 2018).

2.1.4 How many signals can be transmitted in optical fibres?

This section is quoted from (Wikipedia, 2018d) and (Wikipedia, 2018f).

The per-channel light signals propagating in the fibre have been modulated at rates as high as 111 gigabits per second (Gbit/s) by NTT (Nippon Telegraph and Telephone Corporation) (NTT, News Release, 2006) (Alfiad et al., 2008), although 10 or 40 Gbit/s is typical in deployed systems (Yao, 2003). In June 2013, researchers demonstrated transmission of 400 Gbit/s over a single channel using 4-mode orbital angular momentum multiplexing (Bozinovic et al., 2013).

Each fibre can carry many independent channels, each using a different wavelength of light (wavelength-division multiplexing (WDM)) (see below for the definition). The net data transfer rate per fibre is defined as the per-channel data rate reduced by the forward error correction (FEC) overhead, multiplied by the number of channels (usually up to eighty in commercial dense WDM systems as of 2008). As of 2011 the record for bandwidth on a single core was 101 Tbit/s (370 channels at 273 Gbit/s each) (Hecht, 2011). The record for a multi-core fibre as of January 2013 was 1.05 petabits per second (Peach, 2013). In 2009, Bell Labs broke optical transmission record, and set a new record of 100 Petabit per second kilometer barrier (that is, 15.5 Tbit/s over a single 7,000 km fibre).

Wavelength-division multiplexing (WDM)

In fibre-optic communications, wavelength-division multiplexing (WDM) is a technology which multiplexes a number of optical carrier signals onto a single optical fibre by using different wavelengths (i.e., colors) of laser light. This technique enables bidirectional communications over one strand of fibre, as well as multiplication of capacity.

The term wavelength-division multiplexing is commonly applied to an optical carrier, which is typically described by its wavelength, whereas frequency-division multiplexing typically applies to a radio carrier which is more often described by frequency. This is purely conventional because wavelength and frequency communicate the same information (Wikipedia, 2018f).

Figure 2.3 illustrates a WDM system, where a multiplexer at the transmitter to join the several signals together, and a demultiplexer at the receiver to split them apart.

2.1.5 How fast can light travel inside the glass?

The speed of light is normally 186,000 miles per second (or 300,000 km per second) in a vacuum, but this slows to about two thirds of this speed in a fibre-optic cable (Woodford, 2018).

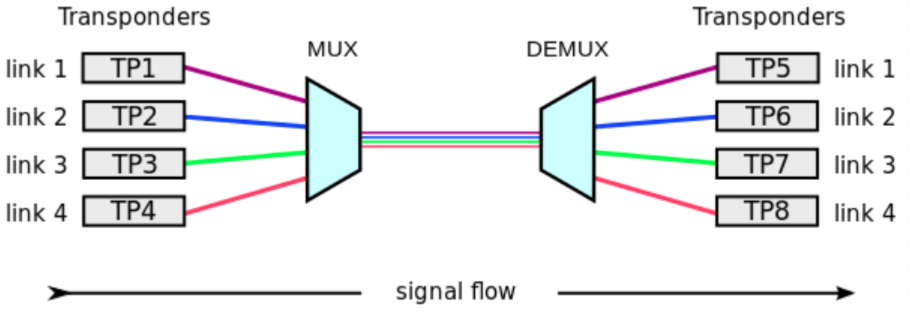


Figure 2.3: A wavelength-division multiplexing (WDM) system (Wikipedia, 2018f). MUX denotes a multiplexer at the transmitter and DEMUX denotes a demultiplexer at the receiver.

2.1.6 How to avoid light leak?

Light travels down a fibre-optic cable by bouncing repeatedly off the walls. If light hits glass at a shallow angle (less than 42 degrees), it reflects back in again - as though the glass were really a mirror. This phenomenon is called *total internal reflection*, see below. The other thing that keeps light in the pipe is the interior structure of the cable, which is made up of two separate parts. The middle of the cable is called the core and is where the light travels. Wrapped around the core is another layer of glass called the cladding, see Figure 2.2. The cladding's job is to keep the light signals inside the core. It can do this because it is made of a different type of glass to the core and has a lower refractive index (Woodford, 2018).

Total internal reflection

Waves going from a dense medium to a less dense medium speed up at the boundary. This causes light rays to bend when they pass from glass to air at an angle other than along the normal to the surface (the line at 90° to the surface). This is called refraction. Beyond a certain angle from the normal, called the critical angle, all the waves reflect back into the glass, they are totally internally reflected. The critical angle for most glass is about 42° (BBC, 2014b). An optical fibre is a thin rod of high-quality glass. Very little light is absorbed by the glass. Light getting in at one end undergoes repeated total internal reflection, even when the fibre is bent, and emerges at the other end, as shown in Figure 2.4 (BBC, 2014a).

2.1.7 What are repeaters and how far apart are they?

This section is quoted from (Davis, n.d.), (R. Sathya, 2014) and (McCool, n.d.).

> Even though modern optical fibres are extremely transparent, attenuation of the intensity of the light traveling along the fibre still occurs, and over long distances the light signal must be boosted back to a larger value (Davis, n.d.). This process was traditionally carried out with

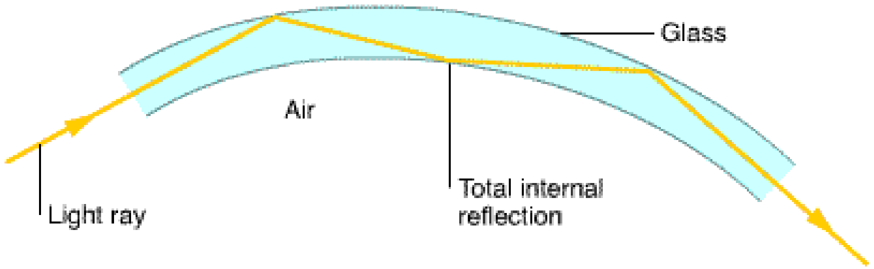


Figure 2.4: Total internal reflection (BBC, 2014a).

a repeater. This is a device incorporating a light detector, processing electronics, and a new laser. An incoming stream of optical pulses, corresponding to the transmitted information in binary code, is detected and becomes an equivalent stream of electrical pulses. These electrical pulses are amplified, reshaped electronically to restore their original shape, and are then used to drive the new laser to re-transmit the information along the next stretch of fibre. In this way, a stream of optical pulses can be transmitted over great distances, such as under the Atlantic Ocean, by spacing a series of repeaters along the fibre cable. Typically repeaters are spaced every 45-70 km. Consequently, a long fibre cable must incorporate conventional electrical wires to provide the power to drive the repeaters (R. Sathya, 2014).

Erbium doped fibre amplifier (EDFA)

Otherwise known as a fibre or optical amplifier, the EDFA is an important component in long distance fibre links. Fibre and component attenuation in modern telecommunications links degrade the transmitted signal. When the signal power becomes too low, errors will occur at the optical receiver as it struggles to recognize the transmitted signal from received noise. Before the introduction of EFDAs, in order to transmit signals over long distances the signal would be detected and re-transmitted at regular intervals, and this process was called regeneration. EDFAs provide the telecommunications engineer with the means to optically amplify the signal en-route without converting the signal from the optical back to the electrical domain. The component works by the principle of stimulated emission. A piece of fibre doped with Erbium irons is pumped by a laser at high powers. The excited erbium irons release their energy when the data signal is passed through the fibre. The process is such, that the energy they release matches the signal exactly, thus amplifying the signal (McCool, n.d.).

2.1.8 Coding and modulation

Coding and modulation are two different things. Coding is about how to convert data from one format to another. For example, the decimal number ”3” can be encoded as a binary number ”11”.

Modulation is a way to convert data into waves for transmission. The digital version of modulation is called keying [1]. The following sub-sections will give more details.

2.1.8.1 Coding

In this section, two types of coding, which are Non-Return-to-Zero (NRZ) and Return-to-Zero (RZ), are introduced.

Non-Return-to-Zero (NRZ) coding

If the transmitted bit stream is simply the presence or absence of light on the fibre (or the changes of voltage on a wire), then the simplest coding possible is NRZ. This is illustrated in Figure 2.5. Here a one bit is represented as the presence of light and a zero bit is represented as the absence of light. This method is used for some very slow speed optical links. (Dutton, 1998).

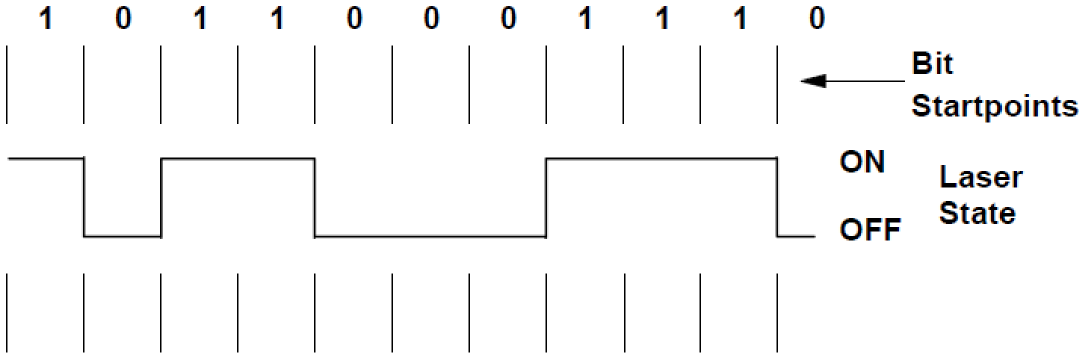


Figure 2.5: Non-Return to Zero (NRZ) coding. (Dutton, 1998)

Return-to-Zero (RZ) coding

Using RZ the signal returns to the zero state during every bit time. As illustrated in Figure 2.6, a "1" bit is represented by a "ON" laser state for only half a bit time. This is not a good coding system in a restricted bandwidth environment because there are two different line states required to represent a "1" bit. In some environments, however, bandwidth is not a major constraint, for example in the optical fibre environment or in the electronic environment using shielded cables (Dutton, 1998).

2.1.8.2 What is the frequency of the carrier wave?

Before I introduce what modulation is, I need to explain what a carrier wave is.

A carrier wave, carrier signal, or just carrier, is usually a sinusoidal waveform that is modulated (modified) with an input signal for the purpose of conveying information. This carrier wave usually has a much higher frequency than the input signal. The carrier is usually either to transmit the

[1] https://www.cse.wustl.edu/ jain/cse574-14/ftp/j$_0$$3phy.pdf$

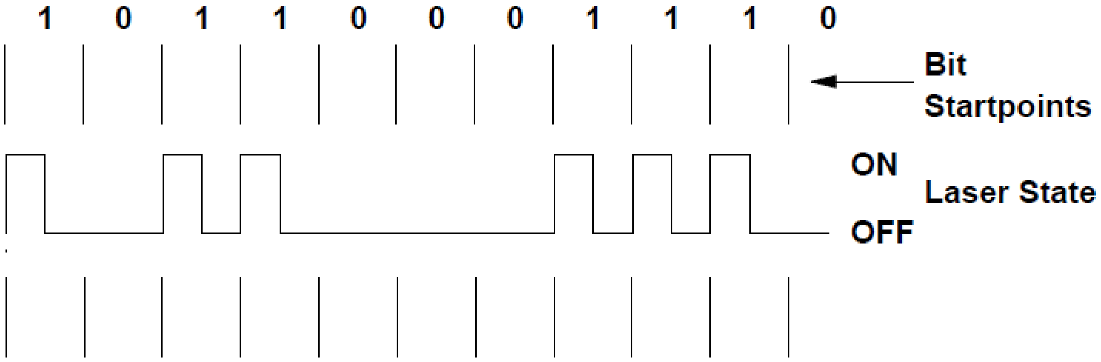


Figure 2.6: Return to Zero (RZ) coding. (Dutton, 1998)

information as an electromagnetic wave or pulse (for example, as in radio communication), or to allow several carriers at different frequencies to share a common physical transmission medium by frequency division multiplexing (as, for example, a cable television system)(Wikipedia, 2018b). With laser transmission over an optical fibre media, a carrier can also be a laser-generated light beam on which information is imposed (Rouse, 2007).

The primary reason that optical fibres have very much larger information-carrying capacity than other media, is that they carry light. The frequency of the light beams that travel along optical fibres is in the vicinity of two hundred trillion cycles per second (Hz) and the frequencies that must be transmitted for voice communications covers the range (bandwidth) from about 50Hz to 20,000Hz (20kHz). Since there is very little need to include high frequencies for understandable voice communications, the actual bandwidth needed is really only about 4 kHz. Hence it is possible, in principle, to carry about 50 billion voice conversations on a single laser beam in an optical fibre, (R. Sathya, 2014).

2.1.8.3 Modulation

A message signal containing information is used to control parameters of a carrier signal, in other words, *the information is embedded onto the carrier. The carrier could either be a 'sinusoidal wave' or a 'pulse train'* [2].

In wired electronics digital signals take (ideally) one of two discrete states. When modulation techniques are used for digital communication, the variations applied to the carrier are restricted according to the discrete information being transmitted. Examples of common digital modulation types are ASK (Amplitude Shift Keying), FSK (Frequency Shift Keying) and PSK (Phase shift Keying). These schemes cause the carrier to assume one of two possible states depending on whether

[2] https://pdfs.semanticscholar.org/presentation/4f3c/530fb2cc239eef62d269737fecca55d93cc6.pdf

the system must transmit a binary 1 or a binary 0; each discrete carrier state is referred to as a symbol (Keim, 2016).

Figure 2.7 shows an example of each modulation type. The top panel shows the desired digital signal (information); the second panel shows the carrier wave. The bottom three panels show the carrier wave after being modulated using two different amplitudes (ASK), two different frequencies (FSK), and two different phases (PSK) for ’0’ and ’1’, respectively (Keim, 2016).

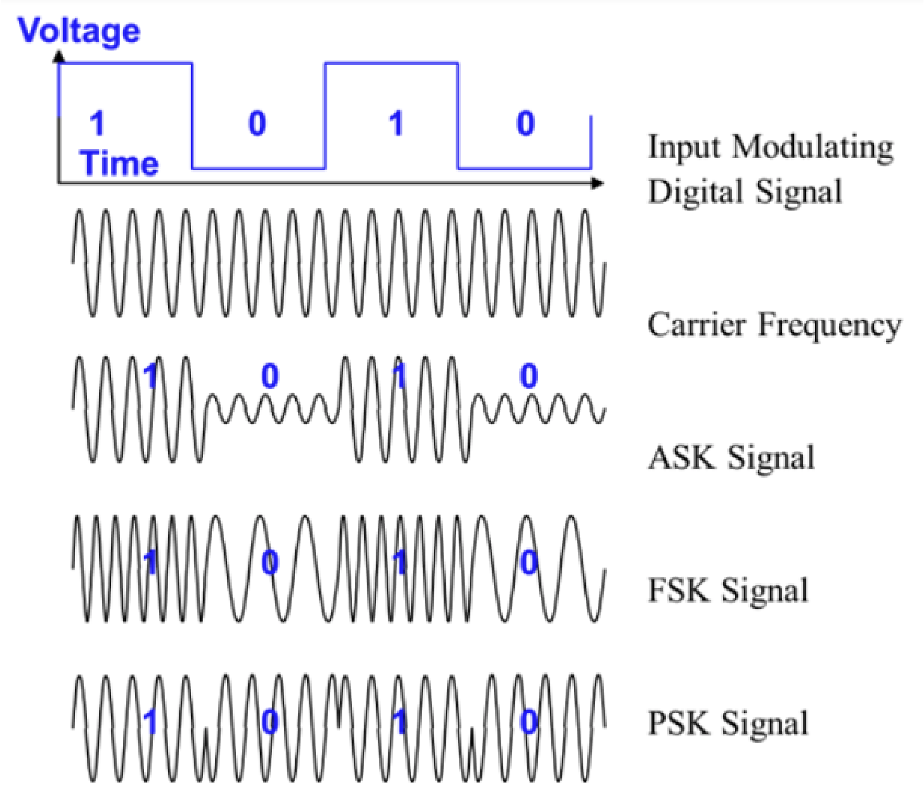


Figure 2.7: Digital Modulation. The top panel shows the desired signal, the second panel shows the carrier wave, the next three panels show an example of Amplitude Shift Keying, Frequency Shift Keying and Phase shift Keying(Faruque, 2017).

Figure 2.8 shows a further illustration regarding amplitude modulation. It shows how the amplitude modulated (AM) wave is interconnected with an imaginary line (the envelope). Amplitude modulation is where the height of the carrier signal is changed in accordance with the height of the message signal. In amplitude modulation, only the amplitude of the carrier wave is changed while the frequency and phase of the carrier wave remain constant. The first panel in Figure 2.8 shows the modulating signal or message signal which contains information, the second panel shows the high frequency carrier signal which contains no information and the last panel shows the resultant amplitude modulated signal. The third panel shows that the amplitude of both the positive and negative half cycles of the carrier wave is varied in accordance with the instant amplitude of the message signal. It can be observed that the positive and negative peaks of the amplitude modulated (AM) wave are interconnected with an imaginary line. This imaginary line on the AM wave is called an envelope (Shaik, n.d.).

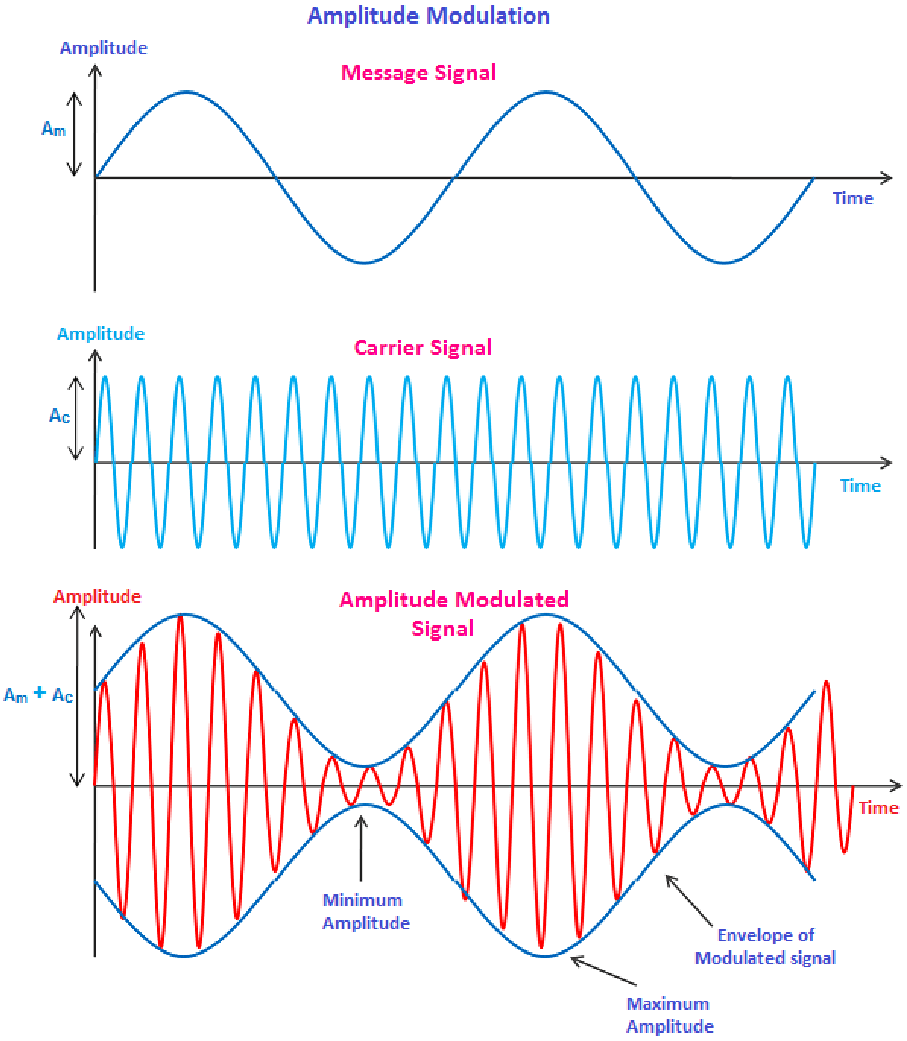


Figure 2.8: An amplitude modulated (AM) wave interconnected with an imaginary line (the envelope), (Shaik, n.d.).

I/Q modulation

If you take two equal amplitude signals that are out of phase by 90°, such as a sine and a cosine, and add them together the resultant signal is of phase 45°. This can be seen by using the standard trig identity in Eq. 2.1:

$$\sin(A) + \sin(B) = 2\sin(\frac{A+B}{2})\cos(\frac{A-B}{2}), \tag{2.1}$$

So that, Eq. 2.2:

$$\begin{aligned} I + Q &= \cos(x) \; + \; \sin(x), \\ &= \sin(x + 90) \; + \; \sin(x), \\ &= 2 \; \sin(x + 45) \; \cos(45), \\ &= \sqrt{2} \; \sin(x + 45) \end{aligned} \tag{2.2}$$

Such a system is referred to as I/Q modulation. The term "I/Q" is an abbreviation for "in-phase" and "quadrature". By convention, the I signal ("in-phase") is a cosine waveform, and the Q signal ("quadrature") is a sine waveform. Another way to express this is that the sine and cosine waves are in quadrature (AAC, n.d.).

Quadrature Phase-Shift Keying

The term "quadrature modulation" refers to modulation that is based on the summation of two signals that are in quadrature. In other words, it is an I/Q-signal-based modulation. Quadrature phase shift keying (QPSK) is an example of quadrature modulation and is particularly interesting because it actually transmits two bits per symbol. In other words, a QPSK symbol does not represent 0 or 1 - it represents “00 ”, “01 ”, “10 ”, or “11 ”. Therefore its bandwidth efficiency is (ideally) higher by a factor of two.

If the I signal is a standard cosine and the Q is a standard sine then we get a phase difference of 45° when we add the two symbols as shown above.

However, if we invert either or both of the sinusoidal signals we get a set of 4 different phases when we add the two signals together. See Table 2.1. For instance, an inverted sine wave is equivalent to a sine wave of an angle with a phase difference of 180° from the non-inverted sine wave.

$$- \; \sin(x) \; = \; \sin(x + 180) \tag{2.3}$$

A similar calculation to the one shown above (Eq. 2.3) then gives all 4 results.

Summing I and Q signals that are either inverted or non-inverted are easily generated using I/Q modulation techniques. Note that these 4 phase states are evenly separated around a circle to maximise the possibility of a clean detection.

We can use these 4 different phases to represent 4 different codings of two consecutive bits, see Figure 2.9. The state of the I signal, either normal or inverted, codes, say, for the first of the two bits, and the state of the Q signal codes for the second of the two bits, so that the combination can then code for the 4 different combinations. Note also that the phase-shift-to-digital-data correspondence shown above (Figure 2.9) is a logical though arbitrary choice; as long as the transmitter and receiver agree to interpret phase shifts in the same way, different correspondence schemes can be used (Keim, 2016).

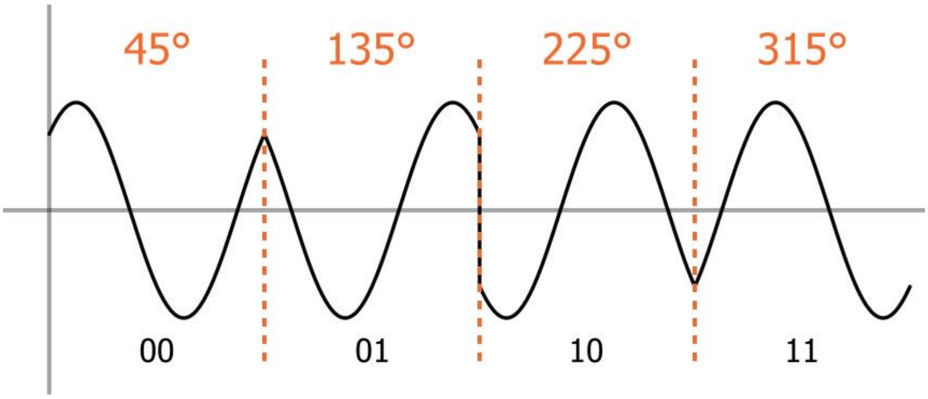


Figure 2.9: Four QPSK phase shifts are 45°, 135°, 225°, and 315° (Keim, 2016). Note: this figure illustrates the principal idea, but it does not show the carrier signal. In practice, we need a high frequency carrier signal to send this signal (baseband signal) that contains the information, look at the example shown in Figure 2.8.

I	Q	Phase shift of I+Q
non-inverted	non-inverted	45°
non-inverted	inverted	135°
inverted	inverted	225°
inverted	non-inverted	315°

Table 2.1: I/Q modulation techniques.

2.2 Literature Review

The second part of this chapter gives an overview of some published research papers that are most relevant to my research. According to the topic of each paper under investigated, I have grouped them into the following sub-sections:

1. General applications of signal processing using machine learning methods, Section 2.2.1.
2. Applications associated with the optical communication systems using machine learning methods, Section 2.2.2.
3. Applications, especially focusing on classifying the optical transmission signals using machine learning methods, Section 2.2.3.
4. The implementation of Support Vector Machines(SVM) on FPGA, Section 2.2.4.

2.2.1 Applications of machine learning techniques to signal processing

There is a big trend to use machine learning techniques (ML) in various fields. Signal processing is one of those fields. ML techniques have formed a successful combination with digital signal processing (DSP) in developing some areas such as speech recognition, bio-medical applications, channel equalization, image/audio/video encoding and algorithms on the utilization of the radio spectrum, and so on (Evgeniou et al., 2008). For example:

- Electroencephalogram (EEG)
  - Hosseinifard et al. (2013) investigates improving the classification accuracy using machine learning methods (ML) when classifying depressed patients from non-depressed subjects based on their EEG signals. Authors summarized that using nonlinear features with ML methods might provide an effective method of analyzing the EEG signals.
  - In (Aaruni et al., 2015), support vector machines with wavelet kernel functions have been used to classify EEG signals into two classes: seizure free and ictal when using EEG to detect epileptic seizures. The proposed method gave an improvement on the classification accuracy.
- Audio signal processing
  - Heittola et al. (2018) has applied deep neural networks (DNN) for analyzing the audio signals for sound scenes and events such as detecting a baby crying. The study shows the techniques can be adapted for different sound classification and detection issues.
  - Dhanalakshmi et al. (2009) proposed an algorithm to classify audio signals in broadcasting. A support vector machine classifier (SVM) was used to classify audio clips into six categories: news, music, advertisement, sport, movie and cartoon. Then, the experiments were extended by using a radial basis neural networks (RBFNN) for classifying the audio further.
  - In (Lin et al., 2005), authors have used both support vector machine and wavelets to classifying audio accurately. When the audio query is first given, the application of wavelets takes place first to extract acoustical features like pitch information and subband power. Then, the SVM is used to do the classification.
- Image processing
  - Regniers et al. (2016) proposed a complete strategy to apply wavelet-based multivariate models in a supervised classification procedure of textures in very high resolution panchromatic data. This strategy relies on a learning a texture database made of texture patches and on the preparation of the image that is to be classified. A classifier based on a similarity measure or a likelihood criterion was next used to produce classification results. The applicability of the proposed strategy was tested in two distinct contexts. In both applications, the use of the proposed strategy has achieved satisfactory classification

  results with at least one of the tested multivariate models displaying higher classification accuracy than the standard texture analysis methods.

  - Anantrasirichai et al. (2013) describes a method for automated texture classification for glaucoma detection using high-resolution retinal Optical Coherence Tomography (OCT). OCT is a non-invasive technique that produces cross-sectional imagery of ocular tissue. The proposed method relies on a support vector machine (SVM) where principal component analysis (PCA) is also used to ensure the performance of classification is improved. These researchers use a dual tree complex wavelet transform (DT-CWT) to extract the texture features.

- The area of communication systems

  - In (Fehske et al., 2005) the authors proposed a method for classification of communication signals based on cyclic spectral analysis and pattern recognition performed by a neural network. The neural network was used to avoid the problems of unknown bandwidths and uncertain carrier of the signals. The researcher confirmed that the neural network method was a robust technique for pattern recognition. In addition, using a neural network for classification constitutes a highly flexible method since the network can be retrained easily in order to incorporate new signal types.

  - Maliuk et al. (2010) has investigated the use of hardware based neural networks for making a built-in analog self-test for analog circuits. The role of this neural classifier is to map a set of simple on-chip measurements to a single-bit decision, which indicates whether the performances of the circuit complies to the specifications or not. The network can be configured into any particular topology of one-hidden-layer network under the restrictions of various inputs and neurons. The storage on digital devices provides a very fast time for computation and in this case also very low power has been consumed. Additionally, a fast training cycle has been obtained. The training algorithm used was annealing-based parallel weight perturbation. Its efficiency in recognition of separating the nominal from faulty circuits has been investigated. Software neural networks have been compared with these results. The experiment illustrates that the hardware classifier achieves similar performance compared to an ideal software classifier and is capable of learning to discriminate good from faulty circuits.

2.2.2 Applications associated with the optical communication systems using machine learning methods

Thrane et al. (2017) suggests that machine learning techniques should be of use for dealing with challenges associated with the optical communication, due to the fact that machine learning methods have been applied to address many tasks of a nonlinear nature and the optical fibre channel characteristics are nonlinear.

In (Thrane et al., 2017), the authors provide an overview on the various machine learning methods and their applications in optical communication. Their investigation shows that advanced machine learning methods have been used in many areas of optical communication systems, for example:

- Blind modulation format classification. Blind modulation classification (MC) is an intermediate step between signal detection and demodulation, with application in both commercial and military communication systems (Dobre et al., 2005). Studies in (Gonzalez et al., 2010) (Borkowski et al., 2013) (Isautier et al., 2015) (Tan et al., 2014) have shown that neural networks, K-means clustering and variational Bayesian methods for mixture models, Bishop (2006) have been successfully employed for blind modulation format classification.

- Estimating optical channel parameters, such as: chromatic dispersion (CD) [3], baud rate [4] and optical signal to noise ratio[5](OSNR), using neural networks ((Wu et al., 2009) (Wu et al., 2011) and (Dong et al., 2016)).

- Performing optimum classification for symbol detection, as shown in (Tan et al., 2011) (Wang et al., 2015) (Koike-Akino et al., 2012) (Zibar et al., 2012) and (Li et al., 2013) using, for example, support vector machine (details can be found in Chapter 4).

- In addition, machine learning methods, such as, nonlinear state-space based Bayesian filtering, have been used to estimate the frequency, amplitude and phase of a noisy signal with time-varying amplitude or phase. In the paper of (Zibar et al., 2015), Bayesian filtering methods are utilized for the characterization or classification of the laser amplitude and phase noise. This research applied the principles of Bayesian filtering and overtook the outdated time-domain method in the presence of noise.

In the research of (Wu et al., 2009), applications utilize artificial neural networks (ANNs) for monitoring the performance of optical networks. A simulation is conducted with a data rate of 40 Gb/s on-off keying and systems of differential phase shift keying. Verification is also conducted simultaneously for the identification of the OSNR, CD and the accumulated fibre nonlinearity. Moreover, through the ANN approach damage has also been monitored. Finally, the ANNs have also been

[3] Chromatic dispersion is a phenomenon that is an important factor in fibre optic communications. Its effect is essentially to stretch or flatten the initially sharply-defined binary pulses of information. This degradation makes the signals (1s and 0s) more difficult to distinguish from each other at the far end of the fibre. Cited from (Miller, 2011)

[4] In telecommunication and electronics, baud is a common measure of the speed of communication over a data channel. Technically speaking, it is the unit for symbol rate or modulation rate in symbols per second or pulses per second. It is the number of distinct symbol changes (signaling events) made to the transmission medium per second in a digitally modulated signal or a line code. Cited from (Wikipedia, 2018a)

[5] Signal-to-noise ratio (abbreviated SNR or S/N) is a measure used in science and engineering that compares the level of a desired signal to the level of background noise. SNR is defined as the ratio of signal power to the noise power, often expressed in decibels. A ratio higher than 1:1 (greater than 0 dB) indicates more signal than noise. Cited from (Wikipedia, 2018e)

used for the recognition of in-phase/quadrature (I/Q) alignment and misalignment in differential quadrature phase shift keying transmitters (Wu et al., 2009).

2.2.3 Classifying optical transmission signals using machine learning methods

Classifying optical transmission signals using ANN and wavelets

Rajbhandari et al. (2009b) proposed a receiver architecture using a discrete wavelet transform (DWT) and artificial neural network (ANN) for indoor optical wireless (OW) systems (Rajbhandari et al., 2009b). ANN was used for classifying signals in the channel, and DWT was used for denoising the artificial light interference (ALI) in indoor OW system. The indoor and outdoor OW system provide the same characteristics of the optical fibre and they offer a high speed system at a low cost (Rajbhandari et al., 2009a). When it comes to indoor optic wireless communication, ANN and DWT improve the link performance in any physical channel. However, there are constraints in the way of getting a realization of the unlimited bandwidth in optical wavelengths, where ambient light interferes and induces a multipath inter-symbol interference. It has been shown that using DWT and ANN in a receiver can minimize the effect of fluorescent light interference (FLI) and inter-symbol interference (ISI) (Rajbhandari, 2010).

Classifying optical transmission signals using ANN, support vector machine (SVM) and wavelets

Sun et al. (2008), Hunt et al. (2008) and Hunt et al. (2009) analyze the bit-error ratio in adaptive electrical signal post-processing in optical communication system. The authors recognized that this is a critical problem which needs to be solved. In optical communication systems, often the data transfer rate is very high. So, a little error can make a huge difference in the actual outcome. Therefore, detection and correction must occur at high speed and has to be extremely accurate depending on the signal speed. Also, it is very common that there are multiple channels which have different characteristics and which can be changeable over time. The authors examined the effect of using the contiguous bits to the bit that is being classified, since the bit might be affected by its context. The feasibility has been provided through using simple artificial neural networks (ANN). Furthermore, various representations of signal waveforms have been studied (Sun et al., 2008) (Hunt et al., 2008) (Hunt et al., 2009).

- In (Sun et al., 2008), the data has been chosen from an electrical domain after converting to an electrical current from an optical signal. This study has especially investigated the classification performance on easy cases and hard cases, which are misclassified by the conventional threshold method, separately. Authors argued that producing a highly accurate classifier but which is sufficiently simple is a challenge but it may be possible. So, the experiment has been restricted to use data for either just the pulse's energy or the light waveform's sampled version. They have been successful in reducing the bit error ratio from 2.81% to 1.33% by using 1 bit either side of the target (that is, using neighbouring information (the target bit, and a bit either side) gives the best results with a SLN). Although this figure is still high, it may be because that there are many more easy cases than hard cases in their experiment.

- In (Hunt et al., 2009), the authors extended their work in (Sun et al., 2008). Different feature extraction methods, such as, the discrete wavelet transformation and independent component analysis have been used. Although the improvement obtained by using wavelet coefficients and independent components is minor, the number of inputs to the single layer network is much smaller than the one using the waveform of each symbol.

Research in (Sun et al., 2008), (Hunt et al., 2008) and (Hunt et al., 2009) shows that searching for a good separator using SLN (that is, training the perceptron) is difficult on a large-scale dataset. Therefore, Metaxas et al. (2013) proposed to use a linear SVM to classify the optical transmission signals, since the linear separator can be implemented easily in the hardware and it offers the high speed required of a de-modulator. A linear SVM was used instead of a neural network as a classifier for large-scale data. These authors argued that the most important feature when dealing with the domain problem is the fact that the classifier needs to be extremely fast. Since the optical channels are capable of operating at a speed of over 50GHz, a classifier should use in-built hardware if it is to be used in practice.

2.2.4 Hardware Implementation of SVM on FPGAs

SVM software implementations can provide highly accurate classification. However, any method of bit-error ratio reduction must be able to work in real time and consequently needs to be implemented in hardware. So the SVM method needs to meet the high requirements of real-time embedded applications in hardware such as fast execution and low cost of intensive computation in hardware (Afifi et al., 2015). In this section, I review existing research about implementing an SVM classifier in hardware.

Afifi et al. (2015) considers a popular re-configurable device that is a Field- Programmable Gate Array (FPGA) to accelerate the SVM model and achieving high performance computing (HPC), while also keeping the cost and the power consumption low. They conclude that the FPGA is superior to the other main contender for hardware implementation, namely the use of a Graphics Processing Unit (GPU).

As is explained on Wikipedia (Wikipedia, 2017), a field-programmable gate array (FPGA) is an integrated circuit designed to be configured by a customer or a designer after manufacturing - hence “field-programmable”. The FPGA configuration is generally specified using a hardware description language, similar to that used for an application-specific integrated circuit. FPGAs contain an array of programmable logic blocks, and a hierarchy of re-configurable interconnects that allow the blocks to be "wired together", like many logic gates that can be inter-wired in different configurations. Logic blocks can be configured to perform complex combinational functions, or merely simple logic gates like AND and XOR. In most FPGAs, logic blocks also include memory elements, which may be simple flip-flops or more complete blocks of memory (Wikipedia, 2017).

Lots of researchers Groleat et al. (2012), Cao et al. (2010), Bustio-Martínez et al. (2010), Cutajar et al. (2013) and Komorkiewicz et al. (2012) confirm that the most difficult thing in the implementation of classifier using an SVM is the training and classification stages and the computations of a complex kernel. However, since the training can be done off line, using the copious amounts of simulated training data that exists, this part of the process is not necessary to do in real time in hardware. Hence most of the existing studies discuss the implementation of the SVM classification stage using an FPGA after executing the training stage using software (Afifi et al., 2015). The studies examine different hardware implementations of SVM classifier that have been done on an FPGA using various hardware mechanisms.

Moreover, some studies (for example, (Koide et al., 2014), (Shigemi et al., 2013) and (Peng et al., 2014)) have investigated achieving the balance between accurate classification and reaching the requirements of the real-time embedded applications in hardware, which is the main aim in most of the existing research. These requirements include high performance, the capacity to change the scale of the computing process, the ability to undertake modification easily, and keeping the consumption of cost and power low. The majority of the classification systems using SVM described previously cannot be developed and adjusted from other applications. Consequently, this is outlined as one of previous implementations' limitations, but is of no real concern to us since a dedicated FPGA would be required. In addition, using parallel processing to speed up the SVM, and the big memory to reduce the time consumption for the large-scale problems are not effectively addressed. Besides that lots of the developed designs were improved without consideration of meeting some of the embedded applications requirements; and were performed on old versions of FPGAs.

Interestingly, there were few papers investigating the use of a non-linear SVM classifier with a more complex kernel in hardware when I started my PhD study about five years ago. In (Ruiz-Llata et al., 2010), the authors proposed a hardware implementation based on FPGAs, that can do both support SVM classification and SVM regression. They presented a hardware-friendly kernel function that can simplify the SVM computation in a constrained hardware while maintaining good classification performance with respect to the widely used Gaussian kernel (see details in Chapter 4). However, in their system the parameters of the hardware-friendly kernel have to be fixed prior the training step. Consequently, my research has mainly focused on using a linear classifier, since I was confident that it can be implemented in hardware. Later on, Afifi et al. (2015) proposed hardware implementation of a non-linear SVM using FPGA technology. Therefore, I have updated my experiments, and results have been added in some cases, for a non-linear classifier in case that eventually it may become easily achievable in hardware (Afifi et al., 2015).

Recently, some research discusses hardware implementations of deep learning neural network, which I indicate as a suitable topic for future work. Deep learning is able to solve many complex problems, which makes the challenge of building a high performance hardware of deep learning neural networks (DLAU) worth achieving (Wang et al., 2017). This study (Wang et al., 2017) proposed a deep learning accelerator unit using a field-programmable gate array (FPGA). DLAU can deal excellently with the large-scale neural networks. DLAU also can attain a 36.1**x** speedup with suitable hardware costs and using low power. In (Wang et al., 2018), the authors discussed hardware implementation of deep convolutional Neural Network (DCNN) models on FPGAs. They conclude that the design outperformed the current ones successfully.

2.2.5 Conclusion

In this Chapter I have introduced some background information on the basic knowledge needed in the area of the transmission of optical data, and reviewed some of the published papers that were the most relevant to my research area. To summarise the literature review: the most common types of machine learning techniques, such as artificial neural networks and support vector machine (SVM) have been used in applications of optical communication systems. The wavelet transform based noise reduction methods have also been widely applied in signal processing related work. The authors of (Sun et al., 2008), (Hunt et al., 2008) and (Hunt et al., 2010) have shown that using an SLN can improve the bit errors, and the best result was when using some neighbouring information. Since training the perceptron to find a good separator is complicated with a large data set as mentioned in (Metaxas et al., 2013), I have focused on using the SVM, which is done by a linear kernel, because the linear separator is easier to be build in hardware at the desire high speed required by the decoder. Also, I pay particular attention to using the neighbouring information in my study. Finally, note that my work is to reduce the bit errors prior to any error correction being used.

Chapter 3

Data and Data Pre-processing

This chapter is in two parts: a description of data and an introduction to data pre-processing which I have used in my work.

In this research, there are three types of optical transmission data, and two types of sinusoidal signals that I have used. All three types of optical transmission data are simulated employing the most up to date methods as used by practitioners in the field; they are provided by one of my supervisors Dr.Alexey Redyuk from the Institute of Computational Technologies in Novosibirsk, Russia. Details on how these data are simulated can be found in Section 3.1.

In addition, to better understand how effective the wavelet transformation (see section 3.3.3) of signals on the phase or frequency distortion in the optical transmission data, I have generated sinusoidal signals in this study. In this way, the signal change over frequency, amplitude and phase can be easily observed. In this study, I have focused on the signal change over frequency with amplitude and phase with amplitude, that is I have generated two different types of sinusoidal signals. Details can be found in Section 3.2.

Section 3.3 discusses data representations and data pre-processing.

3.1 Optical Transmission Signals

This section describes the optical transmission data used in this study. While the detailed mathematical description is shown in Appendix A.1, I shall briefly explain, in Subsection 3.1.1, the three most important equations, which are used to generate the data used in this thesis. This data represents the most realistic simulated data, which can better reflects the character of the real transmission line with the real BER that would be obtained with real world transmitted data (Nasreen et al., 2018), (Agrawal, 2018) and (Antoniades et al., 2011).

3.1.1 The mathematical model of the simulated optical transmission data

The evolution of the complex field envelope $A(z,t)$ in a fibre-optic link can be represented by the stochastic general nonlinear Shrödinger equation (GNLSE) (Agrawal, 1997):

$$\frac{\partial A}{\partial z} + \frac{\alpha}{2}A + \frac{i}{2}\beta_2\frac{\partial^2 A}{\partial^2 t} - i\gamma|A|^2 A = A\sum_{k=1}^{k=S} G\delta(z-kL) + \sum_{k=1}^{k=S} N\delta(z-kL), \tag{3.1}$$

where $A(z,t)$ is the complex field envelope, z is the distance along the fibre, t is the time, α is the fibre loss, β_2 is the dispersive term, γ is the nonlinear term, G is the gain coefficient of erbium doped fibre amplifiers (EDFA), S is the number of spans, L is the length of each span. The term $N(z,t)$ is the one describing amplified spontaneous emission (ASE) noise generation. ASE can be represented by the field that has the statistical properties of additive Gaussian noise. Without the noise term, equation (3.1) can be numerically solved using the split-step Fourier method (SSFM) (Essiambre et al., 2010), which requires the division of the fibre into small steps. The details of SSFM can be found in Appendix A.1.

The complex field envelope at the position $z + \Delta z$ can be approximated by:

$$A(z+\Delta z,t) \approx F^{-1}\left\{\exp\left[\frac{\Delta z}{2}\hat{D}\right] F\left\{\exp\left[\Delta z\hat{N}\right] F^{-1}\left\{\exp\left[\frac{\Delta z}{2}\hat{D}\right] F\{A(z,t)\}\right\}\right\}\right\}, \tag{3.2}$$

where $F\{\}$ and $F^{-1}\{\}$ denotes the forward and backward Fourier-transform operation respectively; $\hat{D}$ and $\hat{N}$ are the linear and nonlinear operators, separately. With the noise term, we simply add the noise discretely to the field. Therefore, The complex field envelope at at the position $z + \Delta z$ can be approximated by:

$$A(z_k+\Delta z,t) \approx F^{-1}\left\{\exp\left[\frac{\Delta z}{2}\hat{D}\right] F\left\{\exp\left[\Delta z\hat{N}\right] F^{-1}\left\{\exp\left[\frac{\Delta z}{2}\hat{D}\right] F\{A(z_k,t)\}\right\}\right\}\right\} + n(z_k,t). \tag{3.3}$$

Using the numerical model described above, the typical return-to-zero and non-return-to-zero transmitters can be simulated.

3.1.2 Description of the optical transmission data

In this study, three types of optical transmission data including two non-return-to-zero and one return-to-zero have been generated. Each simulated pulse is represented by 64 equally spaced phase samples, and is decoded into one of four symbols. For each simulation, each symbol has a corresponding two-bit label, which are "00", "01", "11" and "10", respectively, as shown in see Figure 3.1. The initial encoding of the phase of the signal should be 0, $\frac{\pi}{2}$, $\pi(or, -\pi)$ or $-\frac{\pi}{2}$, if the symbol represents "00", "01", "11" or "10", respectively.

The first type of data is initial data, which is used in order to determine if any of the ideas I have for improving the Bit Error Ratio (BER) are worth pursuing further; the second one is the main data used in this study, on which a set of comprehensive experiments have been undertaken;

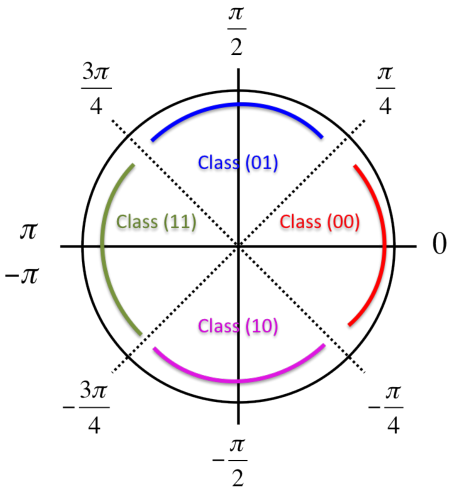


Figure 3.1: Four different variants of the light after QPSK modulation. It can be seen the four classes that my optical transmission data types include. The discontinuity of the phase appears in the area between π and $-\pi$. Note that the initial encoding of the phase of the signal could be 0, $\frac{\pi}{2}$, $\pi(or, -\pi)$ or $-\frac{\pi}{2}$, if the symbol represents "00", "01", "11" or "10", respectively.

the third type is constructed in order to test the effect of having meaningful text in the data. The best methods determined on the second type have been applied on the third type of data at the maximum distance only.

Table 3.1 presents a summary of description of these data. From the table, it can be seen that:

- The main differences between the first type of data and the last two types of data include the channel bit rate, number of channels and format of modulation.
- The only difference between the second type and the third type is that the second one is sets of random signals, while the third type is generated based on a meaningful text.

Data	Optical fibre	Modulation	Polarization	Spectral channel	Random	Bit rate
$1^{st}type$	Single Mode Fibre (SMF)	RZ-QPSK	One polarization	One spectral channel	Random signals	80Gbit/s
$2^{nd}type$		NRZ-QPSK	Dual polarization (DP)	Multi-channel transmission (3 spectral channel)	Random signals	120Gbit/s per channel
$3^{rd}type$					Meaningful text	120Gbit/s per channel

Table 3.1: A summary of description of the optical transmission data types that are used in this research. The difference between the first type of data, and both the second and third type is the signal properties: channel bit rate, number of channels and format of modulation. The only difference between the second type and the third type is that the third type is generated based on meaningful text. RZ and NRZ denote to the modulation type that is Return-to-Zero and Non-Return-to-Zero, respectively. QPSK denotes to the Quadrature Phase Shift Keying (see Figure 3.1).

Next, I shall give more details on each type of data in the following three sub-points 3.1.2.1, 3.1.2.2 and 3.1.2.3. Further details of each type of data such as the number of symbols are described in each chapter of results, see Chapter 5, 6 and 8.

3.1.2.1 The first type of data

This type of data has been simulated as a typical Return-to-Zero (RZ)-QPSK (Note: QPSK is Quadrature Phase Shift Keying) transmitter (see Chapter 2). It is generated at the initial state, that is prior to transmission, and at the distance of 3,000 km that the signal has travelled. (Note that the initial state is the state that is before the data is transmitted through the optical link, in other words, it is after modulating the digital bits directly.) The simulation has been repeated 50 times (data-sets/runs) with different random realizations of Amplified Spontaneous Emission (ASE) noise and input Pseudorandom Binary Sequence (PRBS).

Figures 3.2 shows the amplitude of a signal including ten consecutive symbols. As one can see, the amplitude in the initial state is identical over ten symbols due to the use of the return-to-zero modulation. Figure 3.3 shows the phase of the same signal. Two green circles in the figure present the angle's discontinuity between π and $-\pi$. The phase value of these two points on the top and bottom are actually next to each other. Note that only the first pair of discontinuous points is circled as an example (see Figure 3.1). For more details about the first type of data, see Appendix A.2.

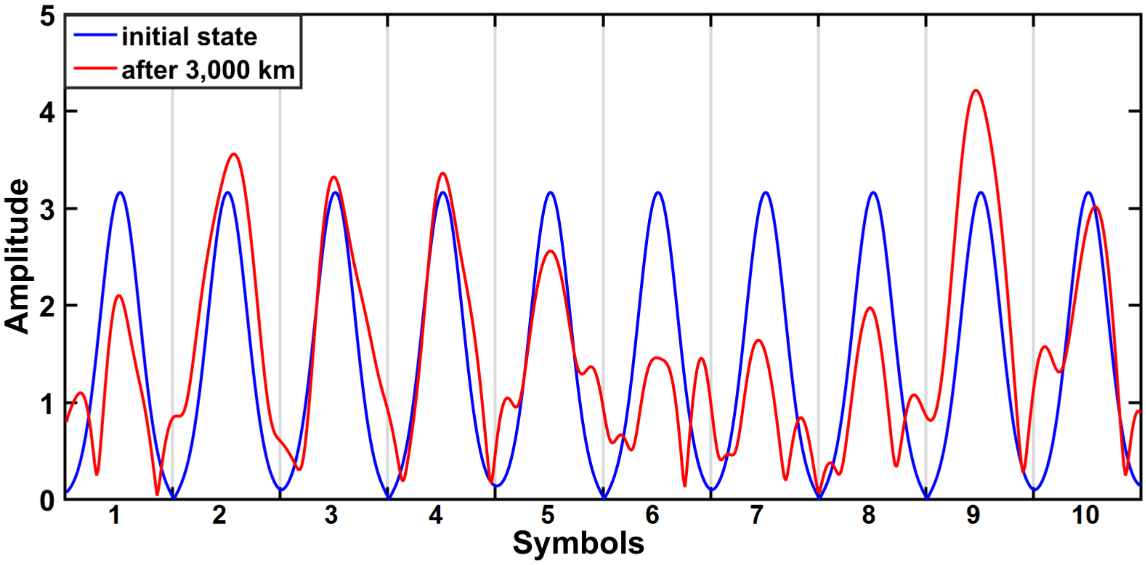


Figure 3.2: The amplitude of a part of an optical transmission signal that contains ten symbols. This signal is obtained from the first type of data, first data set, at the initial state and after a distance of 3,000 km. The amplitude in the initial state is identical over ten symbols due to the use of RZ modulation.

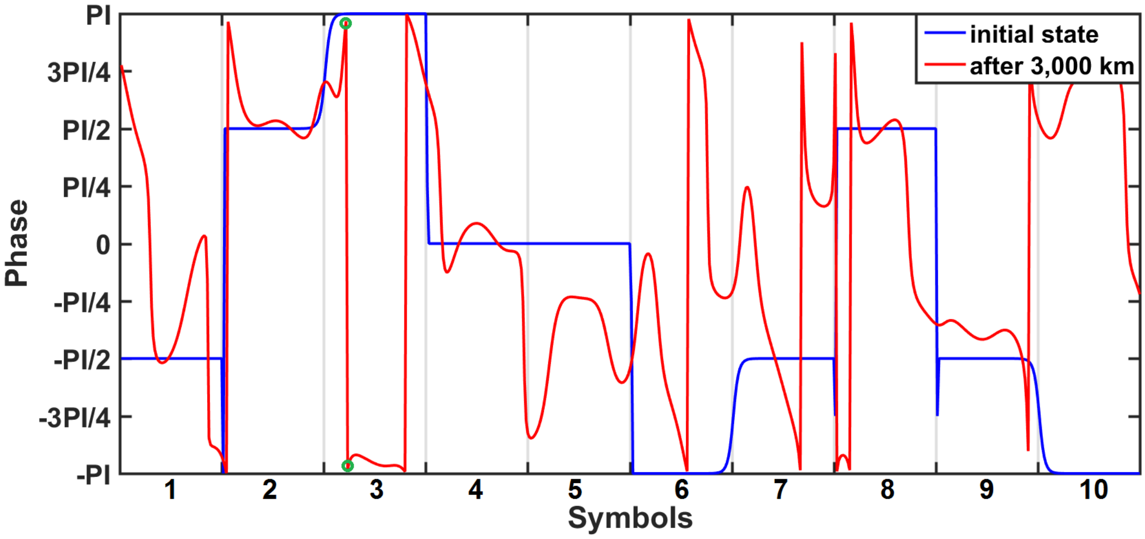


Figure 3.3: The phase of a part of an optical transmission signal that contains ten symbols. This signal is obtained from the first type of data, the first data set, at the initial state and after a distance of 3,000 km. The green circles, show an example of the angle's discontinuity between π and $-\pi$ radians, where the phase values of the two points at the top and bottom are actually next to each other (Note that only one pair of discontinuous points is circled as an example). More details can be found in section 3.3.1 (see Figure 3.1). Note that PI denotes π.

3.1.2.2 The second type of data

This type of data has been simulated as a typical non-return-to-zero (NRZ)-DP-QPSK (Dual Polarization Quadrature Phase Shift Keying) transmitter (see Chapter 2). Figure 3.4 shows an example of how a signal is modulated using QPSK. The top panel shows two carrier signals, which are in-phase and quadrature carriers as mentioned in Chapter 2, representing the first bit and the second bit of each symbol, separately. The bottom panel shows the encoded signal using the QPSK modulation type.

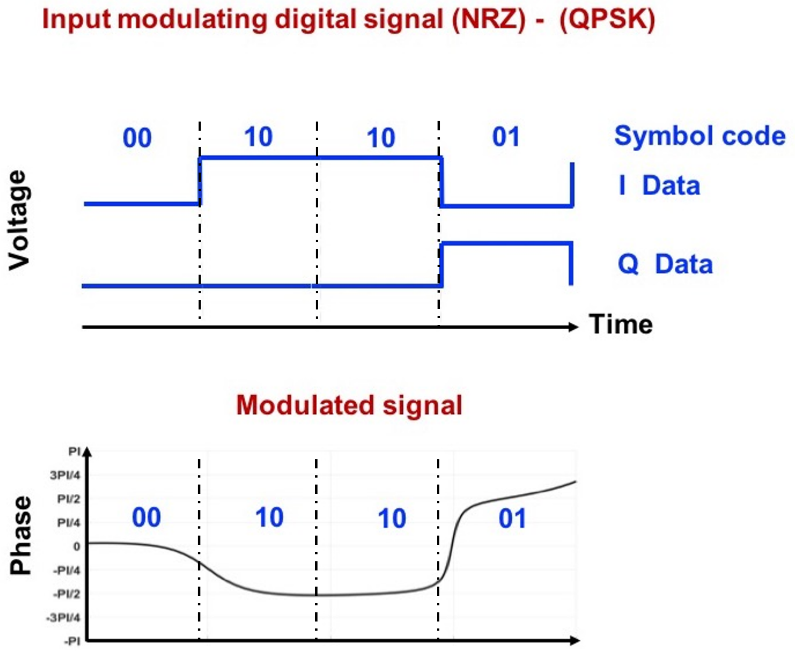


Figure 3.4: Quadrature Phase Shift Keying (QPSK) modulation. The top panel shows the digital signal (I) which represents the first bit and the second bit is represented by the (Q) data. The bottom panel shows the encoded signal using the QPSK modulation type, they are four symbols obtained from the second type of optical transmission data, the first data set. Note that PI denotes π.

The simulation has been repeated ten times with different random realizations of ASE noise and input PRBS for each different distance. The signal is detected at intervals of 1,000 km to the maximum distance of 10,000 km. This type of data has been simulated with dual polarisation (that is X- and Y- Polarizations). I have focused on X-Polarization data and used Y-Polarization data for validating my results only.

Figures 3.5 and 3.6 show the amplitude and phase of an optical signal, respectively. This signal contains ten consecutive symbols and is selected from the first data set, at the initial state and after travelling a distance of 10,000 km using X-polarization. As one can see in Figure 3.5, the amplitude in the initial state is different from symbol to symbol due to the use of the non-return-to-zero modulation type, which is different from the first type of data (see Figure 3.2).

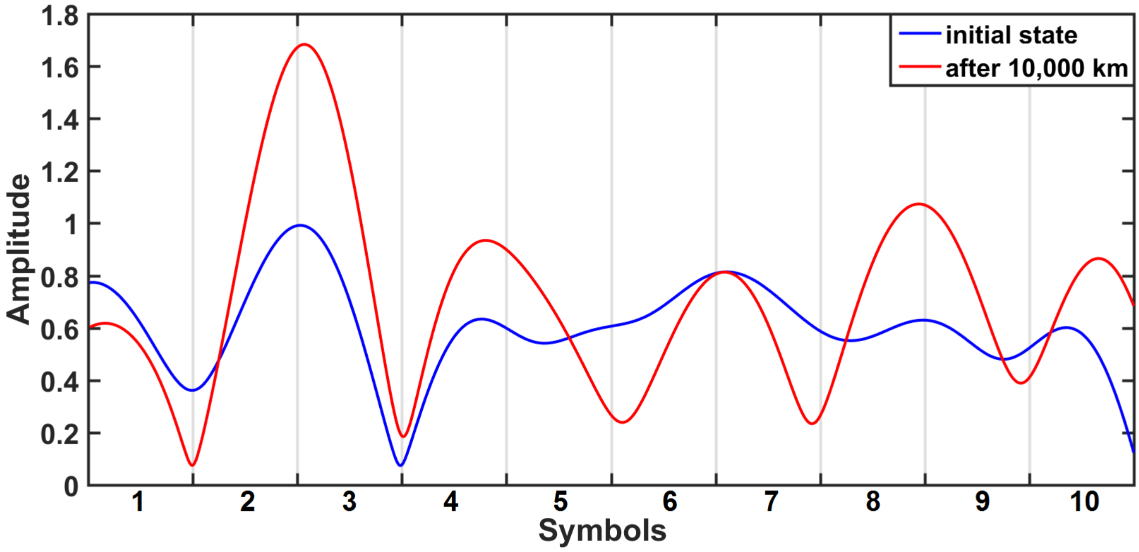


Figure 3.5: The amplitude of a part of an optical transmission signal that contains ten symbols. This signal is obtained from the second type of data (the first data set), at the initial state and after a distance of 10,000 km using X-polarization. As one can see, the amplitude in the initial state is different from one symbol to another, because of the NRZ modulation type (different from the first type of data in Figure 3.2).

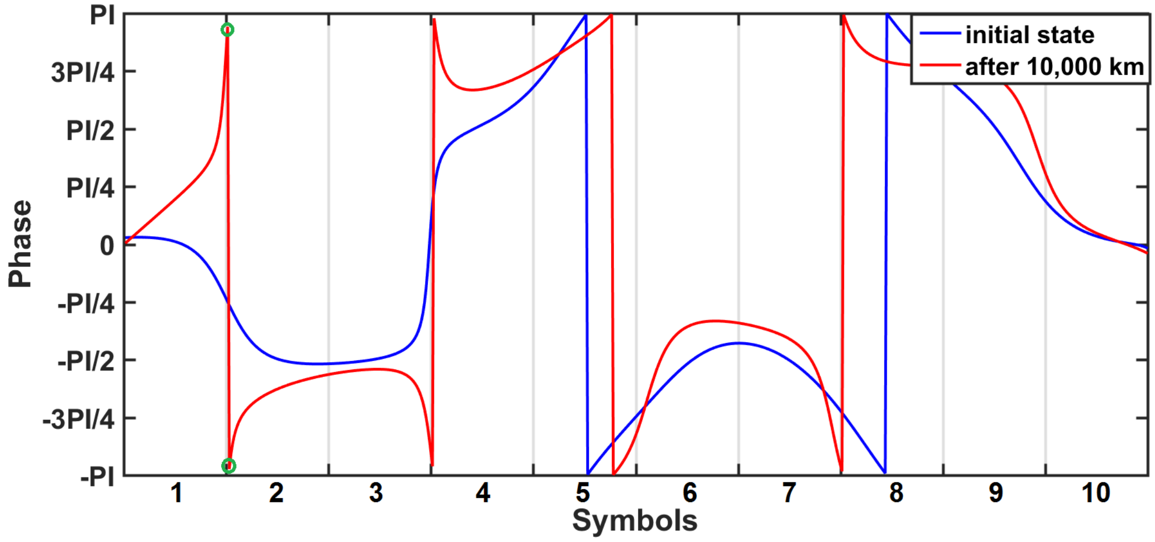


Figure 3.6: The phase of a part of an optical transmission signal that contains ten symbols. This signal was obtained from the second type of data (the first data-set), at the initial state and after a distance of 10,000 km, X-polarization. The green circles show the angle's discontinuity between π and $-\pi$ radians, where the phase values of the two points at the top and bottom are actually next to each other (Note that only the first pair of discontinuous points is circled as an example). More details can be found in section 3.3.1 (see Figure 3.1). Note that PI denotes π.

Comparing the signal in the initial state in Figure 3.6 with the one in Figure 3.3, one can see that the phase value of each symbol changes in the second type of data, while they are consistent in the first type. For example, looking at the first symbol (64 samples) in Figure 3.6, the phase value changes from a value near to zero to $-\pi/4$, while in Figure 3.3, the phase value of the first symbol is $-\pi/2$ consistently. In addition, the green circles in Figure 3.6 (Note that only the first pair of discontinuous points is circled as an example in the figure) again show the angle's discontinuity between π and $-\pi$, where the phase values of the two points on the top and bottom are actually next to each other (see Figure 3.1).

As expected, since this is random data both the symbols ("00 ", "01 ", "10 ", and "11") and the bits ($0'$s and $1'$s) are evenly distributed in each data-set. Also, the test set keeps the same distribution (see Figures A.1 and A.2 in Appendix A.3). For more details about the second type of data, see Appendix A.3.

3.1.2.3 The third type of data

This type of data has been generated using the same modulation type and fibre characteristics as used in simulating the second type of data. However, this type is generated based on a meaningful text. The text is English and includes upper case letters, lower case letters, blank spaces, new lines, some special symbols and some punctuation marks. In total, the text consists of 36,864 characters (83 unique characters), which are divided into nine data sets. Each data set in the third type of data consists of 4096 characters. The signal is detected at different distances from 1,000 km to a maximum distance of 8,000 km at intervals of 1,000 km. Since each character contains 4 symbols, that is 8 bits, there are $16,384$ symbols in each run (data set) in this simulation.

Figure 3.7 shows the frequency of some characters in the text obtained from the first data set. It can be seen that the frequency of the space and some of the lower case letters are much more than the other characters such as numbers or upper case letters. Therefore, I suppose for example if the lower case letters appeared more frequently during the training process, the classifier would be more familiar to them, and might predict them correctly more than the other patterns. For more details about the text in the first data set, see Figure A.3 and Table A.1, in Appendix A.4.

Figure 3.8 shows the percentage of symbols and the percentage of bits of the first data-set from the third type of data on the top and bottom panel, respectively. As can be seen, the number of 1's and the number of 0's are roughly balanced, while the number of each symbol is not. For example, the number of "01 " symbols is double of the number of "11"symbols. This is different from the second type of data, where data is generated randomly and has a very similar number for each symbol in each data-set (see Figures A.1 and A.2 in Appendix A.3). A randomness test (The MathWorks, 2018), which is a statistical test to check whether a data set is in random order or not, has been undertaken for both the third and second type of data. The result on the third type of data confirms that it is non-random, since the number of patterns of ones and zeros of different lengths is not the same as the one generated from a random series of bits, for example, the second type of

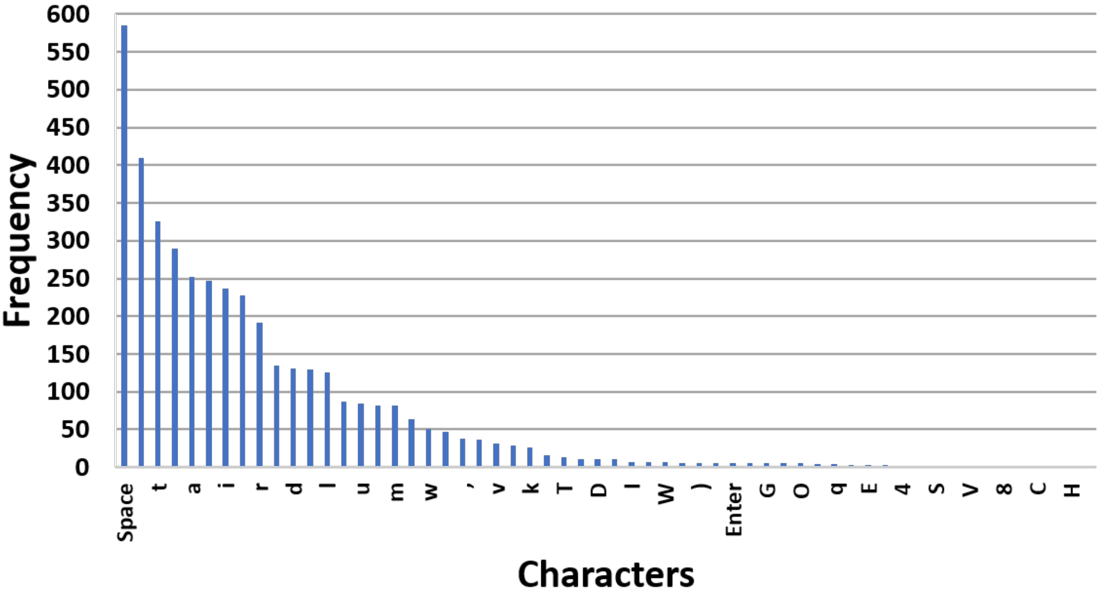


Figure 3.7: The frequency for some characters in the third type of data. It can be seen that the frequency of the space and some of the lower case letters are much more than the other characters such as numbers or upper case letters. Therefore, I suppose for example if the lower case letters appeared more frequently during the training process, the classifier would be more familiar to them, and might predict them correctly more than the other patterns. For more details, see Table A.1 in Appendix A.4.

data. Figure 3.9 shows the percentage of symbols and the percentage of bits of the test set, which has the similar distribution to the whole data-set as shown in Figure 3.8.

Figures 3.10 and 3.11 show the amplitude and phase of ten consecutive symbols, respectively. This signal is randomly selected from the first dataset, at the initial state and a distance of 8,000 km (X-polarization). Comparing with the previous figures that are obtained from the second type of data (Figures 3.5 and 3.6), it can be seen that the phase value of each symbol in the third type of data changes over the 64 samples, which is the same as shown in the second type of data. This is because both types of data are generated using the same modulation method.

3.2 Sinusoidal Signals

Optical transmission data can be distorted in its amplitude, frequency and phase during transmission. In order to fully quantify how effective my data pre-processing method (specifically, wavelet transforms, which is shown in Subsection 3.3.3) might be with the distorted data I start with analyzing how effective the method would be on simple data that has noise added to its amplitude, frequency or to its phase. Therefore, I have generated the following two types of sinusoidal signal

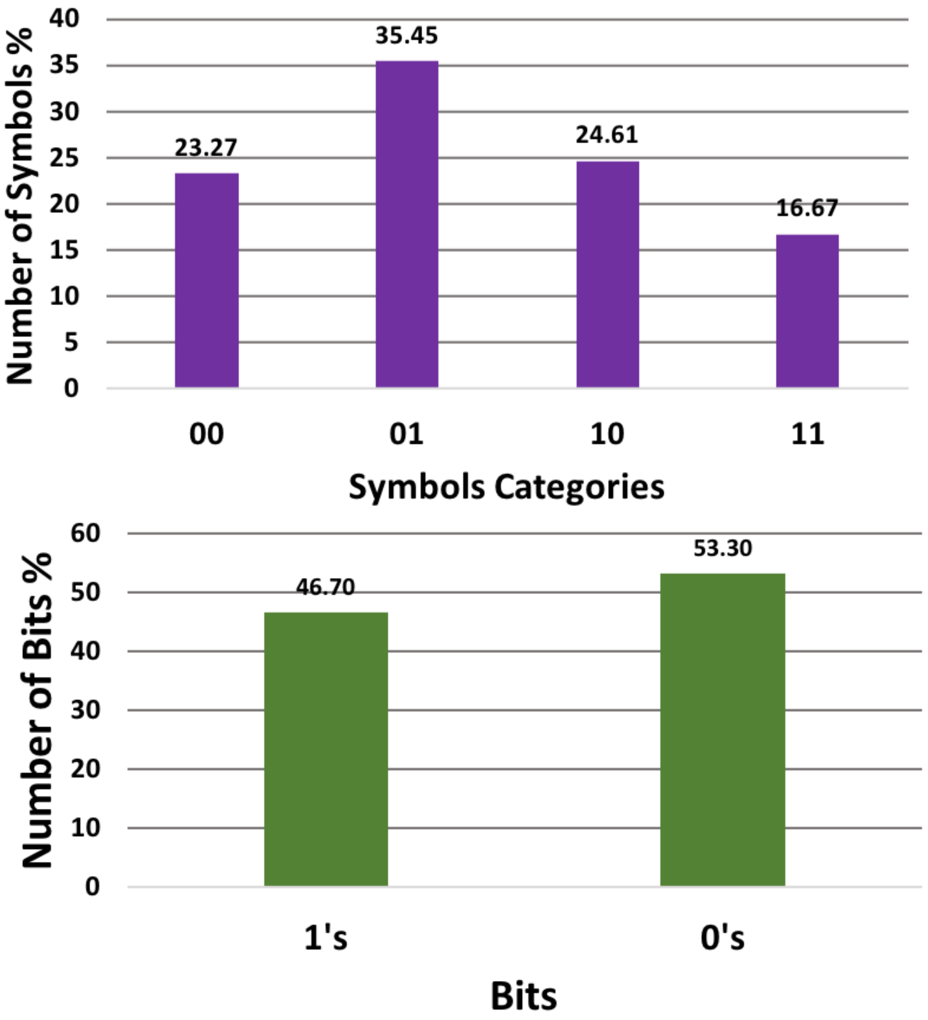


Figure 3.8: The top panel shows the percentage of symbols in each class. Note that the percentage of symbols in each class are not similar as they were in the second type of data (Figure A.1, in Appendix A.3). Class "01" has the highest percentage. The panel in the bottom shows the percentage of bits for $1'$s and $0'$s, which are quite similar to each other. The percentages are obtained from the whole first data-set of the third type of data.

shown in Sections 3.2.1 and 3.2.2, respectively.

3.2.1 Sinusoidal signals with frequency noise

Four classes A, B, C and D of sinusoidal signals have been generated with simulated noise added to its frequency. This frequency noise was added via a Gaussian distribution based on a different value of signal frequency (with a unit of (Hz)), which is 10 (A), 15 (B), 20 (C) and 12 (D), respectively. Each class of data consists of 500 signals, and each signal consists of a vector of 640 equally spaced sample values (y-values) samples. Each vector (wave) has a corresponding class label, that is either A, B, C or D, namely. The signal is defined by:

$$Signal(t) = \sin(2\pi\,\tilde{f}\,t) + \mathcal{E}_s\,, \tag{3.4}$$

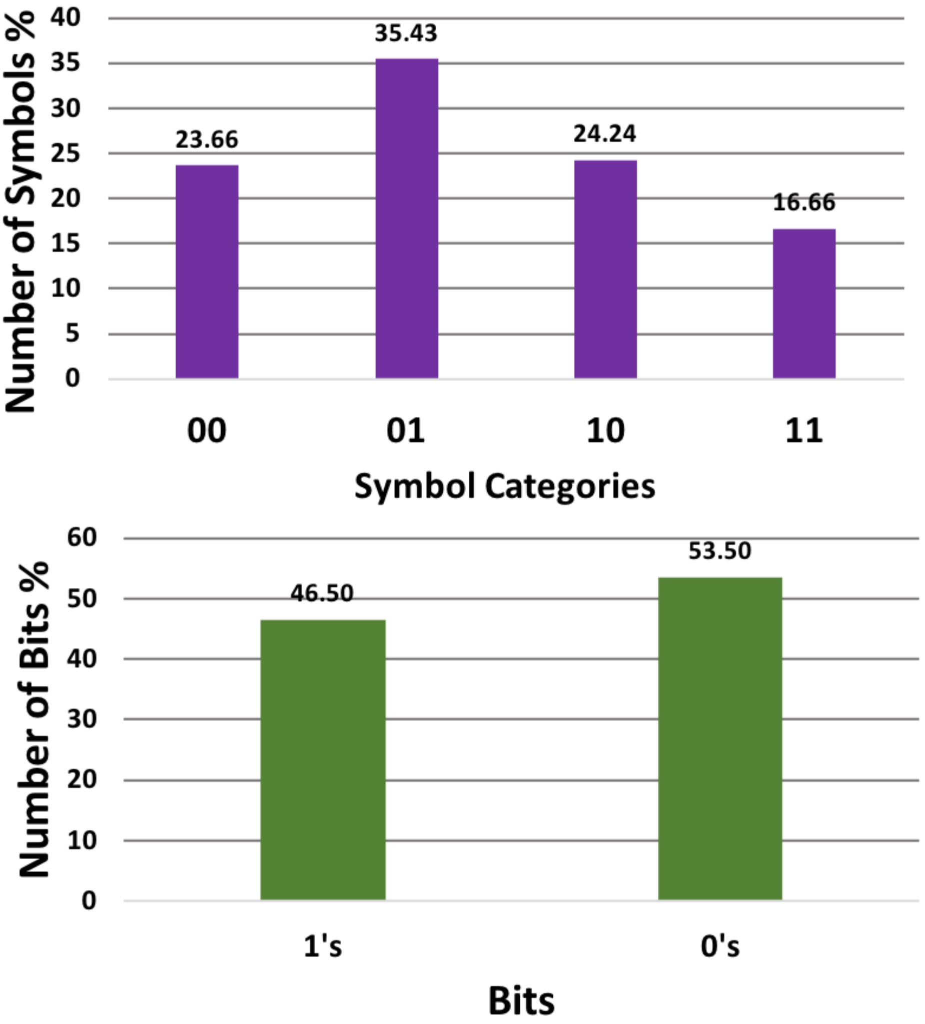


Figure 3.9: The top and bottom panels in this figure provide the same information that is shown in Figure 3.8, but about only the test set. Comparing with the test set of the second type of data (Figure A.2, in Appendix A.3), there is a difference between the percentage values of the symbols for each class because the data is generated based on real text. But all of them have similar percentages of the bits′ patterns (0′s and 1′s). The percentage values are obtained from the first test set of the third type of data.

where t is time, and $\tilde{f}$ is the noisy frequency defined as follows:

$$\tilde{f} = f + \mathcal{E}_f \, , \tag{3.5}$$

where f is the signal frequency of the wave (Table 3.2). $\mathcal{E}_s$ and $\mathcal{E}_f$ represent the signal noise and frequency noise, respectively. The white Gaussian noise ($\mathcal{E}$) is defined as follows:

$$\mathcal{E} \in \mathcal{N}(0, \sigma^2), \tag{3.6}$$

where a value of 2 is used for σ to calculate $\mathcal{E}_f$ and a value of 0.5 is used for σ to calculate $\mathcal{E}_s$ in my simulation. Four data sets of sinusoidal signals are used, each is a different combination of two

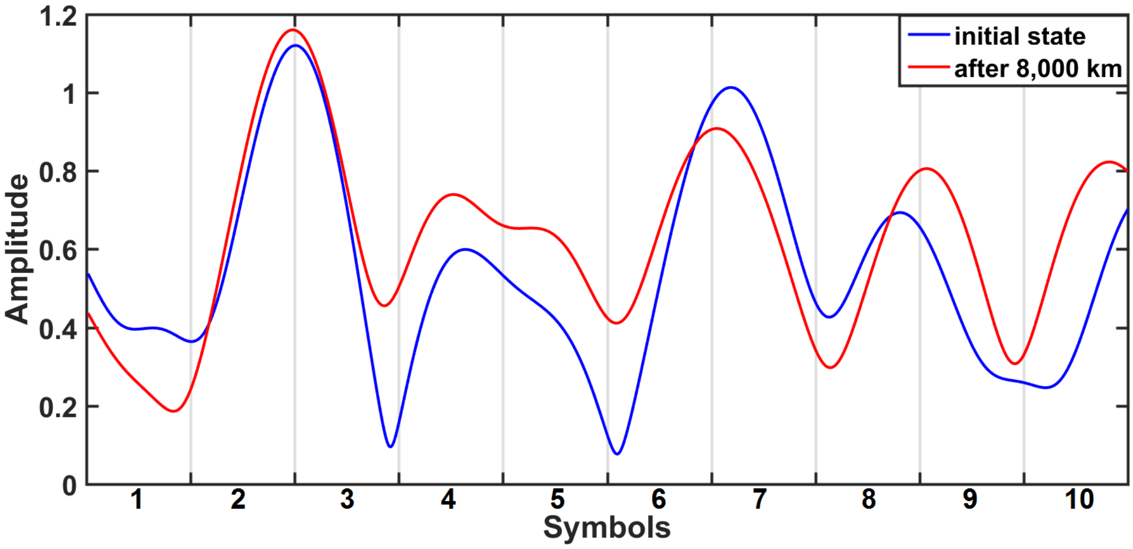


Figure 3.10: The amplitude of a part of an optical transmission signal that contains ten symbols. This signal is obtained from the third type of data (the first data set) at the initial state and after a distance of 8,000 km (X-polarization). As one can see, the amplitude in the initial state is different from one symbol to another, because of the NRZ modulation type. It is the same as the second type of data (Figure 3.5), and different from the first type of data that is generated using a different modulation type (RZ) (Figure 3.2).

classes: AC, AB, BD and AD, namely. For example, the data set AC is a combination of A and C classes.

Figure 3.12 shows that the difference between the values of the signal frequency for class A and C is quite high (a difference of $20 - 10 = 10$), while the value of standard deviation for each class is relatively small, that is a value of 2. Consequently, 1.33% of the waves are ambiguous. Figure 3.13 shows that for classes with closer values of signal frequency, such as A and B with a difference between the values of signal frequency of 5, and with the same standard deviation value (that is 2), more data are overlapped. Note that Figure 3.12 and 3.13 will be important when you look at results in Chapter 7.

3.2.2 Sinusoidal signals with phase noise

Two classes of sinusoidal signals are initialized with simulated phase noise via a Gaussian distribution based on a different value of signal phase; that is 0 radians (first class), and $\frac{\pi}{2}$ radians (second class). Again each class of data consists of 500 signals. Each signal has a corresponding class label, and is

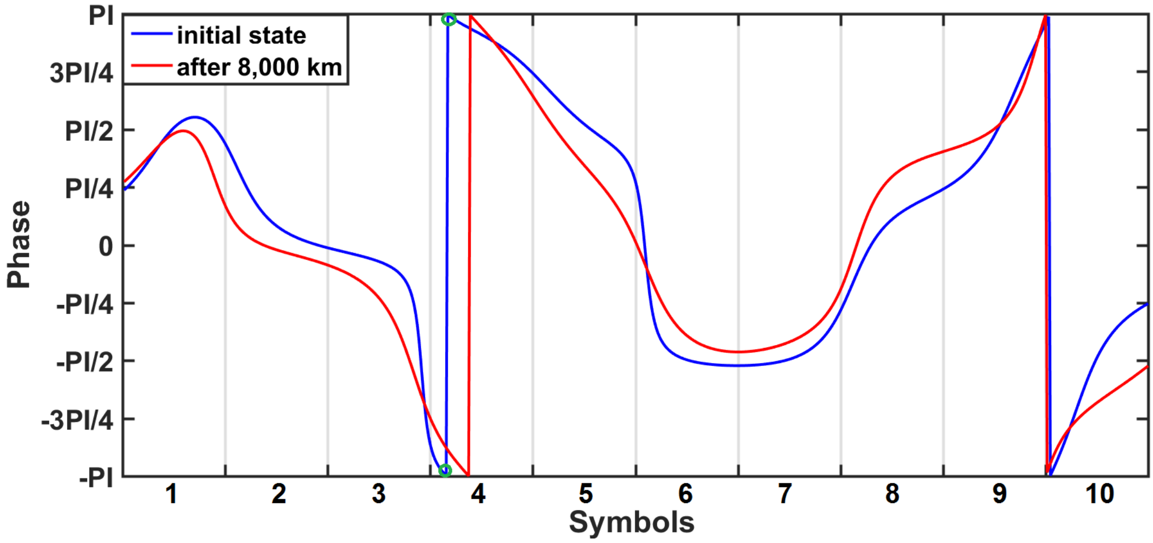


Figure 3.11: The phase of a part of an optical transmission signal that contains ten symbols. This signal was obtained from the third type of data (the first data-set/run) at the initial state and after a distance of 8,000 km (X-polarization). The green circles show the angle's discontinuity between π and $-\pi$ radians, where the phase values of the two points at the top and bottom are actually next to each other (Note that only one pair of discontinuous points is circled as an example). More details can be found in section 3.3.1 (see Figure 3.1). The shape of the signal here is similar to the signal from the second type of data since both types are generated using the same modulation method. Note that PI denotes π.

Data	**Type of noise**	**Data sets**	**Class**	f **(Hz)**
Sinusoidal signals with noisy frequency	$\mathcal{E}_s$ with $\mathcal{E}_f$	AC	A	10
			C	20
		AB	A	10
			B	15
		BD	B	15
			D	12
		AD	A	10
			D	12

Data	**Type of noise**	**Data sets**	**Class**	φ **(°)**
Sinusoidal signals with noisy phase	$\mathcal{E}_s$ with $\mathcal{E}_\varphi$	One	1^{st}	90
			2^{nd}	0

Table 3.2: A summary of description of the sinusoidal signals. f and φ refer to the signal frequency and signal phase of the wave in each class, respectively. $\mathcal{E}_s$ with $\mathcal{E}_f$ refer to signal noise and frequency respectively. $\mathcal{E}_s$ with $\mathcal{E}_\varphi$ refer to signal noise and phase noise, respectively. The number of waves/signals in each class is 500 signals.

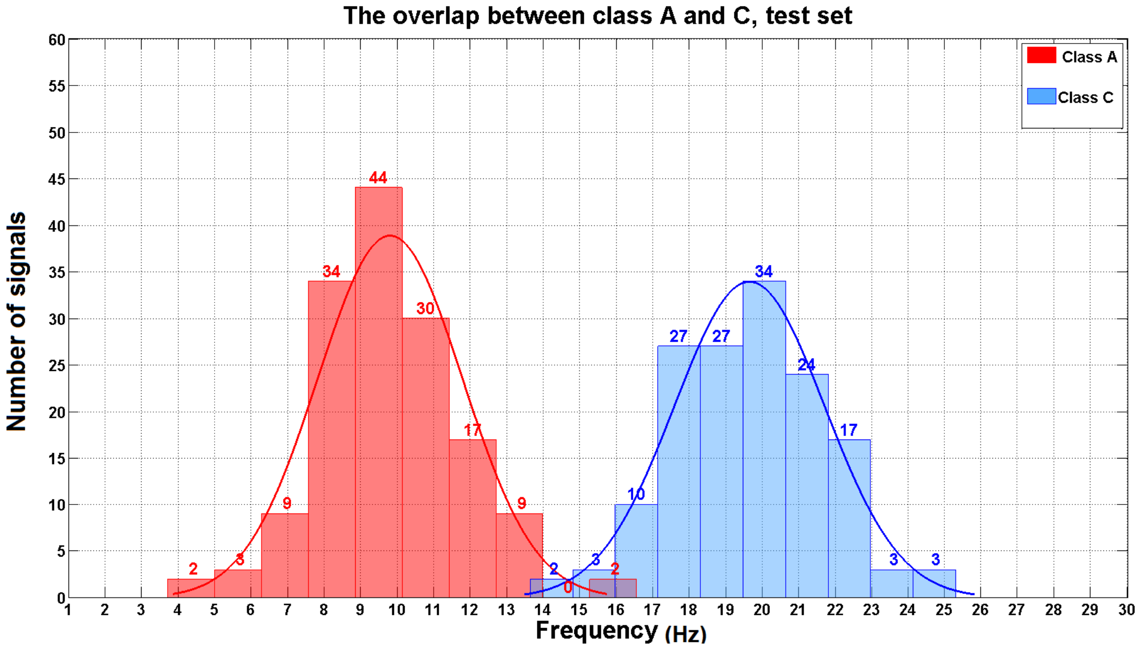


Figure 3.12: The histogram of the test set (AC). The difference between the values of the signal frequency for class A and C is quite high (a difference of 10). Consequently, only 1.33% of the waves are ambiguous. Note: this figure will be important when you look at results in Chapter 7.

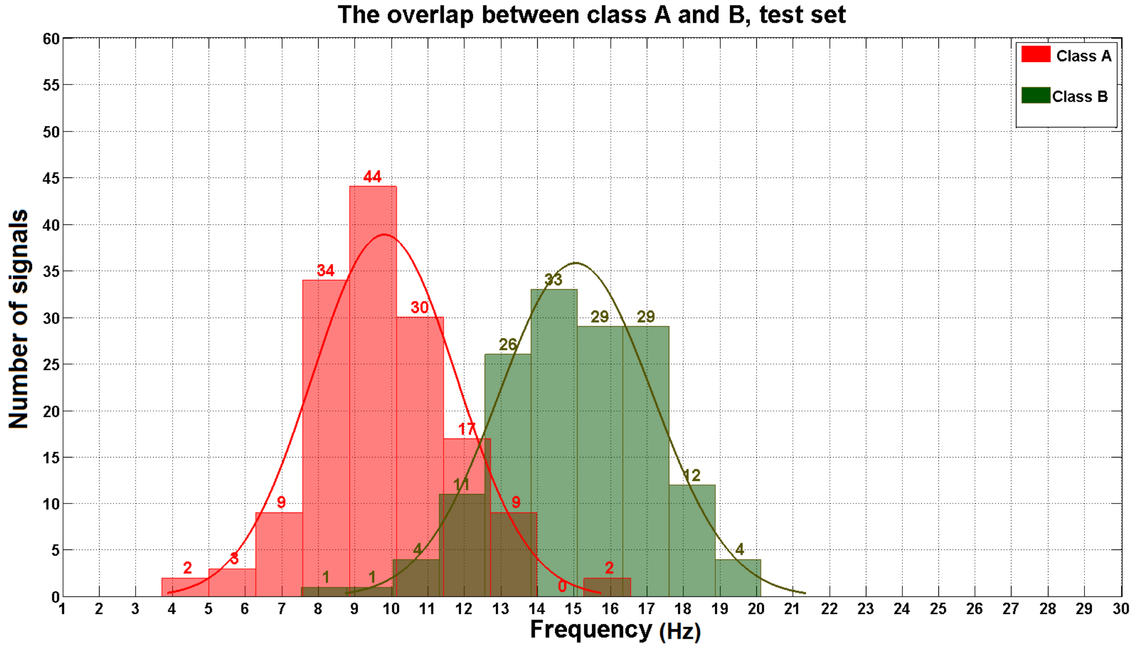


Figure 3.13: The histogram of the test set (AB). The difference between the mean values (signal frequencies) of these two Gaussian distributions is 5. The figure shows that for these classes with closer means the data are more overlapping compared with the test set (AC) in Figure 3.12. Note: this figure will be important when you look at results in Chapter 7.

represented as a 640 vector (samples). A signal noise is again added at each y values of each signal. The signal is generated as follows:

$$Signal(t) = \sin(t + \tilde{\varphi}) + \mathcal{E}_s \, , \tag{3.7}$$

where t is time, $\tilde{\varphi}$ is the noisy phase, defined as follows:

$$\tilde{\varphi} = \varphi + \mathcal{E}_\varphi \tag{3.8}$$

where φ is the signal phase of the wave (Table 3.2). $\mathcal{E}_s$ and $\mathcal{E}_\varphi$ represent the signal noise and the phase noise, respectively. The white Gaussian noise ($\mathcal{E}$) is defined previously in Eq. 3.6, where a value of 0.5 is used for σ to calculate $\mathcal{E}_\varphi$ and a value of 1 is used for σ to calculate $\mathcal{E}_s$ in my simulation. A data set of the sinusoidal signals having phase noise is generated from the two classes 0 and $\pi/2$ radian.

3.3 Data Pre-processing

This section presents the types of processing that I have done on my data before starting the experiments. All of them have been applied to the optical transmission data, but wavelet transformation has been also applied to sinusoidal signals.

3.3.1 Continuity of the classes

Each symbol in my optical transmission data consists of 64 complex numbers as $a + ib$, which can also be represented as $r(\cos\theta + i\sin\theta)$, and θ is the phase of the signal. The phase of the signal can be calculated as follows:

$$\tan\theta = \frac{b}{a}$$

The difficulty of using only phase values as input to the classifier is discussed in this section.

Figures 3.14 and 3.15 show plots of phase values for the central sample of the optical transmission signal at the initial state and at the distance of 8,000 km, respectively. The reason that I use the central sample is because this is the value widely used in decoding in the optical transmission field. More details can be found in Section 4.1. It can be seen from both figures that class "11" was split into two parts, which makes the classifier treat those two parts as two different classes. This is caused by the discontinuity between the phase values π and $-\pi$ (as it is shown by the green circles in Figures 3.3, 3.6 and 3.11), or between the phase values 0 and 2π (as it is shown by the green circles in Figure 3.18). For this reason, values of cos and sin (the real and imaginary parts of the complex number) have been used as input to the classifiers. As can be seen in Figures 3.16 and 3.17, using the sin and cos value makes the four classes continuous.

Considering symbol errors, there are two types: one-bit errors and two-bit errors. For example, if a symbol from "00" (red) (see Figure 3.15) is mis-classified with a one-bit error, it can appear either in the class of "01" (blue) or "10" (green). That is, the symbol from the current class appears (or is predicted) in the next class. But in the case of a two-bit error, the symbol from the current class passes the adjacent classes to appear in the class that follows the adjacent class. For instance, it is clear in Figure 3.15 that some symbols have two bit errors and some have one bit errors, some symbols from class "01" (blue dots) have a two-bit error, which make them appear between the green dots, which are supposed to belonged to class "10".

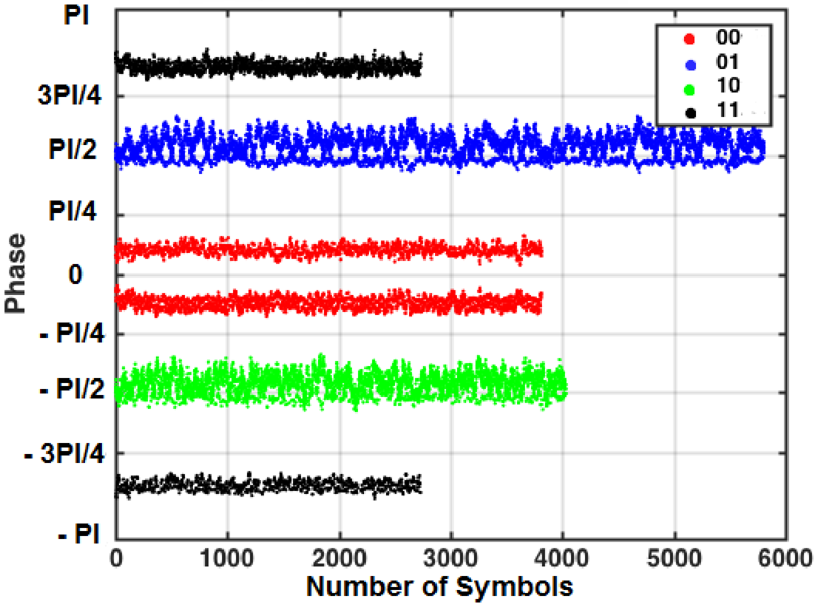


Figure 3.14: The phase value of the 33^{rd} sample at the initial state, the first data set of the third type of data, X-polarization. As can be seen here class "11" (in black) has split into two parts, which makes the classifier considers those two parts as two classes. This because of the discontinuity between the phase values π and $-\pi$. Note that PI denotes π.

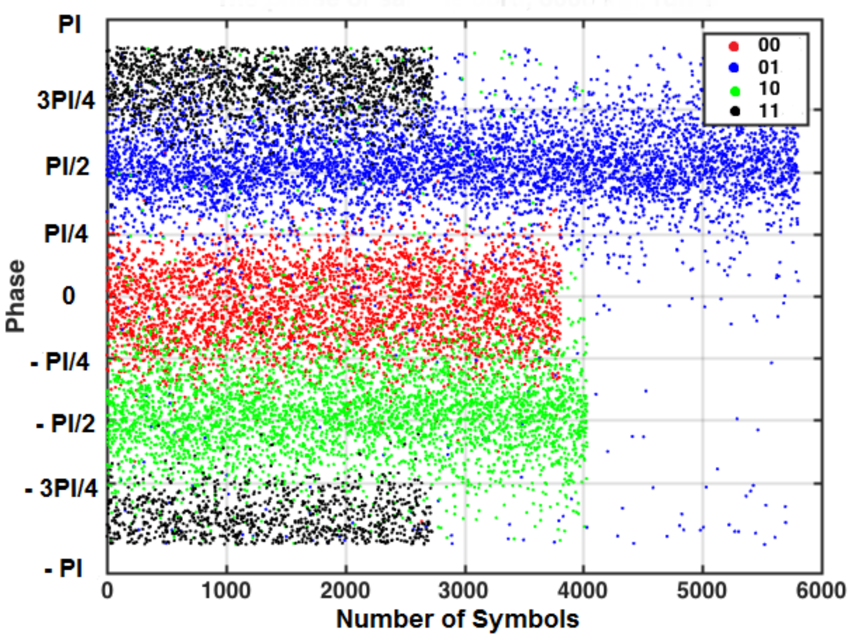


Figure 3.15: The phase value of the 33^{rd} sample at 8,000 km, the first data set of the third type of data, X-polarization. As can be seen here class “11” (in black) has split into two parts, which makes the classifier considers those two parts as two classes. This because of the discontinuity between the phase values π and $-\pi$. Note that PI denotes π.

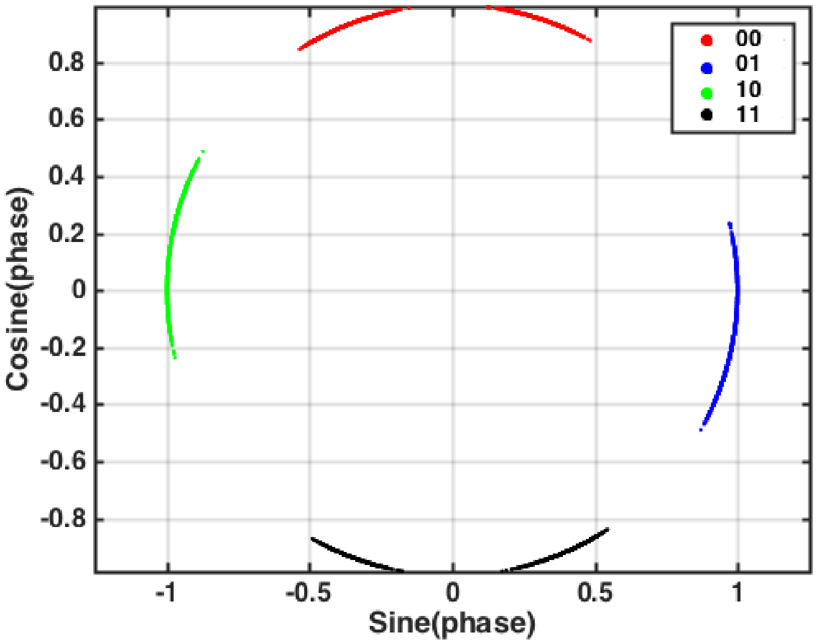


Figure 3.16: The continuity of the classes using Trigonometric functions at the initial state, the first data set of the third type of data, X-polarization. Using sine and cosine of the phase values ensures that the four classes are continuous. As a result, the two parts of class “11”, which resulted from using just the phase value of the signal (in Figure 3.14), can be distinguished as one class by the classifier.

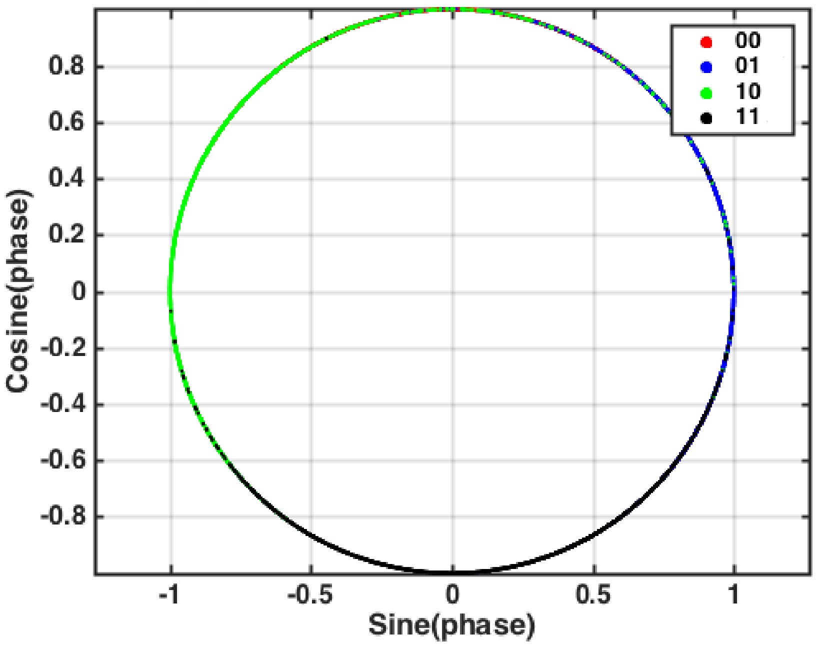


Figure 3.17: The continuity of the classes using Trigonometric functions at 8,000 km, the first data set of the third type of data, X-polarization. Using sine and cosine of the phase values ensures that the four classes are continuous. As a result, the two parts of class "11", which resulted from using just the phase value of the signal (in Figure 3.15), can be distinguished as one class by the classifier.

3.3.2 Neighbouring information

Using information from either side of the symbol being analyzed as a part of the input to a classifier can help to decrease the number of errors on decoding the signal. This has been shown in (Hunt et al., 2009). The reason that using neighbouring information can help to improve the BER can be illustrated in Figure 3.18, which shows three consecutive symbols at the initial state and at a distance of 10,000 km (example taken from the second type of data).

As we can see from Figure 3.18, the first symbol has a phase of π whereas the phase of the middle symbol is 0 (or 2π). However, at the distance of 10,000 km the central (second) symbol has been degraded. Since the first (preceding) symbol has pulled the second symbol from 2π towards its own value of π, which led to the prediction of the middle symbol at the middle point as having a value nearer to $3\pi/2$ and so belonging to the class "10". From this observation, I conclude that the neighbouring symbols can affect the target symbol, for which I want to predict the label. Therefore, I shall investigate the effect of using symbols either side of the target in an attempt to reduce the bit error ratio by including a set of different numbers of neighbouring symbols over all distances.

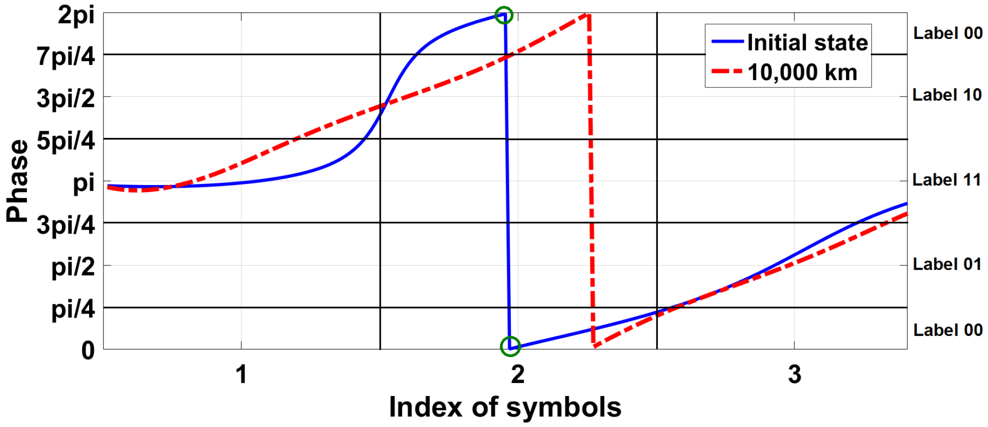


Figure 3.18: Three contiguous symbols show the effect caused by the symbols either side. First of all, looking at the middle symbol (in blue) at the initial state and at the distorted signal (in red) after 10,000 km which has been affected by the symbol either side, especially the preceding symbol. Now if you look at the middle of the second symbol, where the red and blue lines cross, we can notice that the red line has been dragged to be just in the area of class “10”. Note the green circles, show the angle’s discontinuity between 2π and 0 radians, where the phase values of the two points at the top and bottom are actually next to each other, more details in Section 3.3.1. Note that the y axis is plotted using the positive angle values from 0 to 2π instead of from 0 to π and from 0 to $-\pi$ (see Section 1.6, Figure 1.4). Note that pi denotes π.

3.3.3 Wavelet transformation

The first historical reference to the wavelet transform is the Haar wavelet, which was proposed by the mathematician Alfrd Haar in 1909. At that time, the concept of wavelets did not exist, until the geophysicist, Jean Morlet, proposed the idea in 1981. Then, the term wavelet was invented by Jean Morlet and the physicist Alex Grossman in 1984. The Haar wavelet was the only orthogonal wavelet people knew before 1985. In 1985, the second orthogonal wavelet was constructed by the mathematician Yves Meyer, and it is called Meyer wavelet. A year after holding the 1st international conference in France in 1987, many types of wavelets were proposed including the concept of multi-resolution that was introduced by Stephane Mallat and Yves Meyer. Also, finding a systematical method to construct the compact support orthogonal wavelet was proposed by Ingrid Daubechies in the same year, 1988. In 1989, the fast wavelet transform was proposed by Stephane Mallat, which helped in using wavelets in lots of signal processing applications (Chun-Lin, 2010), (Hubbard, 1998). Figure 3.19 shows different types of wavelets: Daubechies, Haar, Morlet, Symlets, Coiflets, and Meyer (Alegria et al., 2015).

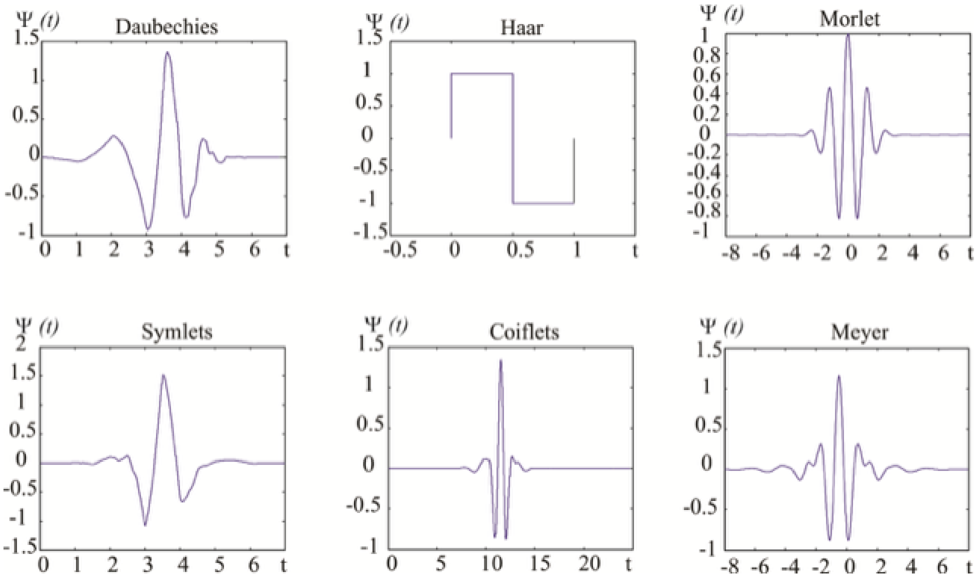


Figure 3.19: Different types of wavelets: Daubechies, Haar, Morlet, Symlets, Coiflets, and Meyer (Alegria et al., 2015).

Wavelet transform (WT) is a mathematical tool that can be used for extracting information from a variety of data forms, such as image and audio signals (Lee and Lim, 2012). It is important in the deconstruction of non-stationary and/or non-periodic signals. Therefore, it has been used in different places in signal processing (Hunt et al., 2009). The theory of wavelet is currently utilized as an essential technique in specialized research in electronics, mechanics, computers, communications, medicine, biology, astronomy and so on. In the field of image and signal processing, the fundamental use of wavelet is to compress and de-noise data (Liu et al., 2013). When undertaking the WT, two components are extracted from the signal: the approximation and detail components. The approximation sub-signal is defined as the high scale, low-frequency component of the original signal. It is also referred to as a smoothed signal. The detail sub-signal is the low scale, high-frequency component of the original signal (Hunt et al., 2009). In this work, I start with the simplest wavelet transform: Haar wavelet transform, which can be used for signal decomposition (Walker, 2008).

Haar wavelets have been used extensively as examples in teaching due to its simplicity. In fact, it is the simplest wavelet and has been a prototype for all other types of wavelet transforms (Varma et al., 2012). Suppose a signal $\mathbf{x}$ consists of N elements, that is

$$\mathbf{x} = \{x_i\}_{i=1}^{N}$$

By using Haar transform, the signal can be decomposed into the approximation part and the detail part. The approximation part captures the trend of each signal and is defined as follows:

$$\mathbf{a_1} = \frac{1}{\sqrt{2}}\,[x_1 + x_2,\, x_3 + x_4,\, ...,\, x_{N-1} + x_N] \tag{3.9}$$

It can be seen from Eq. (3.9), that the approximation part consists of a set of average values for each pair in the original signal, divided by the square root of 2.

On the other hand, the detail part of the signal $\mathbf{x}$ keeps the small fluctuations of each feature and is defined as follows:

$$\mathbf{d_1} = \frac{1}{\sqrt{2}}\,[x_1 - x_2\,,\, x_3 - x_4\,,\, ...,\, x_{N-1} - x_N] \tag{3.10}$$

Eq.(3.10) shows that the detail part captures the difference between each pair in the original signal, divided by the square root of 2.

Note that an important property of the Haar transform is that it conserves the energies of signals. The energy of the signal (E) is defined as follows:

$$E = \int \mathbf{x}^2 dt, \tag{3.11}$$

In the discrete case, where the energy is defined as the sum of the squares of signal values, that is:

$$E = \sum_{i=1}^{N} \mathbf{x}_i^2 \tag{3.12}$$

The energy of the approximation part accounts for a large percentage of the total energy. $\sqrt{2}$ is used in Eq. (3.9) and (3.10) so that the Haar transform conserves the energy of a signal (Walker, 2008).

The Haar transform can be performed at multiple levels. At the top level, that is level one (level-1), a signal is transformed to two sub-signals a_1 (see Eq. (3.9)) and d_1 (see Eq. (3.10)). The second level is then carried out by computing a second trend a_2 and a second fluctuation d_2 for a_1 only. That is, we can continue recursively with the same process to work on the next level, where the signal is always the approximation part obtained from the preceding level.

For example, a signal $\mathbf{x} = [11, 9, 31, 4, 21, 5, 10, 8]$ consists of eight values. The first approximation of the signal is computed as follows:

$$\mathbf{a_1} = [\frac{11+9}{\sqrt{2}}, \frac{31+4}{\sqrt{2}}, \frac{21+5}{\sqrt{2}}, \frac{10+8}{\sqrt{2}}] = [10\sqrt{2}, 17.5\sqrt{2}, 13\sqrt{2}, 9\sqrt{2}]$$

The first detail is calculated as follows:

$$\mathbf{d_1} = [\frac{11-9}{\sqrt{2}}, \frac{31-4}{\sqrt{2}}, \frac{21-5}{\sqrt{2}}, \frac{10-8}{\sqrt{2}}] = [\sqrt{2}, 13.5\sqrt{2}, 8\sqrt{2}, \sqrt{2}]$$

The approximation part keeps the trend of the original signal and the biggest difference comes from the second pair of data shown in the detail.

The second approximation and detail parts of the signal ($\mathbf{x}$) is computed by decomposing the first approximation part (a_1). The second approximation of the signal ($\mathbf{x}$) is computed as follows:

$$\mathbf{a_2} = [\frac{10\sqrt{2} + 17.5\sqrt{2}}{\sqrt{2}}, \frac{13\sqrt{2} + 9\sqrt{2}}{\sqrt{2}}] = [27.5, 22]$$

The second detail is calculated as follows:

$$\mathbf{d_2} = [\frac{10\sqrt{2} - 17.5\sqrt{2}}{\sqrt{2}}, \frac{13\sqrt{2} - 9\sqrt{2}}{\sqrt{2}}] = [-7.5, 4]$$

One more example is displayed in Figure 3.20, the top panel shows a sinusoidal signal with noise (in purple). The panels in the middle and the bottom show the 2-level Haar transform of the noisy signal at the top. It can be seen that the approximation part of the signal (in the middle panel) becomes smoother than the original noisy one.

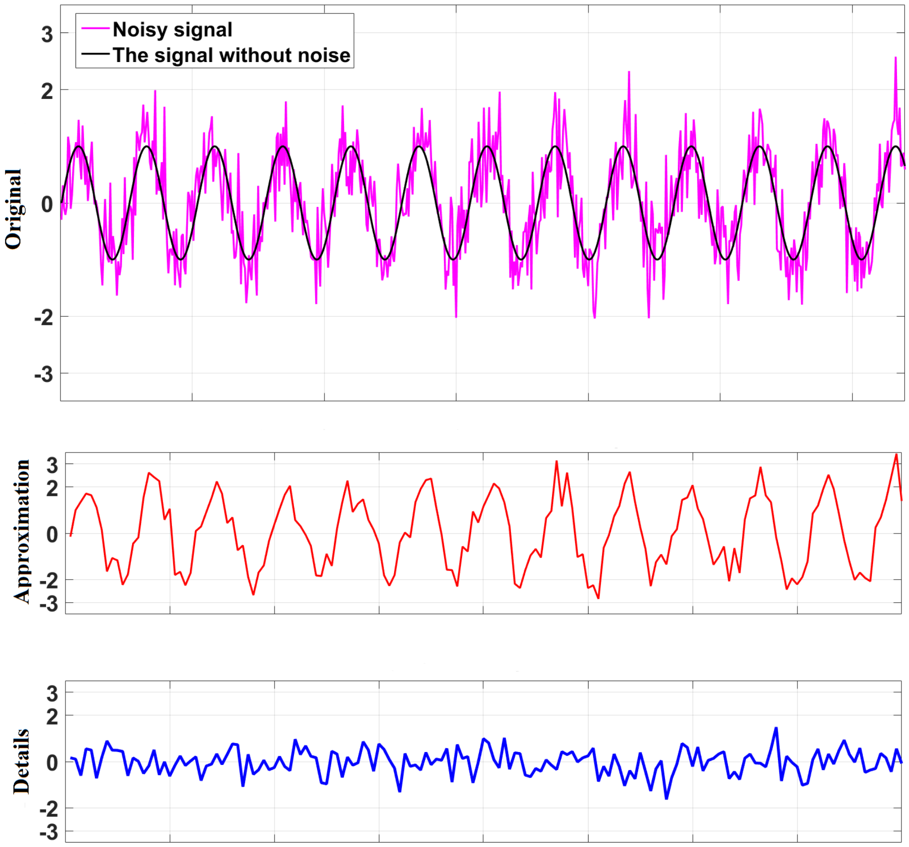


Figure 3.20: The top panel shows a sinusoidal signal with noise (in purple). The black signal is the same noisy signal, but without any noise, to show how much the signal was distorted by the noise. Haar wavelet decompositions (level-2) of the noisy signal in the top are shown in the middle and bottom panels.

Apart from the Haar transform, there are many other different types of wavelet transforms. For example, another widely used type is referred to Daubechies wavelets. Daubechies wavelet transform has been successfully applied in many engineering related works (Williams and Amaratunga, 1994). Similar to the Haar wavelets, it decomposes signals into the approximation and detail parts, and preserves the energy of each signal. However, Daubechies uses more data points to compute both the trend and fluctuations rather than using just pairs of data as was done in the Haar transform. Moreover, unlike the Haar wavelets using $\sqrt{2}$, Daubechies uses different numbers to multiply with data points so that the signal's energy can be kept. One of the Daubechies wavelet family is Daub4 wavelet transform. The scaling numbers in Daub4 transform are defined as follows:

$$\alpha_1 = \frac{1+\sqrt{3}}{4\sqrt{2}},\ \alpha_2 = \frac{3+\sqrt{3}}{4\sqrt{2}},\ \alpha_3 = \frac{3-\sqrt{3}}{4\sqrt{2}},\ \alpha_4 = \frac{1-\sqrt{3}}{4\sqrt{2}}$$

The wavelet numbers in Daub4 transform are defined as follows:

$$\beta_1 = \frac{1-\sqrt{3}}{4\sqrt{2}},\ \beta_2 = \frac{\sqrt{3}-3}{4\sqrt{2}},\ \beta_3 = \frac{3+\sqrt{3}}{4\sqrt{2}},\ \beta_4 = \frac{-1-\sqrt{3}}{4\sqrt{2}}$$

Foe a signal $\mathbf{x} = \{x_i\}_{i=1}^{N}$ has N elements, the first approximation part can be calculated using the daub4 transform as follows:

$$\mathbf{a_1} = [e_1^a\,,\, e_2^a\,,\, ...\,,\, e_{N/2}^a]$$

where e_1^a , e_2^a , ... , $e_{N/2}^a$ are values that can be computed using the scaling numbers α_1, α_2, α_3 and α_4 as follows:

$$\begin{aligned} e_1^a &= x_1 \times \alpha_1 \,+\, x_2 \times \alpha_2 \,+\, x_3 \times \alpha_3 \,+\, x_4 \times \alpha_4 \\ e_2^a &= x_3 \times \alpha_1 \,+\, x_4 \times \alpha_2 \,+\, x_5 \times \alpha_3 \,+\, x_6 \times \alpha_4 \\ &\vdots \\ e_{N/2}^a &= x_{N-1} \times \alpha_1 \,+\, x_N \times \alpha_2 \,+\, x_1 \times \alpha_3 \,+\, x_2 \times \alpha_4 \end{aligned}$$

Also, the first detail part will be computed the same as calculating the first approximation part, but using the wavelet numbers β_1, β_2, β_3 and β_4 as follows:

$$\mathbf{d_1} = [e_1^d\,,\, e_2^d\,,\, ...\,,\, e_{N/2}^d]$$

where e_1^d , e_2^d , ... , $e_{N/2}^d$ are values that can be computed as follows:

$$\begin{aligned} e_1^d &= x_1 \times \beta_1 \,+\, x_2 \times \beta_2 \,+\, x_3 \times \beta_3 \,+\, x_4 \times \beta_4 \\ e_2^d &= x_3 \times \beta_1 \,+\, x_4 \times \beta_2 \,+\, x_5 \times \beta_3 \,+\, x_6 \times \beta_4 \\ &\vdots \\ e_{N/2}^d &= x_{N-1} \times \beta_1 \,+\, x_N \times \beta_2 \,+\, x_1 \times \beta_3 \,+\, x_2 \times \beta_4 \end{aligned}$$

For instance, the first approximation part of the signal $\mathbf{x} = [11, 9, 31, 4, 21, 5, 10, 8]$ can be calculated as follows:

$$\begin{aligned} e_1^a &= 11 \times \alpha_1 \,+\, 9 \times \alpha_2 \,+\, 31 \times \alpha_3 \,+\, 4 \times \alpha_4 \\ e_2^a &= 31 \times \alpha_1 \,+\, 4 \times \alpha_2 \,+\, 21 \times \alpha_3 \,+\, 5 \times \alpha_4 \\ e_3^a &= 21 \times \alpha_1 \,+\, 5 \times \alpha_2 \,+\, 10 \times \alpha_3 \,+\, 8 \times \alpha_4 \\ e_4^a &= 10 \times \alpha_1 \,+\, 8 \times \alpha_2 \,+\, 11 \times \alpha_3 \,+\, 9 \times \alpha_4 \end{aligned}$$

As a result, e_1^a will be:

$$\begin{aligned} e_1^a &= (11 \times 0.483) \,+\, (9 \times 0.836) \,+\, (31 \times 0.224) \,+\, (4 \times (-0.129)) \\ e_1^a &= 5.313 + 7.524 + 6.944 + (-0.516) \\ e_1^a &= 19.265 \end{aligned}$$

e_2^a will be:

$$\begin{aligned} e_2^a &= (31 \times 0.483) \,+\, (4 \times 0.836) \,+\, (21 \times 0.224) \,+\, (5 \times (-0.129)) \\ e_2^a &= 14.973 + 3.344 + 4.704 + (-0.645) \\ e_2^a &= 22.376 \end{aligned}$$

e_3^a will be:

$$\begin{aligned} e_3^a &= (21 \times 0.483) \,+\, (5 \times 0.836) \,+\, (10 \times 0.224) \,+\, (8 \times (-0.129)) \\ e_3^a &= 10.134 + 4.18 + 2.24 + (-1.032) \\ e_3^a &= 15.522 \end{aligned}$$

e_4^a will be:

$$\begin{aligned} e_4^a &= (10 \times 0.483) \,+\, (8 \times 0.836) \,+\, (11 \times 0.224) \,+\, (9 \times (-0.129)) \\ e_4^a &= 4.83 + 6.688 + 2.464 + (-1.161) \\ e_4^a &= 12.821 \end{aligned}$$

Then $\mathbf{a_1}$ will be as follows:

$$\mathbf{a_1} = [19.265\,,\ 22.376\,,\ 15.522\,,\ 12.821]$$

The first detail part of the signal $\mathbf{x}$ is calculated as follows:

$$\begin{aligned}
e_1^d &= 11 \times \beta_1 \ + 9 \times \beta_2 \ + 31 \times \beta_3 \ + 4 \times \beta_4 \\
e_2^d &= 31 \times \beta_1 \ + 4 \times \beta_2 \ + 21 \times \beta_3 \ + 5 \times \beta_4 \\
e_3^d &= 21 \times \beta_1 \ + 5 \times \beta_2 \ + 10 \times \beta_3 \ + 8 \times \beta_4 \\
e_4^d &= 10 \times \beta_1 \ + 8 \times \beta_2 \ + 11 \times \beta_3 \ + 9 \times \beta_4
\end{aligned}$$

Then, the result will be as follows:

$$\mathbf{d_1} = [20.549\,,\ 10.246\,,\ 0.667\,,\ 1.767]$$

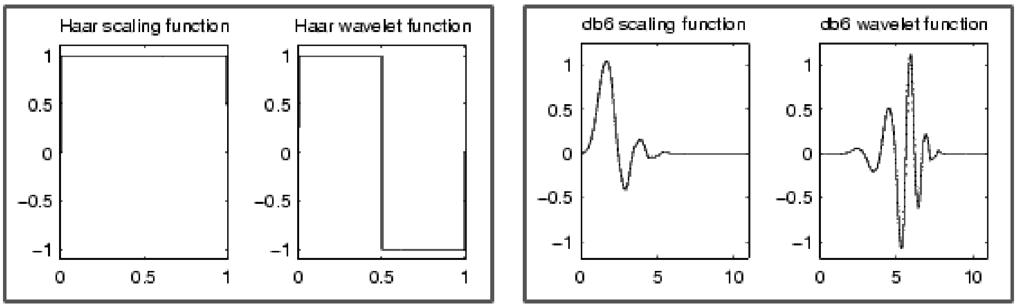


Figure 3.21: An example of One-Dimensional Wavelets, db6 and Haar wavelet transforms (cited from Wavelet Toolbox, n.d.).

In this work the Daubechies wavelet representation has not provided better results than the Haar wavelet transform. Therefore, I have focused on the Haar wavelet transform. Figure 3.21 shows an example of the scaling and wavelet function of both Haar and Daubechies (db6) wavelet transforms. As we can see from Figure 3.21, the Haar wavelets has two constant scaling and wavelet numbers for the approximation and detail parts, but daub4 has four variable scaling and wavelet numbers for both of them.

A MATLAB function, which is dddtree(), was used to decompose the signals. It takes four parameters as follows:

$$wt = dddtree(typetree, \mathbf{x}, level, df)$$

where discrete wavelet transform 'dwt' was used as a typetree, $\mathbf{x}$ is the input signal, level is the number of times the transform is applied, and 'df' is the decomposition filters that is used by the wavelet transform, which are 'haar' and 'db4' in this research. The argument 'wfilters' was used for wavelet filters.

3.4 Conclusion

In this chapter, I have presented the description of my data that I have used in this research. I have described some data pre-processing that is used before using the data as an input to the classifier, such as using neighbouring information and the wavelet transforms (WTs). I conclude that I am going to use the neighbouring information in my research, since they have affected the target symbol being classified, hence should reduce the bit errors. Also, in this research, I start with using the simplest wavelet transform (Haar wavelet transform), for decomposing the signal. Next chapter introduces all the methods that I have used in my research.

Chapter 4

Methods

There are several categories in machine learning including: supervised learning, unsupervised learning and reinforcement learning (Sanjeevi, 2017). During my PhD study, I have learned and applied methods from both supervised and unsupervised learning. In this chapter, I shall introduce all methods that are used in my research.

We can divide data analysis into two separate areas: 1) when the data is labelled, supervised learning is used; 2) when the data is unlabelled, unsupervised learning (or clustering, which is the most important unsupervised learning method) is used (Mishra, 2017). Apart from clustering, data visualisation and topological mapping also belongs to unsupervised learning. Supervised learning is used to find a mapping from inputs to outputs, and then use this mapping to make the best guess of the label of a new data point. Unsupervised learning is used to identify natural clusters in the data.

In Section 4.1, I present the benchmark method, the threshold method, that is currently used in the hardware implementation of the optical transmission set up. Then I shall introduce Principal Component analysis (PCA) in Section 4.2. PCA can provide the best linear visualization of data so that the structure of data can be easily analysed and interpreted. Section 4.3 describes Support Vector Machine (SVM) for classification. Section 4.4 introduces the toolbox used to apply SVM. The performance measurements that evaluate the efficiency of an SVM classifier are described in Section 4.6.

4.1 The Threshold Method

The threshold method is used as a benchmark method in this work. It is based on the phase value of each symbol's central sample (33^{rd} sample), where the pulse (symbol) is less distorted at the initial state. Then, each symbol is classified to one of four classes according to this value (see Table 4.1). That is: symbols whose central phase values are in the range of $-\frac{\pi}{4}$ and $\frac{\pi}{4}$, belong to the class "00"; in the range of $\frac{\pi}{4}$ and $\frac{3\pi}{4}$ belong to the class "01"; in the range of either $\frac{3\pi}{4}$ and π, or $-\pi$ and $-\frac{3\pi}{4}$, belong to the class "11"; and in the range of $-\frac{3\pi}{4}$ and $-\frac{\pi}{4}$ belong to the class "10".

Phase value of the central sample of symbol (θ)	**Label based on QSPK modulation**
$-\frac{\pi}{4} \leq \theta < \frac{\pi}{4}$	00
$\frac{\pi}{4} \leq \theta < \frac{3\pi}{4}$	01
$\frac{3\pi}{4} \leq \theta < \pi$ (or) $-\pi \leq \theta < -\frac{3\pi}{4}$	11
$-\frac{3\pi}{4} \leq \theta < -\frac{\pi}{4}$	10

Table 4.1: The threshold method.

Another way to show how this threshold method works can be seen in a phase circle diagram (Figure 4.1). For the purpose of convenience, I have presented the same figure (Figure 3.1) shown in Chapter 3. As I have mentioned in Chapter 3, the discontinuity happens in the range of $[-\frac{3\pi}{4}, \frac{3\pi}{4}]$, that is a change from $-\pi$ to π.

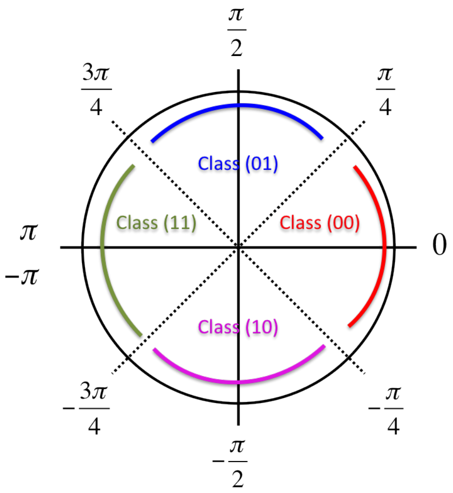


Figure 4.1: Four different variants of the light after QPSK modulation.

4.2 Principal Component Analysis(PCA)

One of the most common mathematical algorithms for pre-processing and visualizing high-dimensional data is Principal Component Analysis PCA (Zou et al., 2006). PCA captures the most variance of the original data (Scholkopf and Smola, 2001) by rotating the data into a new orthogonal coordinate system.

The sum of eigenvalues, which are obtained by computing the new orthogonal coordinates, equals to the total variance in the original data. To visualize the data, we can obtain projections by taking the dot product of the data (with mean of zero) with the eigenvectors. Note each of these projection is a linear combination of the original variables, weighted by the corresponding element in each eigenvector (principal component).

In this work, PCA has been used to visualise, analyse and interpret the structure of the data sets. Figure 4.2 shows a visualization plot obtained on one data set from the first type of data at the initial state using the first two principal components that has captured almost 98.97% of the original signals. As we can see from the figure, there are seven clusters represented by four classes. Each color represents a class which in turn represents a range of phase values according to QPSK modulation (see Figure 4.1). The class “11” is divided into four parts (black color), and that is because of the discontinuity at (π) and ($-\pi$).

Table 4.2 shows a summary on figures (Figures 4.2 - 4.7) obtained using PCA. These figures are produced on a randomly selected dataset from each data type. This table is very important, it explains the difference between the first and second type of data. It can be seen that:

- The first two PCs can capture most (that is greater than 95%) of the variance of the data on the first type of data at the initial state, which means that the data is close to be linear.
- The biggest change on variance captured by the first two PCs happened to the first type of data, where the captured variance dropped from 98.97% to 58.62% from the initial state to the maximum distance. That means the first type of data changed a lot after 3,000 km, and got distorted more than the second and third type of data at the maximum distance.
- The first two PCs can capture most (that is greater than 80%) of the variance of the second and third type of data at the initial state and the maximum distance, which means those types of data are less distorted and more complicated in the classification process.

Figure Index	**Data type**	**Variance captured by the first two PCs %**	**Distance (km)**
4.2	1^{st}	98.97	Initial state
4.3		58.62	3,000
4.4	2^{nd}	83.53	Initial state
4.5		81.14	10,000
4.6	3^{rd}	89.61	Initial state
4.7		84.65	8,000

Table 4.2: A summary of figures from Figure 4.2 to Figure 4.7. PCA has been done on both the initial state and the maximum distance within each type of data. It can be seen that the first two PCs can capture most of the variance of the original data in the initial state. The variance captured by the first two PCs has been decreased at the maximum distance compared with the corresponding initial state, where the largest decrease happened in the first type of data. That means the first type of data changed a lot after 3,000 km, and got distorted more than the second and third type of data at their maximum distance.

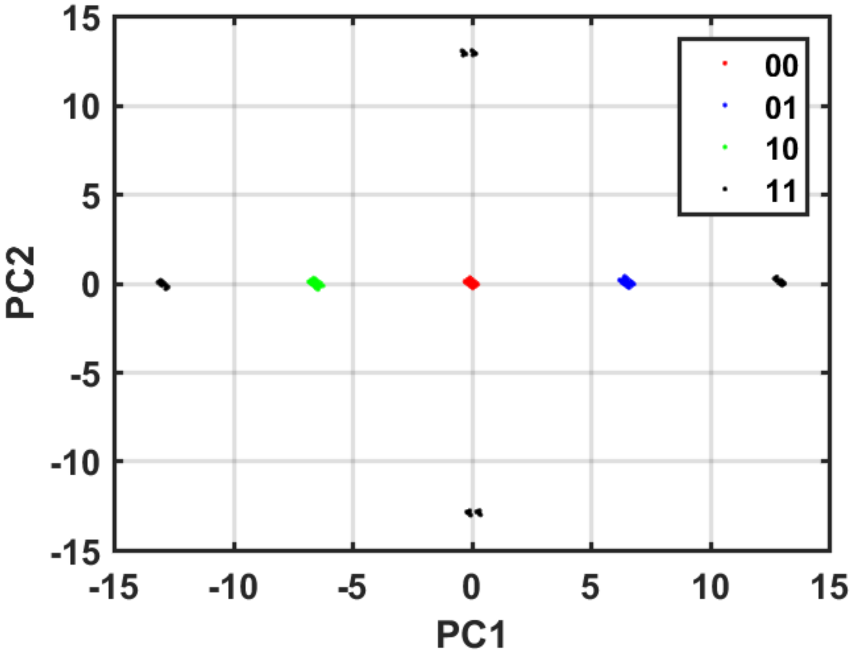


Figure 4.2: PCA biplot with signals' samples of the first type of data (The first data set) at the initial state, plotted in two dimensions using their projections onto the first two principal components.

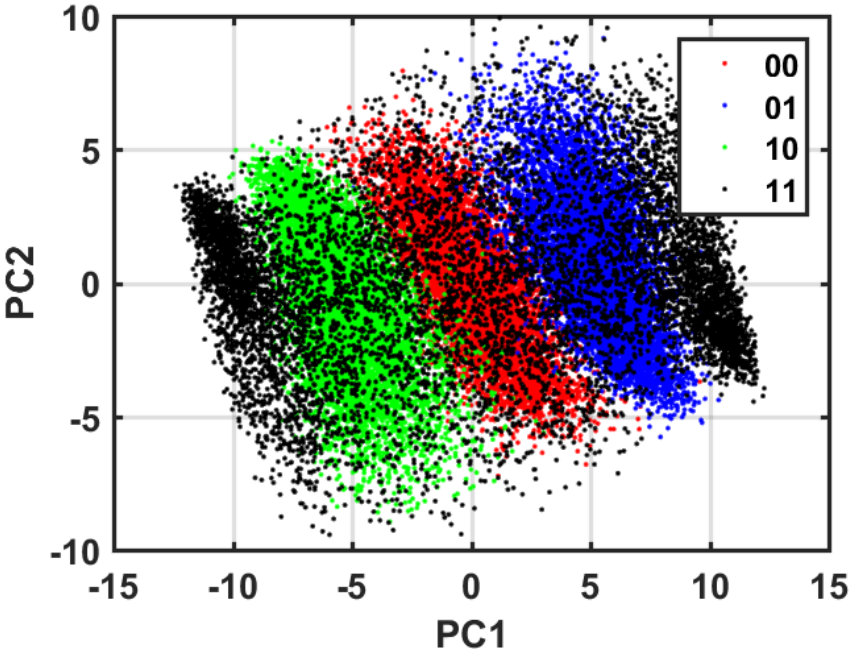


Figure 4.3: PCA biplot with signals' samples of the first type of data (The first data set) at the distance of 3,000km, plotted in two dimensions using their projections onto the first two principal components.

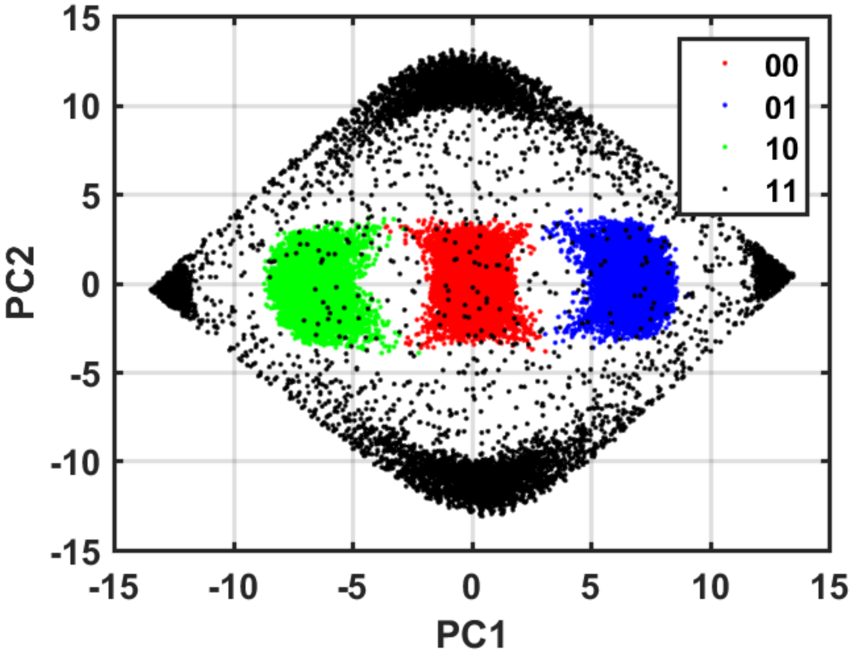


Figure 4.4: PCA biplot with signals' samples of the second type of data (The first data set) at the initial state, X-polarization, plotted in two dimensions using their projections onto the first two principal components.

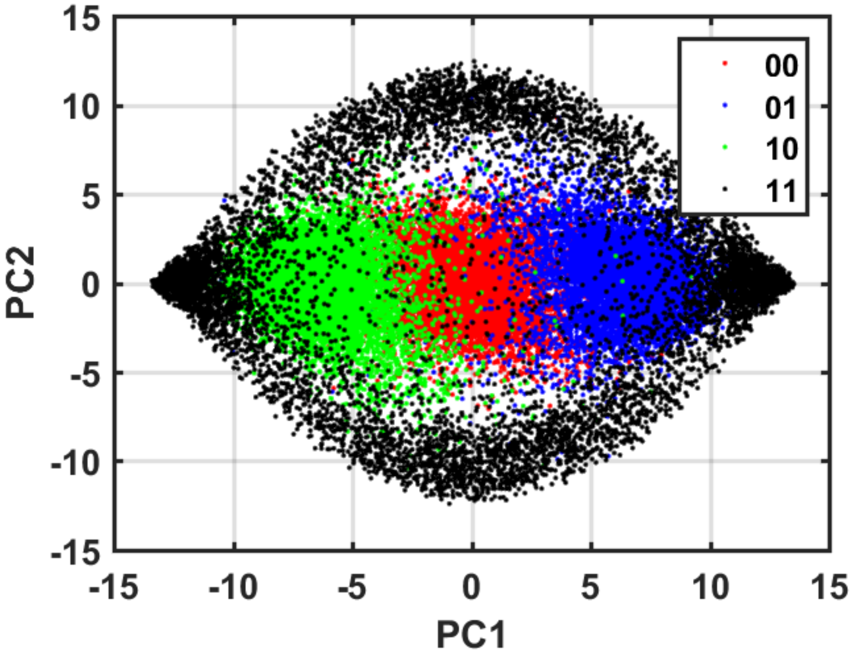


Figure 4.5: PCA biplot with signals' samples of the second type of data (The first data set) at the distance of 10,000km, X-polarization, plotted in two dimensions using their projections onto the first two principal components.

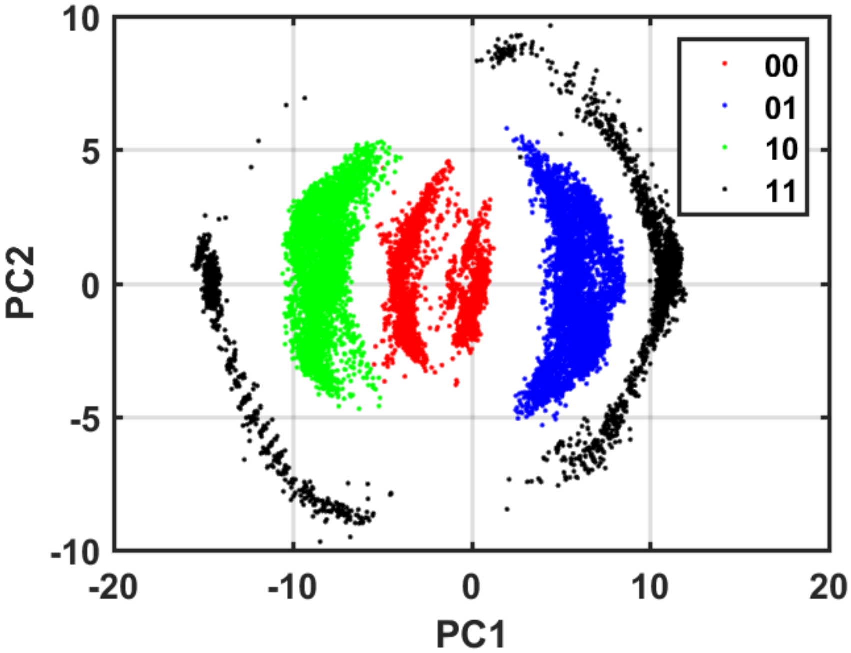


Figure 4.6: PCA biplot with signals' samples of the third type of data (The first data set) at the initial state, X-polarization, plotted in two dimensions using their projections onto the first two principal components.

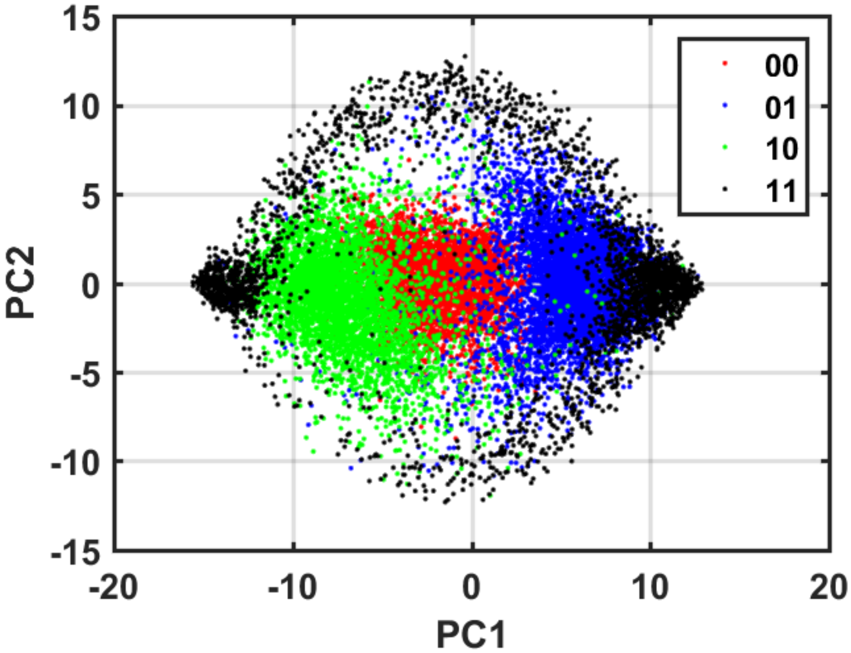


Figure 4.7: PCA biplot with signals' samples of the third type of data (The first data set) at the distance of 8,000km, X-polarization, plotted in two dimensions using their projections onto the first two principal components.

4.3 Support Vector Machine (SVM) Classification

Support Vector Machines (SVM) have been used for classification and regression analysis. They are one of the most popular machine learning (ML) methods that can solve difficult pattern recognition tasks in many different domains when I began my PhD study. Even now they are arguably one of the most successful methods in ML. SVMs can be both linear and non-linear. In the previous work, a Single Layer Neural Network (SLN) was used for classifying the signals obtained from the optical fibre (Wass et al., 2016; Hunt et al., 2008) for the purpose of comparison. The results showed that the linear SVM outperformed SLN. In my research, I have used both linear and non-linear SVM for classification, though the focus is still on the linear one due to the difficulty with any hardware implementation of a non-linear one.

4.3.1 Linear Classification

A linear separator can be used as a classification method to classify a data point $\mathbf{x}_*$ to a specific group. Let's consider a simple classification problem with only two classes. The data point is denoted as $\mathbf{x} = [x_1, x_2, ..., x_d]$, where d is the number of input features.

Dot product

Classifying data using a linear classifier relies on the dot product as follows:

$$f(\mathbf{x}) = sign(\mathbf{w}^{\mathbf{T}}.\ \ \mathbf{x} + b) \tag{4.1}$$

where $\mathbf{w}$ denotes weight vector, and b bias. Once the best separator has been found it is straightforward to predict the class of a new data point by computing which side of the separator the point lies on (see Figure 4.8). All points on one side of the plane have a dot product less than 0; all points on the other side have a positive dot product value. Then, a new data point $\mathbf{x_{new}}$ can be classified into one of the two classes based on the following:

$$\mathbf{x_{new}} \ \in \ Class\,1 \ \ if \ \ \mathbf{w}^{\mathbf{T}}.\,\mathbf{x_{new}} + b < 0 \tag{4.2}$$

$$\mathbf{x_{new}} \ \in \ Class\,2 \ \ if \ \ \mathbf{w}^{\mathbf{T}}.\,\mathbf{x_{new}} + b > 0 \tag{4.3}$$

Maximum margin

Figure 4.9 shows an example of data that consists of two classes. The data can be divided using a linear separator as it is shown in Figure 4.10, and this separator, which is called a hyperplane, can be placed any of the ways shown in Figure 4.11. The idea of the maximum margin classification is to maximize the distance between the hyperplane and the nearest data points in each class. Figure 4.12 shows the margin and Figure 4.13 shows the maximum margin that defines the best separator between the two classes. The data inputs that push up against the margin are called Support Vectors as shown in Figure 4.14.

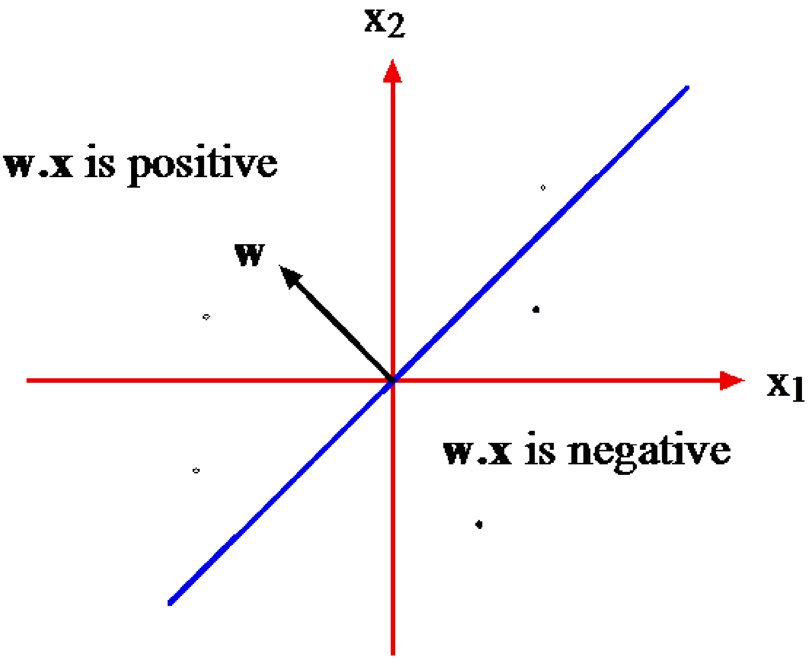


Figure 4.8: Calculate the linear SVM classifier.

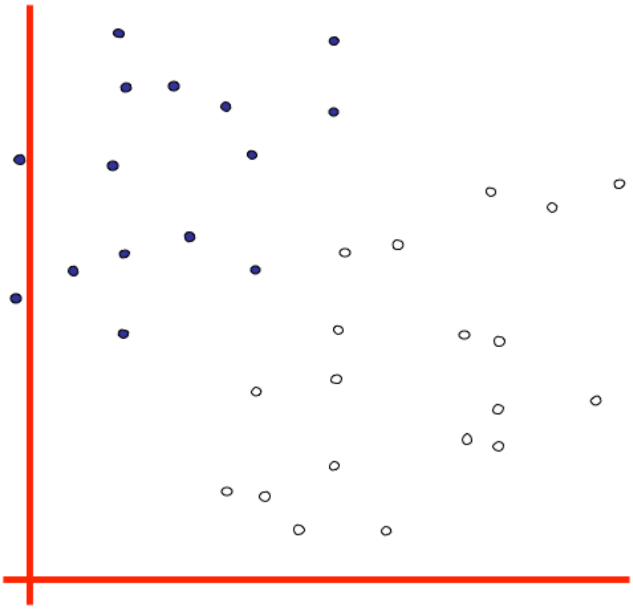


Figure 4.9: An example of data that is needed to be classified.

Generalizing the maximum margin performance

Figure 4.15 presents another example of data that can be classified into two classes using a linear classifier. As we can see in this figure, the maximum margin hyperplane is being determined by one support vector that could be an outlier or simply an inaccurate reading. Therefore, the better

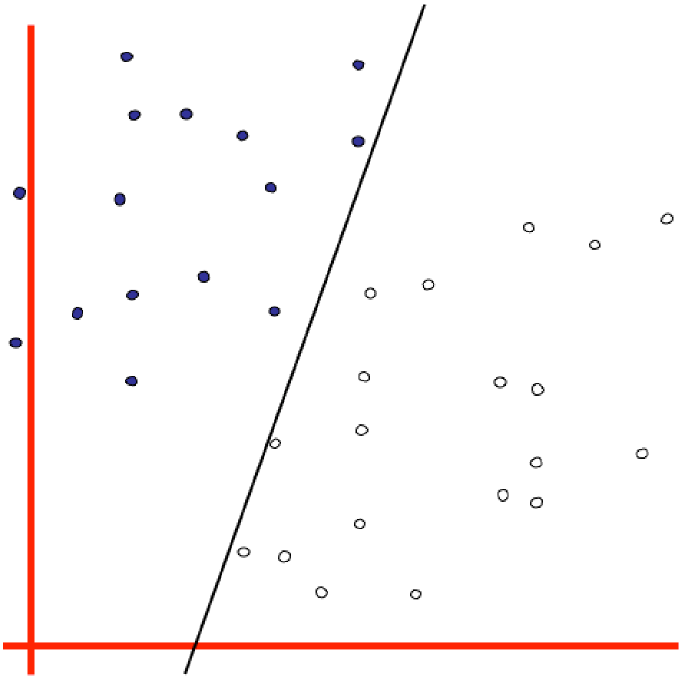


Figure 4.10: Separate the linear separable data.

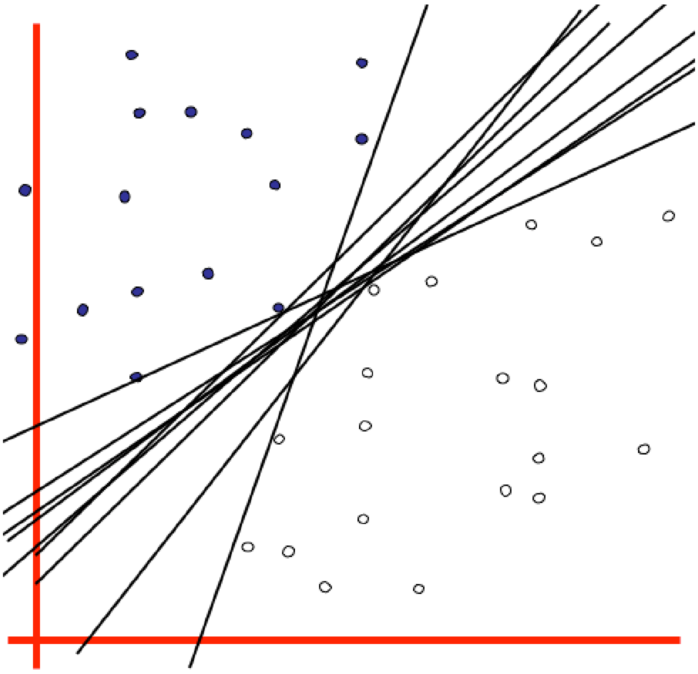


Figure 4.11: The separable data can be separated using many separators.

separator should be as it is shown in Figure 4.16 which is seen to be more concerned with the width of the margin than a single mis-classification.

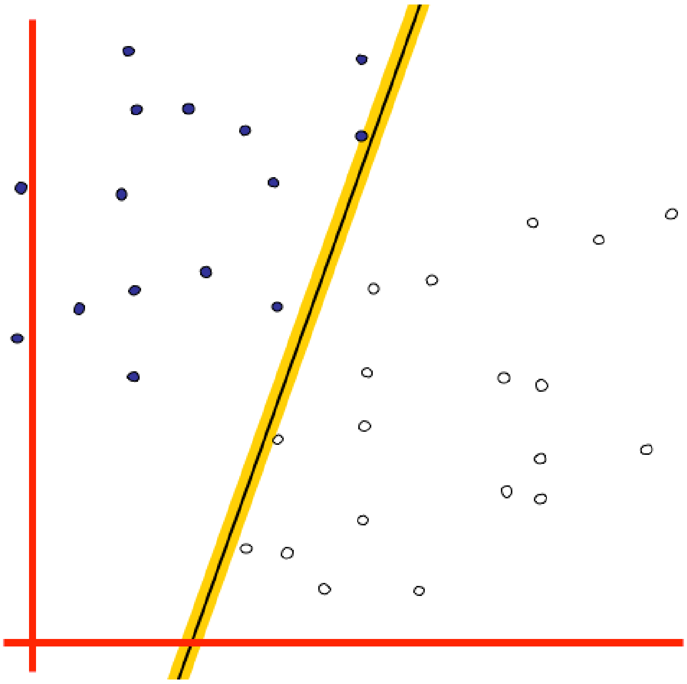


Figure 4.12: Separate data using a margin.

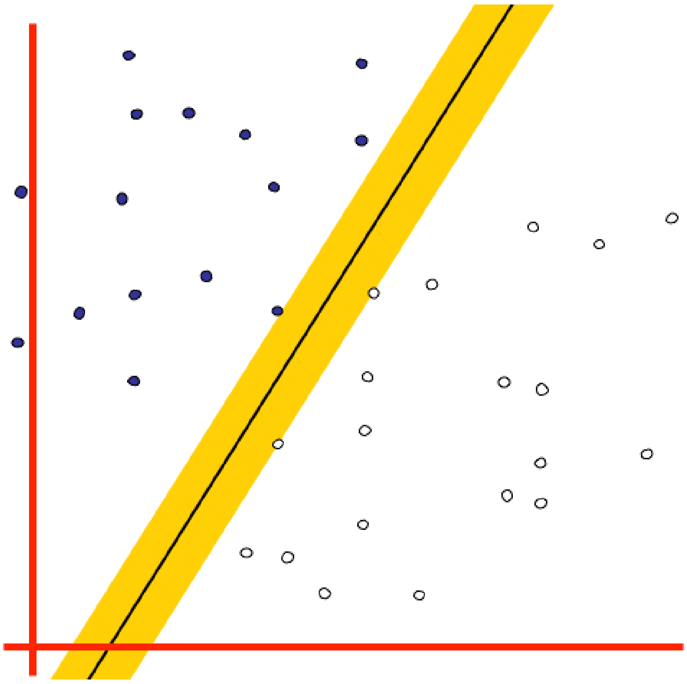


Figure 4.13: Separate data using the maximum margin.

4.3.2 Linear SVM and its Margin

A linear SVM has only one non-learnable parameter, which is the cost parameter C (Smola and Schölkopf, 1998). This parameter allows the cost of mis-classification to be specified. If C is 0

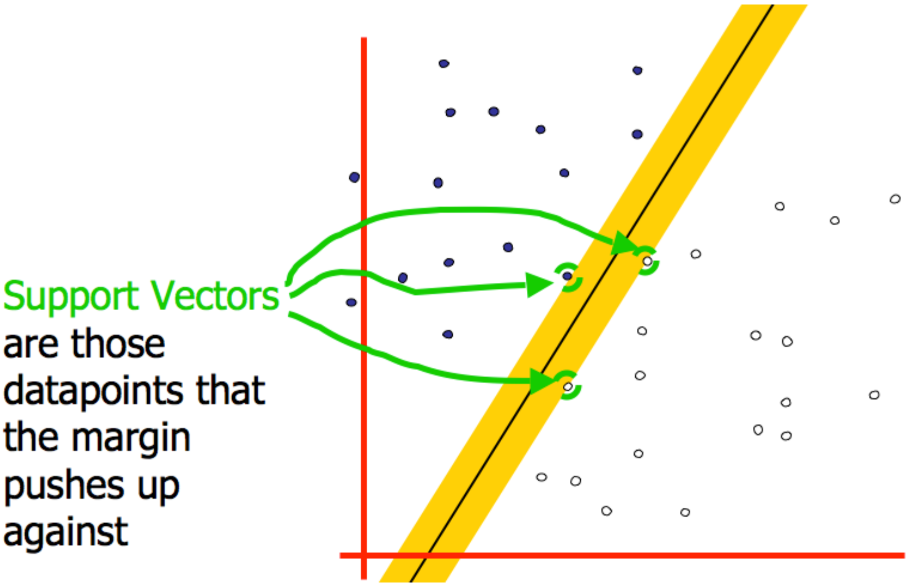


Figure 4.14: Support vectors that determine the maximum margin.

then all the classifier cares about is the width of the margin. If C is large (say 10,000) then mis-classifications are very costly. The Linear SVM model is trained on a set of training data; the training data are linearly separable by a margin and categorized into groups. Each input data sample is tested against the margin while the model tries to maximize the margin as much as possible (Wang et al., 2012).

Separating the two classes without error using the decision separator is called as hard margin SVM, as it can be seen in Figure 4.15. Allowing the classifier to mis-classify data points leads to a margin with a better generalization performance, which can deal with the unseen data much better later. The latter one is called the soft margin (see Figure 4.16).

Figure 4.17 shows an example of controlling the width of the maximum margin using the cost parameter C. The data is linearly separable. It shows how the soft-margin SVM works with a set of different values of C. As we can see from Figure 4.17, if C is large ($C = 100$), the soft-margin SVM works as the hard-margin SVM. It leads to the resultant decision boundary classifying the training data 100% correctly. As a result, the margin will be very narrow which generates a lower level of classification performance, that results in what is called over-fitting the data. When the cost value is small ($C = 3$), a larger margin is produced, and some data points fall within the margin. Furthermore, if you use a smaller value of 1 for the cost parameter, the margin is further maximized which causes under-fitting, that is the classifier can not classify the training data properly. Hence, the middle panel in Figure 4.17 is expected to provide a better generalization behaviour than the others.

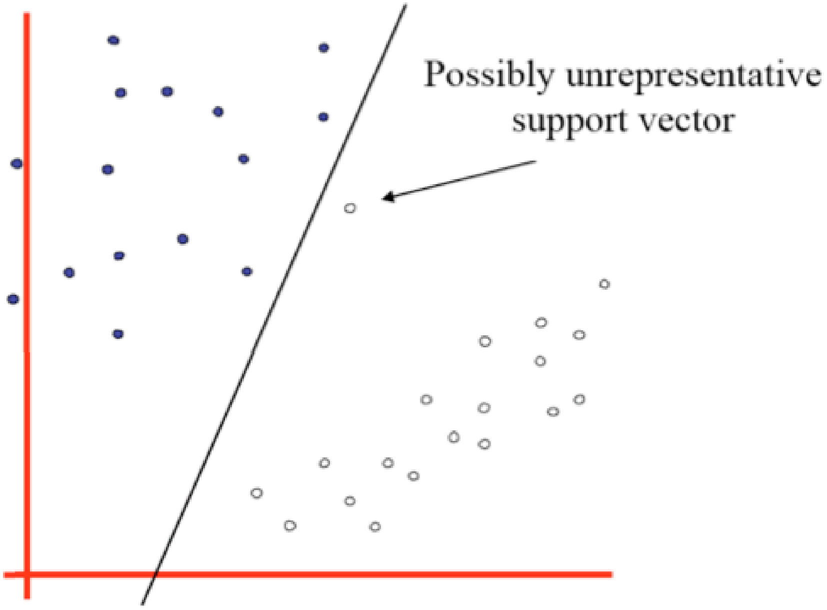


Figure 4.15: An unrepresentative support vector.

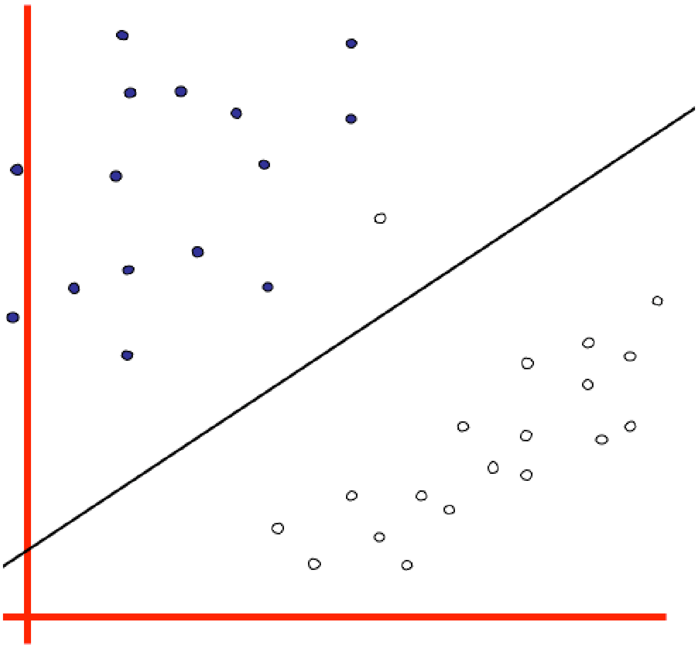


Figure 4.16: The outlier data point.

4.3.3 Non-linear SVM (Feature space)

For the non-linearly separable data as shown in Figure 4.18, the line or the hyperplane cannot be used to separate the data. To solve this problem, a non-linear SVMs should be used instead of a linear SVMs for the classification. The process of classifying the non-linearly separable training data

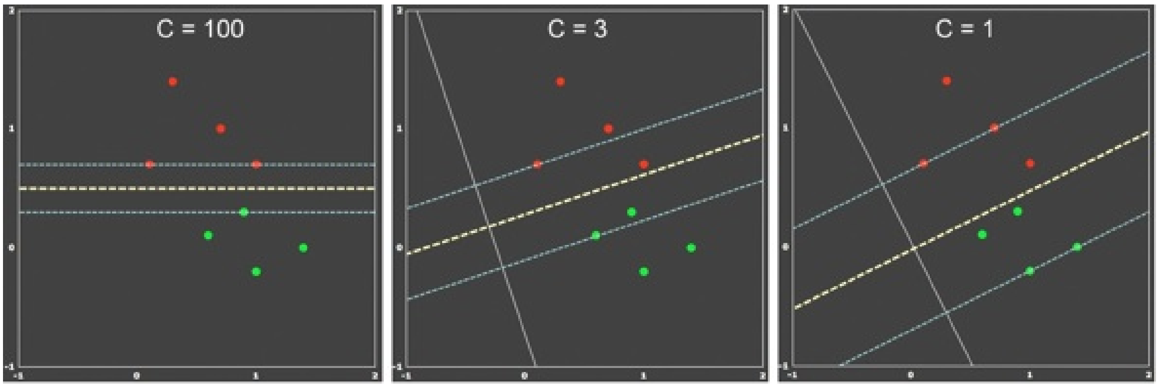


Figure 4.17: Controlling the width of the maximum margin to obtain the optimal generalization of the linear SVM classifier using the cost parameter C (BrainVoyager QX, 2015). In the left panel, when C value is big, the margin is narrow and it works as a hard margin. In the right panel, when C value is very small, the margin is very big and causes under-fitting. The middle panel is an example of providing better generalization.

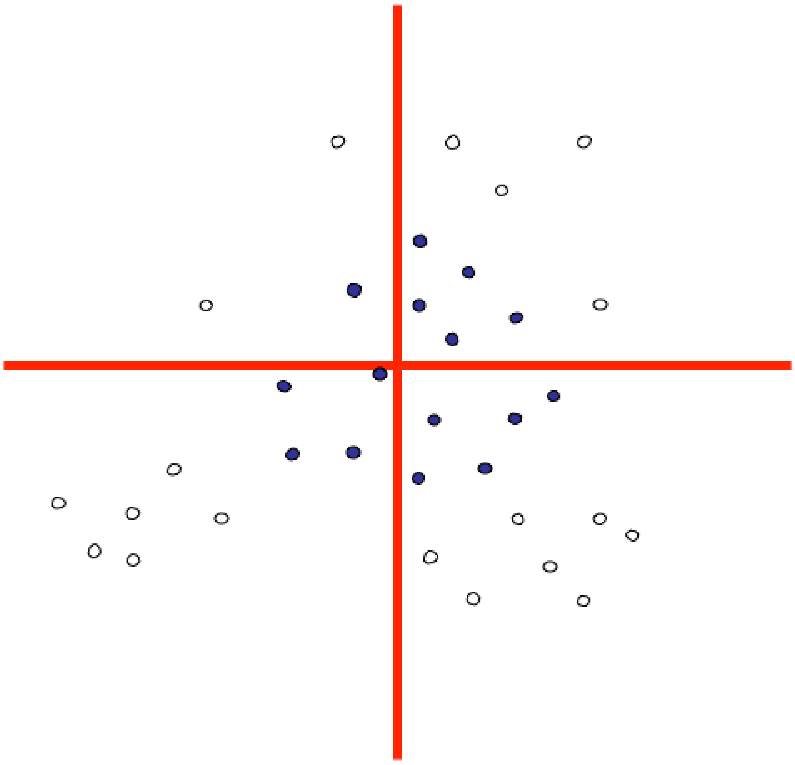


Figure 4.18: Non-linear separable data in the original space.

using a non-linear SVM involves two steps:

1. Projecting the data into a feature space where the data can be linearly separated;

2. finding the hyperplane of maximal margin in the feature space.

As an example, in order to separate the data in Figure 4.18, the following steps have been taken:

1. The data is mapped from the original space in (x, y) coordinates into a different coordinate system (feature space) that is the polar coordinates (r, θ), where r is the distance of the data point from the centre (0,0), and θ is the angle that the data point makes with the horizontal axis.

2. The data can then be separated linearly in this feature space, that is one class is nearer to the centre (0, 0) than the other (see Figure 4.19).

3. As shown in Figure 4.19, the separator has a fixed value of r, which means that it represents a circle in the (x, y) coordinates (a non-linear separator in the original space) (see Figure 4.20). The separating hyperplane is a linear classifier in the feature space, but a nonlinear one in the original space.

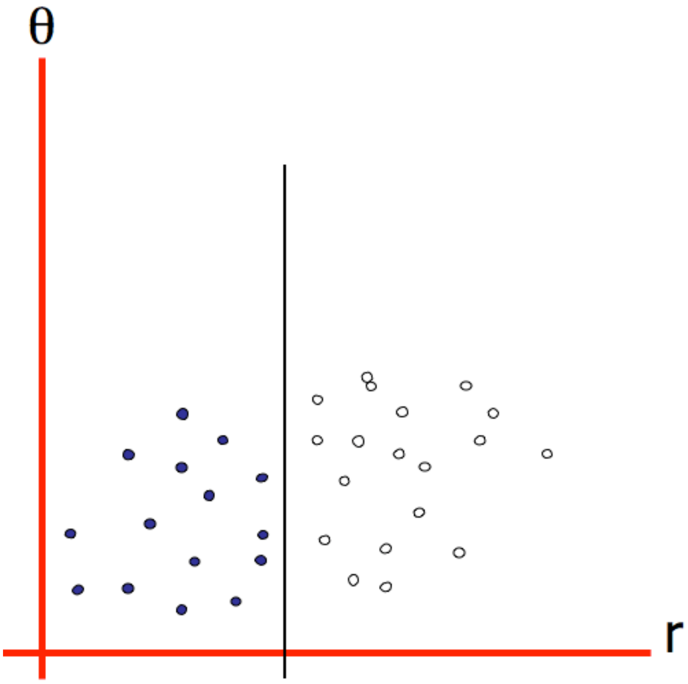


Figure 4.19: Classifying the non-linear separable data after projecting it into a feature space.

Usually, the feature space is a higher dimensional space where an SVM may find a linear separator easily. To explain the idea behind this, let's see the following example. Figure 4.21 shows a two-class data set locating in a one-dimensional space where data can be classified by a single value. The top panel of Figure 4.22 shows another set of two-class data which can not be classified by one single value in the one-dimensional space. For this reason, a new feature y is added and we can use a linear

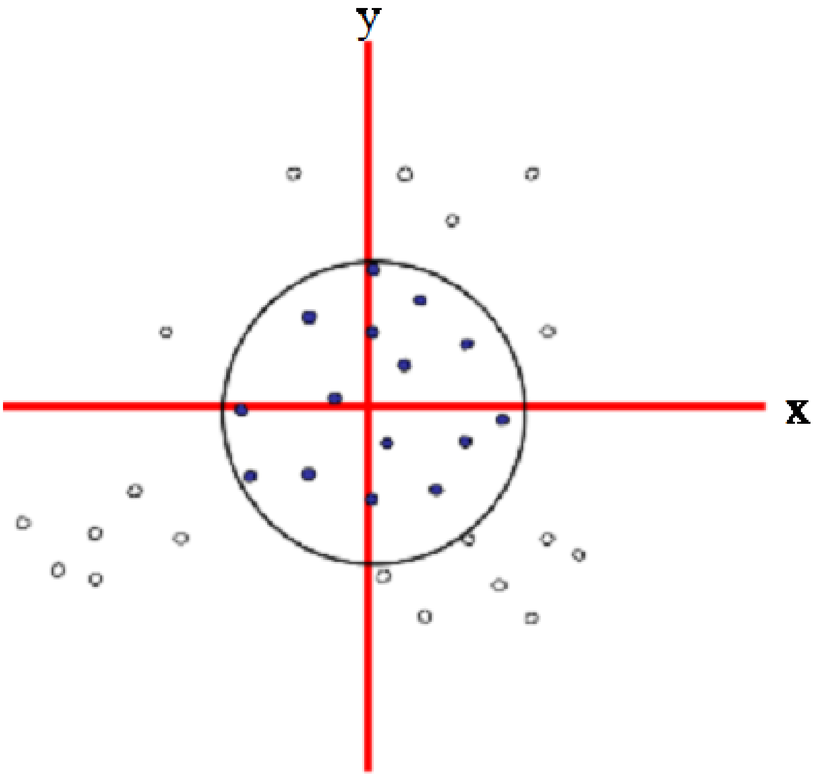


Figure 4.20: The non-linear separator in the original space after classifying the data in the feature space.

classifier in this two-dimensional space. The bottom panel of Figure 4.22 shows how the same data can be classified easily after being mapped into a two-dimensional space using a quadratic function.

One more example can be seen in Figures 4.23, 4.24 and 4.25. Figure 4.23 shows a set of non-separable data in the original 2-dimensional space that need to be classified using an SVM. Figure 4.24 shows the data is mapped into a feature space that is a 3-dimensional space, where a linear separator can be found easily. This is called a non-linear SVM separator in the original space as it is shown in Figure 4.25.

4.3.4 Kernel Trick for Nonlinear Classification

As is mentioned in the previous section, the general idea of using a feature space is that a higher-dimensional space may give more chance to find a linear separator to classify the data.

Let $\mathbf{x}$ denote data in a d-dimensional original space and $\phi(.)$ is a function to project the data from the original space into the high-dimensional feature space. A hyperplane separator in the feature space is defined as follows:

$$f(\mathbf{x}) = sign(\mathbf{w}^{\mathbf{T}}.\ \phi(\mathbf{x}) + b) \tag{4.4}$$

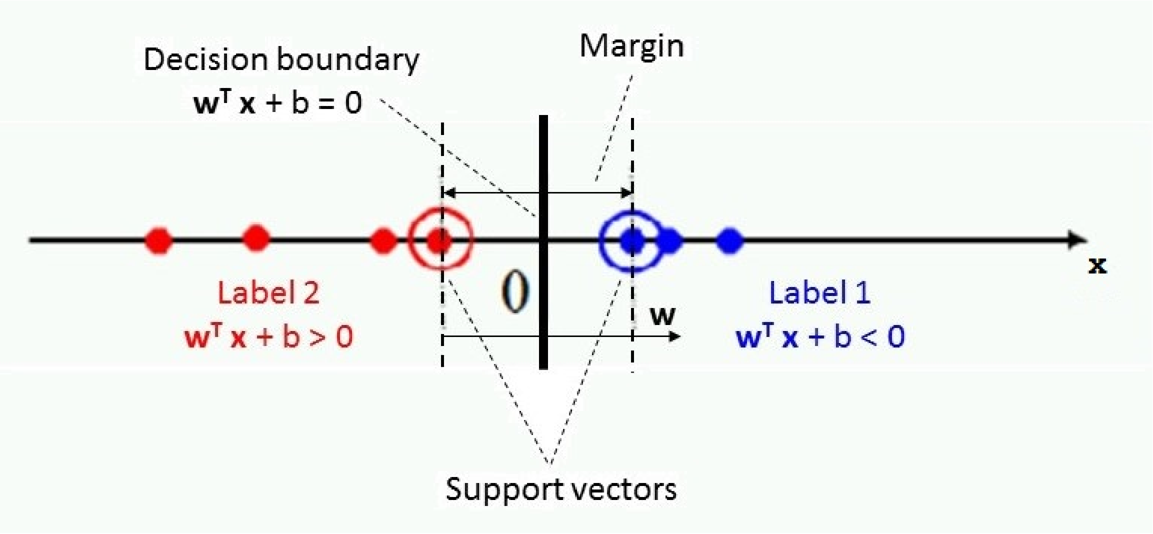


Figure 4.21: Linear separator with data that is linearly separable. A modified figure from (Schütze et al., 2008).

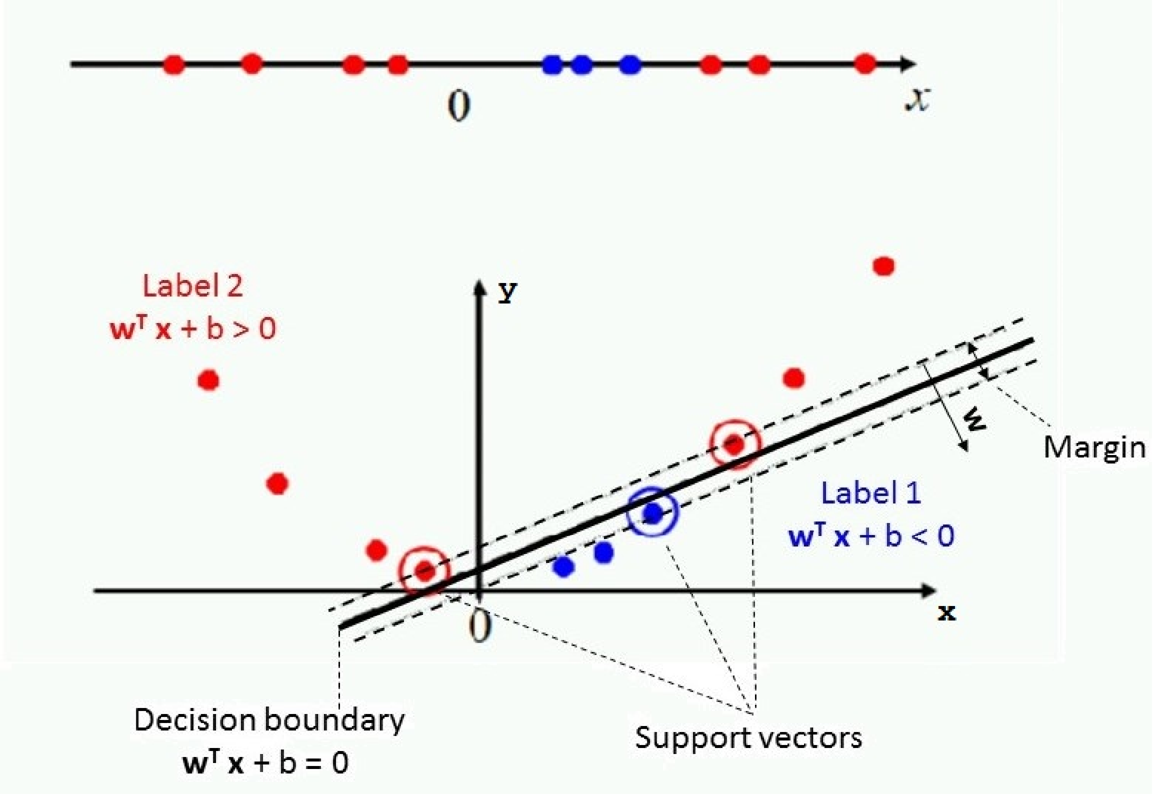


Figure 4.22: Using a linear separator can be possible after the data, which is not linearly separable, is projected into a higher dimensional space. A modified figure from (Schütze et al., 2008).

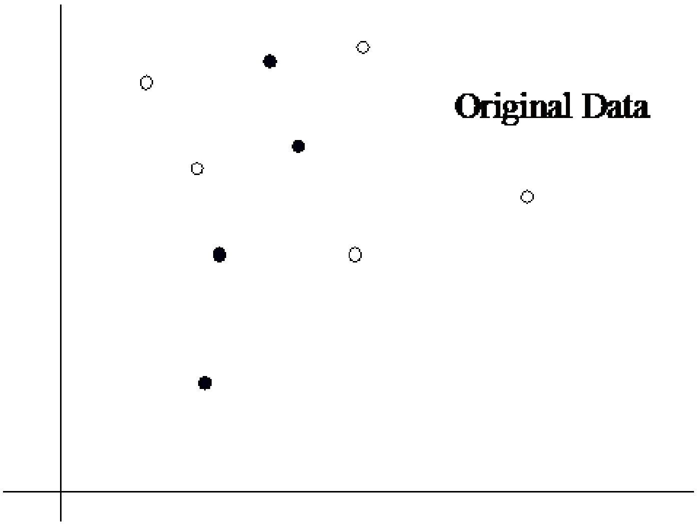


Figure 4.23: Non-linear separable data in original space for which a linear separator can not be found.

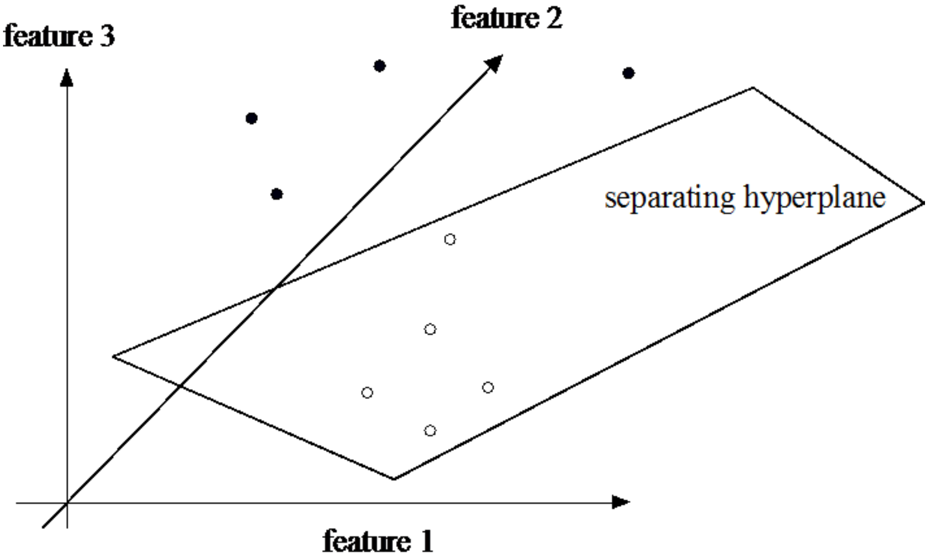


Figure 4.24: Projecting the data into a high-dimensional space/feature space for which a linear separator can be found.

where **w** is the vector of weights, T denotes the transpose, and b is bias. If we have data in a two-dimensional space, (x, y), then any of the following could be features:

- $x + y$
- xy
- $\sqrt{x^2 + y^2}$ (which is r in our previous example, see Section 4.3.3, Figure 4.19)
- or any other combination of x and y.

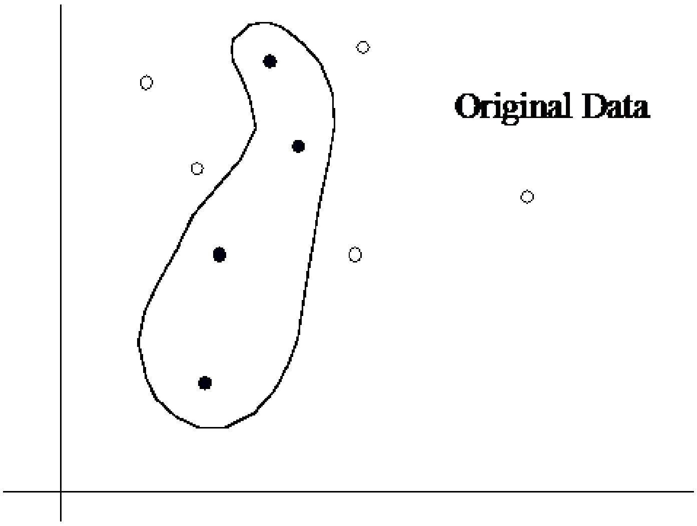


Figure 4.25: The non-linear separator in the original space after classifying the data in the high-dimensional/feature space.

Due to the complexity of calculating the dot product function in the feature space, the kernel trick is used in the input space instead of computing the inner product in the feature space. The kernel function k in the original space can be defined as follows:

$$k(\mathbf{x}, \mathbf{y}) = \phi(\mathbf{x}) \; . \; \phi(\mathbf{y}) \tag{4.5}$$

where $\mathbf{x}$ and $\mathbf{y}$ are two input vectors. A kernel function measures the similarity between two input vectors. There are some popular kernel functions:

- Linear: $k(\mathbf{x}, \mathbf{y}) = \mathbf{x} \; . \; \mathbf{y}$
- Polynomial: $k(\mathbf{x}, \mathbf{y}) = (\mathbf{x} \; . \; \mathbf{y} + \mathbf{1})^{\mathbf{d}}$, where d is the degree of the polynomial
- Radial Basis Function (RBF) (see Section 4.3.5).

The non-linear SVM, which is used for classification in this research, is RBF.

4.3.5 Non-linear SVM, RBF kernel

The Radial Basis Function (RBF) kernel is also called a Gaussian kernel, which is defined as follows:

$$k(\mathbf{x}, \mathbf{y}) = exp(-\gamma \|\mathbf{x} - \mathbf{y}\|^2) \tag{4.6}$$

where $\mathbf{x}$ and $\mathbf{y}$ are two vectors in the original space, γ is a free parameter greater than 0 or equal to $\gamma = 1/2\sigma^2$ (sigma (σ) sets the width of Gaussian distribution). The RBF kernel provides a function that maps vectors into an infinite dimensional space, and measures the similarity between two input vectors by calculating the distance (for example: Euclidian distance) between these vectors. If

$\|\mathbf{x} - \mathbf{y}\|$ is small, the vectors are similar to each other and the corresponding kernel value is relative large. On the other hand, if γ is small, which means the width of Gaussian distribution (σ) is large, then the corresponding kernel value is large.

4.3.6 Model Selection

SVM with a RBF kernel can be trained to effectively classify a data set. Its performance is affected by two parameters, the cost parameter (C), and the kernel parameter (γ). Finding the best values for parameters is called optimization of the model, or fitting the model to the data. If the model cares too little about mis-classifications, it may under fit the data (the value of cost parameter C is very small), or it may over fit the data when it cares too much about mis-classifications (the value of cost parameter C is too big), as mentioned in Section 4.3.2.

Cross Validation

To fit an SVM model for a given data set, the data is divided into a training and a test set (typically it is divided into four fifths training and one fifth test). The training set is used to train the model, and the test set is used to find out how well the trained model performs on data it has never seen before (the model must not see the test data in the training process). The next step is to find the best values of C and γ, which provides the highest classification accuracy. The search process for the best SVM model parameters is done using cross-validation. In the cross-validation process, the training set is divided, with the selection of a cross validation data set (Figure 4.26), that acts as a test set during the training phase (typically five, or ten folds cross validation is used), Figure 4.27 shows a visual representation of 10-fold. Then, a pair of values for C and γ are chosen, and the SVM is trained on the remaining part of the training set. After that, the performance of the trained model is estimated from on the validation set. Then, the search for the best pair of values of C and γ is undertaken. This process is called as grid search. Once the best values have been found the model is retrained on the whole training set, thus the result is given for the unseen (test set).

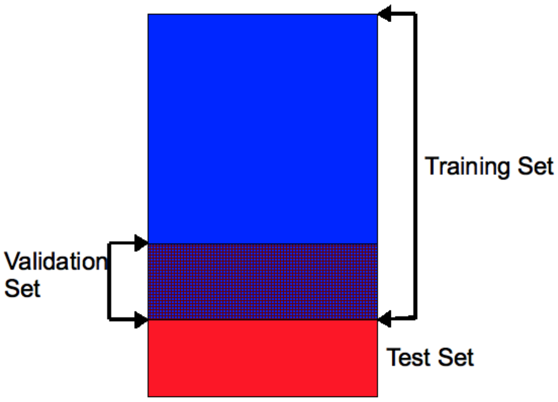


Figure 4.26: Visual Representation of Train/Test Split and Cross Validation.

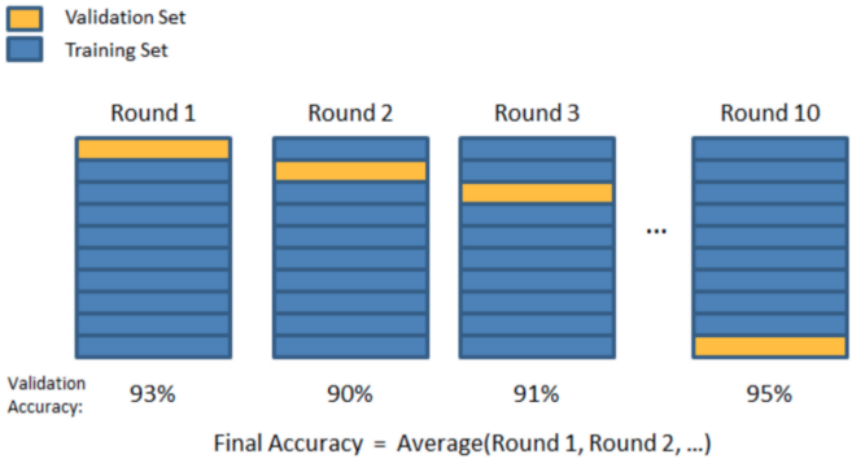


Figure 4.27: Visual representation of K-Folds (Bronshtein, 2017).

4.3.7 Multi-class Classification

I have shown how an SVM can be applied for simple two-class classification problems. When dealing with multi-class problems, there are usually two strategies, which I shall show as follows.

One-against-all

Suppose there are K classes in a data set. The one-against-all strategy constructs K binary base classifiers. Instances from the k^{th} class of the training set are labeled as positive whereas the rest of the instances in the training set are negative. The one-against-all method needs each base classifier to produce a real-valued score rather than a class label. Then each of trained base classifiers gives a prediction to the unseen data. The final predicted label for the unseen data is obtained from the classifier whose confidence score is the highest among all base classifier.

One-against-one

One-against-one strategy constructs $K(K-1)/2$ classifiers, where K is the number of classes and each of which involves training data from two classes, as positive class and a negative class. In the context of linear classification, one-against-one has shown better performance as compared to one-against-all. The better performance of one-against-one comes at the cost (complexity) of $\mathcal{O}(K^2n)$ spaces for storing models and testing instances as compared to complexity of $\mathcal{O}(Kn)$ of one-against-all, where n is the number of data points in the test set, (Huang, 2010). In study (Yuan et al., 2012), it is identified that one-against-one is not ideal (suitable) for large-scale linear classification because it requires large space to store $K(K-1)/2$ classifiers. However, in the context of sparse weight vectors, one-against-one can handle large-scale data for classification.

4.4 The SVM Toolbox for Classification

All my data sets in this study include four classes. I have used the toolbox LIBSVM (versions 3-1.17) (Chang and Lin, 2011) and LIBLINEAR (version 1.94) (Fan et al., 2008) in my study.

LIBLINEAR: it follows the strategy of one-against-all

LIBLINEAR uses the one-against-all strategy and is considered as an efficient multi-class approach for large scale classification as an open source machine learning library with different regularization terms (for example, the supports L2-regularized logistics regression (LR), L2-loss, and L1-loss linear SVM (Fan et al., 2008)) added into the cost function. The LIBLINEAR library has been developed by the Machine Learning Group at National Taiwan University (Fan et al., 2008) and is easy to use for new users.

LIBSVM: it follows the strategy of one-against-one

LIBSVM is an open source machine-learning library, which implements SVM for classification, regression and density estimation. As an integrated toolbox, it has been widely used in different applications such as computer vision, natural language processing, neuroimaging, bioinformatics and so on. It is easy to employ LIBSVM, which mainly involves two steps (Chang and Lin, 2011): 1) training an SVM model using a training data set. 2) using the trained model to predict the information of testing data.

All initial results obtained by using the LIBLINEAR toolbox are worse than the best result obtained using the LIBSVM toolbox with the first type of data, which are shown in Appendix B.1. So, I have decided to use the LIBSVM toolbox in the rest of my experiments.

4.5 SVM Experimental Set-up

As mentioned before, each symbol in my optical transmission dataset has 64 samples, which are represented by complex numbers. The samples, which are used as an input vector to the SVM classifier, are sin, cos of the phase of the samples and the amplitude of samples. As a result, each symbol is represented by two types: 1) the number of samples ×2: when each sample is converted into $(\sin\theta, \cos\theta)$. 2) the number of sample ×3: when each sample is converted into $(\sin\theta, \cos\theta, Amplitude)$. For example, when the complete set of samples of each signal (64 samples) are used as an input vector to the classifier, the number of the samples for each data point (symbol) is 128 (64 ×2) using the sample representation $(\sin\theta, \cos\theta)$; and it is considered as 192 (64 ×3) using the sample representation $(\sin\theta, \cos\theta, Amplitude)$.

I have divided each data set into two-thirds $(\frac{2}{3})$ of the data points as a training set and one-third $(\frac{1}{3})$ as a test set. Each of the training set and test data set are written in a LIBSVM format file with their corresponding labels. Then, the SVM model is trained using the training data set. I have used the default range values to undertake a grid search to search for the most suitable parameters when using the non-linear SVM (Radial Basis Function, RBF) kernel.

4.6 Performance Measurements

For each experiment, the average of Symbol Error Ratio (SER), and the average of Bit Error Ratio (BER) over all data sets are computed. Also, p-values from *t*-tests, which measures the significance of the improvement over the threshold method, are reported.

The SER is defined by the following equation:

$$SER = \frac{NES}{NS} \tag{4.7}$$

where NES is the number of symbol errors and NS is the total number of symbols, respectively, in each test set.

The BER is computed by the following equation:

$$BER = \frac{NEB}{NB} \tag{4.8}$$

where NEB is the number of bit errors and NB is the total number of bits, respectively, in each test set.

Note that my data is balanced, therefore I do not need to use a confusion matrix, and using the bit and symbol error ratio is not a problem. All the results of SER and BER are multiplied by ($\times 10^{-4}$) to facilitate reading the results easily in tables.

4.7 Conclusion

This chapter describes in detail the SVMs that I consider are the best for classification in my research. I have used both linear and non-linear SVM for classification. But I focus on the linear one because it is easier to produce a fast hardware implementation of a linear one. A non-linear SVM can be implemented in hardware using a field-programmable gate array (FPGA), but such implementations may not be fast enough for my application, at least they were not at the point in my study when the decision was made. There is no concrete evidence that they are fast enough even now at the time of writing this.

Chapter 5

Initial Study, Results for the First Type of Modulated Data

This chapter describes an initial investigation of the methodology that will be used in subsequent experiments as reported in chapters 6, 7 and 8. It presents all the initial experimental results that have been obtained throughout this research using an SVM on the first type of optical transmission data. Both linear and non-linear SVMs are applied on the first type of data of optical transmission signals, which is generated (simulated) by a numerical model with a typical return-to-zero (RZ)-QPSK transmitter. As is mentioned in Chapter 3, Section 3.1.2.1, the first type of data consists of 50 data sets, and each data set consists of 32,768 data points. Each data set is divided into two-thirds (22,000 data points) as a training set, and one-third (10,768 data points) as a test set in the SVM experiments. This type of data only has values after the signal has travelled 3,000 km, so all the results are only for this distance. A summary of the experiments that are described in this chapter are shown in the following:

Experiment A: Linear SVM with different number of samples from each symbol, section 5.1.
Experiment B: Non-Linear SVM with a Gaussian kernel (RBF), section 5.2.
Experiment C: Linear SVM using neighbouring information, section 5.3.

In this work, I have decided to use a linear SVM for classification using two strategies: one-against-one and one-against-all to see which toolbox is the best. Appendix B.1 shows how I have made my decision. Table 5.1 shows all the SVM results that are compared with the results obtained using the traditional method: the threshold method. This method uses a hard threshold value. The value of the signal at the centre of a symbol is compared to this threshold to determine the phase of the received symbol, more details in Chapter 4, Section 4.1. Table 5.1 presents the Symbol and Bit Error Ratio (SER and BER), which are averages over 50 data sets. Also, it shows the statistical test results (p-value) of t-test that measures the probability under the assumption of no difference between the two methods' performance: the threshold method and the Support Vector Machine (SVM) method.

5.1 Experiment A: A Linear SVM

In this experiment, an investigation is undertaken by applying a linear SVM classifier using a different number of samples from each optical transmission symbol. This is to see whether it is possible to improve the classification performance above that of the threshold method by using an SVM and also using more values of the signal than just the central one. I have used four different input vectors to the linear SVM classifier:

1. The central sample (1×2 is when the optical transmission symbol is represented as $\sin\theta$ and $\cos\theta$), and (1×3 is when the optical transmission symbol is represented as $\sin\theta$, $\cos\theta$ and amplitude).

2. The complete set of samples (all 64 of the available sample points on the curve) of the symbol (64×2 is when the optical transmission symbol is represented as $\sin\theta$ and $\cos\theta$), and (64×3 is when the optical transmission symbol is represented as $\sin\theta$, $\cos\theta$ and amplitude).

Two thirds of the symbols in each data set are used to train the linear SVM model as a training set, and the rest of the signals are used for testing the model. The SVM model is trained using different values of the C parameter (0.01, 0.03, 0.5, 0.06, 0.13, 0.25, 1, 8, 64, 256, 512, 600, 800, 1000, 1028 and 2048).

The linear SVM results reveal that just using the central sample of the symbol as an input to the linear SVM classifier gives similar results to the threshold method result. This is not really surprising and p-values of t-test provide strong evidence that the observed slight difference between both the linear SVM classifier using only the central sample and the threshold method is likely to be due to chance, since the p-values are 0.56 and 0.42. However, using the complete set of samples of the symbol shows a significant improvement over the threshold method, where the BER is reduced to 28.09×10^{-4}, see Table 5.1. This is clear since the p-value of the t-test provides strong evidence that the observed difference between the threshold method and linear SVM using the complete set of samples of the symbol is unlikely to be due to the chance (the p-value is 1.28E-32). This is because we now have a higher dimensional input vector and a linear SVM usually works better in such a high-dimensional space. For example, the hyper-plane in two dimensional space is a line, in three dimensional space is a plane, but in more than three dimensional space is a hyper-plane, which splits the space into too many dimensional sub-spaces depending on the number of dimensions (see Figure 5.1). Consequently, using 128 features (64×2 for the complete set of samples of the symbol) as an input to the linear SVM classifier gives much better result than using just two features (1×2 for the central sample of the symbol), which does not give any improvement over the threshold method in this experiment.

Method	**Kernel**	**No. of symbols**	**No. of samples**	**No. of features**	**SER** $(\times 10^{-4})$	**BER** $(\times 10^{-4})$	**P-Value**[1]
Threshold	-	One	Central	-	121.27±26.16	60.9±13.21	
SVM	Linear	One	**Whole**	64 × 2	**55.78±13.87**	**28.09±6.99**	**1.28E-32**
			Whole	64 × 3	58.38±14.44	29.36±6.24	1.99E-31
			Central	1 × 2	121.06±25.24	60.8±12.75	0.56
			Central	1 × 3	121.6±25.33	61.07±12.79	0.42
	RBF	One	Whole	64 × 2	55.79±15.03	28.13±7.7	7.76E-36
			Central	1 × 2	121.06±25.35	60.80±12.81	0.52
	Linear	Three	128	128 × 2	57.07±13.31	29.05±6.71	4.69E-32
			192	192 × 2	60.45±13.31	30.44±6.73	1.86E-31

Table 5.1: SVM results using the first type of data at the distance of 3,000 km. All the results are averages over 50 data sets, and compared with the threshold method result. (×2) denotes that each an optical transmission symbol input is represented by ($\sin\theta$,$\cos\theta$). (×3) denotes that each an optical transmission symbol input is represented by ($\sin\theta$,$\cos\theta$, *amplitude*). 128 samples from three symbols means that 32 samples of the preceding (from the 33rd sample to the 64th sample), 64 samples of target (The complete set of samples) and 32 samples of succeeding (from the first sample to the 32nd sample) symbols are used as an input. 192 samples from three symbols means that the complete set of samples of the preceding, target and succeeding symbols are used as an input. Using only the central sample of the symbol with SVM gives a similar result to the threshold method (the text in blue colour). Using the complete set of the samples of the symbol as ($\sin\theta$,$\cos\theta$) provides the best result (the text in red colour). Using the amplitude and the neighbouring information does not improve the result.

Moreover, adding the amplitude of the optical transmission symbol with the phase value as an input to the classifier (giving 192 features since I am still using as $\sin\theta$ and $\cos\theta$ values) shows a little bit worse result than the result obtained using just the phase, even though it is still much better than the threshold method result. This might be because adding the amplitude value adds some conflicting informative features to the classifier, which affects negatively on the accuracy of the classification (Destrero et al., 2009). Consequently the rest of the results in the chapter do not use the amplitude value.

[1] The P-value approach involves determining "likely" or "unlikely" by determining the probability (assuming the null hypothesis were true) of observing a more extreme test statistic in the direction of the alternative hypothesis than the one observed (The Pennsylvania State University, 2018). In my research, the null hypothesis represents that the difference between the results is likely to be by chance. For example, if the P-value is small, less than (or equal to) 0.05, then the difference between the results is unlikely to be by chance; then the null hypothesis is rejected. And, if the P-value is large, for example, more than 0.05, then the difference between the results is likely to be by chance; and the null hypothesis is accepted.

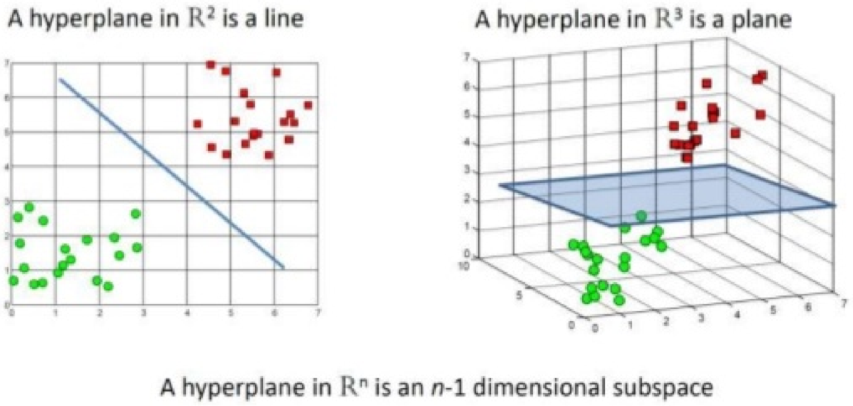


Figure 5.1: A hyperplane in $\mathbb{R}^n$ (KDAG, 2015).

To conclude this section, it can be seen that using a linear SVM using on the complete symbol's set of samples as an input to the classifier gives much better classification result than the traditional threshold method. In the next section, I intend to investigate the non-linear SVM classifier to see whether or not it can improve the BER.

5.2 Experiment B: Non-linear SVM

In this section, a non-linear SVM classifier is used with the first type of data. A non-linear SVM applies a particular transformation to the input vector(depending on the non-linear kernel being used) before it tries to use a linear method for separation. In this experiment, I use a Gaussian (Radial Basis Function/RBF) kernel, which usually give good classification accuracy. This section investigates whether or not using the non-linear SVM kernel can further improve the classification accuracy more than the linear SVM kernel. Since the best result for using the linear SVM is obtained when using the complete symbol's set of samples (128 features as $\sin\theta$ and $\cos\theta$). I use the same input vector to train the non-linear SVM model.

In the previous section, I used a range of $\log 2(C)$ values with the linear SVM from the minimum (-6.6) to the maximum (11). Herein, I use the default range values of cost(C) and scaling(γ) parameters in the toolbox to undertake a grid search to tune the parameters for the RBF kernel, which means that I investigate a large range of values for the parameters $\log 2(C)$ and $\log 2(\gamma)$. During this training process, the training data is divided into 5 parts, using cross validation, to find a suitable pair of values for C and γ. The default range values of cost(C) and scaling(γ) are shown in Table 5.2.

As we can see from Table 5.1, the non-linear SVM gives a similar result to the linear SVM classifier. It seems that the data is mostly linearly separable, so the non-linear SVM classifier does not improve the result much; even after I extend the range of $\log 2(C)$ and $\log 2(\gamma)$, Table 5.2.

Parameter	Minimum value	Maximum value	Step
Default range of parameters' values			
$\log 2(C)$	-5	15	2
$\log 2(\gamma)$	-15	3	-2
Extending the range of parameters' values			
$\log 2(C)$	-3	20	2
$\log 2(\gamma)$	-20	-2	-2

Table 5.2: The default values of the parameters that are used to train the non-linear SVM model on the complete set of samples from one symbol. (C) is the cost parameter, and (γ) is Scaling parameter.

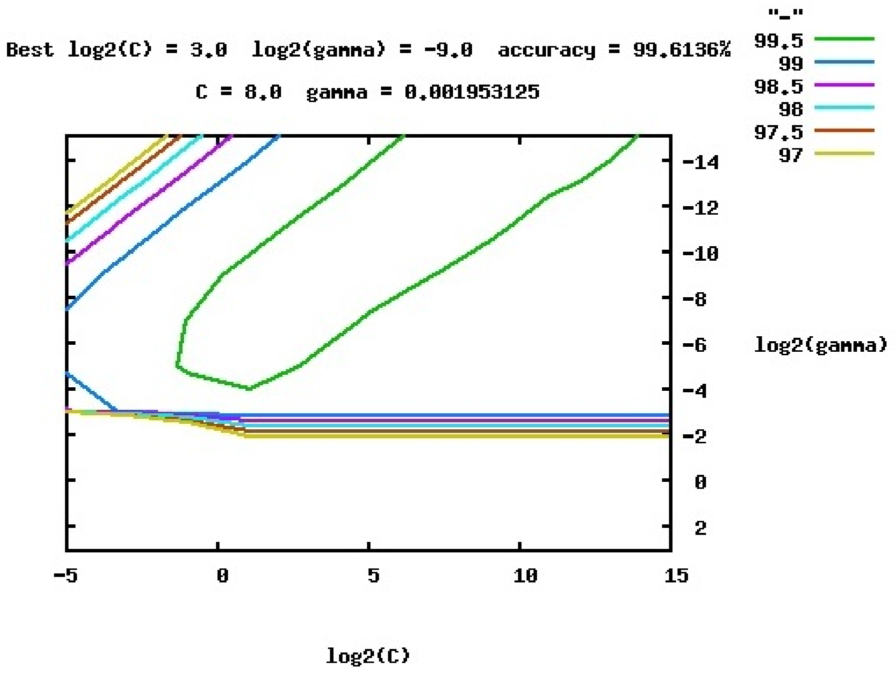


Figure 5.2: Selecting the best values of C and γ using cross-validation. The green area in the console plot shows the range that has the best values of C and γ for providing the highest classification accuracy rate.

Figure 5.2 shows the area that has the best parameters for the RBF kernel to provide the highest classification accuracy. As is shown, this area is not completely surrounded, which is why I decided to extend the γ value to be much smaller than that used previously (the default values), (see Table 5.2). These results can be seen in Appendix B.3. Since even extending γ values still gives a similar result to the linear SVM classifier, I decided to stop at this point and focused on the linear SVM, since the non-linear SVM provides a similar result. Also, crucially, the linear SVM classifier would be much easier to be build into the hardware.

In next section I aim to investigate using the linear SVM with the neighbouring information to see whether it can improve the BER more than using information from just one optical transmission symbol as an input vector.

5.3 Experiment C: Using Neighbouring Information

Since the previous experiment reveals that the linear and non-linear SVM provide similar results, and building the linear SVM classifier in the hardware would be easier, this section, only investigates the use of a linear SVM classifier. Here it is used to investigate the effect of the neighbouring information on improving the classification process when it is used as an input vector to the classifier. The linear SVM classifier is applied for the classification process using different values of the cost parameter C, and the best results are given here. This section presents two experiments using the linear SVM based on two different input vectors: 1) using the target symbol, and half of each symbol on either side. 2) using the target symbol, and the whole symbol either side.

5.3.1 Using 128 samples from 3 neighbouring symbols

In this experiment, each input signal has 128 samples (256 features (as $\sin\theta$ and $\cos\theta$, see Chapter 3, Section 3.3.1)), which are the complete set of samples of the target symbol and 32 samples from the symbol either side of the target. In other words, each data input consists of 64 samples from the symbol being decoded, a range of samples from the 33^{rd} sample to 64^{th} from the preceding symbol, and a range of samples from the 1^{st} sample to sample 32^{nd} from the succeeding symbol. See Figure 5.3.

As it is shown in Table 5.1, using the target symbol and half symbol either side does not show any improvement compared with using the complete set of samples from only the symbol being decoded. But still the result is much better than the threshold method. It might because the samples of the neighboring waves are not so informative to help the linear SVM classifier learn. Therefore, I extend this to use the target symbol and full symbol either side as an input vector to the linear SVM classifier in next section.

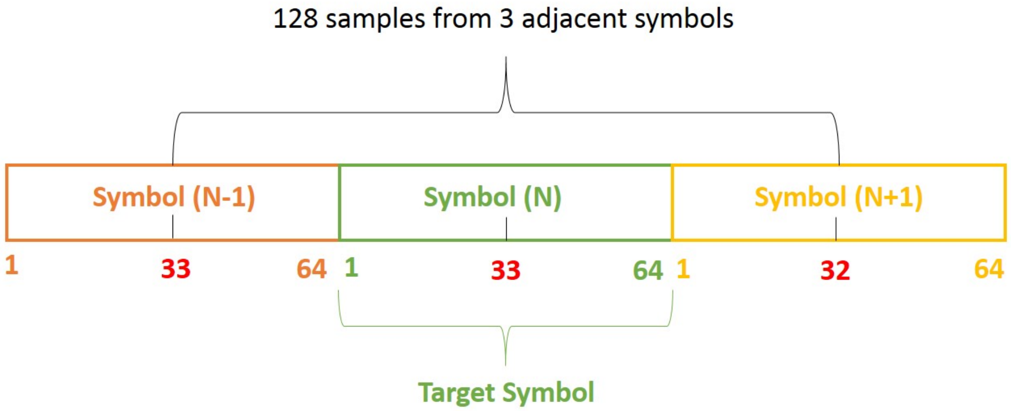


Figure 5.3: Using neighbouring information, the complete set of samples from the target symbol, and 32 samples from symbols on either side. The target symbol is the symbol that is being decoded.

5.3.2 Using the complete set of samples from 3 neighbouring symbols

In this experiment, the input vector signal consists of 192 samples (384 features as ($\sin\theta$ and $\cos\theta$, see Chapter 3, Section 3.3.1)), which are the complete set of samples of the target symbol and both of the symbols either side. As we can see in Table 5.1, using the complete set of samples from three neighbouring symbols as an input to the linear SVM classifier also does not improve the bit error ratio more than using the complete set of samples from only one symbol. It might because of the same reason to that given previously, that not all the neighbouring samples are relevant and useful to be used in training the classifier.

5.4 Conclusion

To conclude this chapter, I have shown that the BER can be improved significantly using the SVM classifier, compared with the threshold method (the method used currently in the hardware). This is because the threshold method is inflexible since it depends on measuring just the phase of the mid-point of the optical transmission symbol, then classify it based on the given threshold values. But the SVM classifier is trained on the examples of the symbol's actual shape to classify the optical transmission symbol. In particular the SVM can use more than one sample from the symbol as an input, which increases the relevant information to train the SVM classifier; and, as shown in this chapter, that improves the classification performance considerably. Interestingly the result obtained using linear SVM classifier with only the central sample of the symbol gives a very similar result to the threshold method despite the training. It is only when extra information is added to the training vector that significant improvements are made.

Furthermore, the non-linear (the Gaussian kernel (RBF)) and the linear SVM classifier give

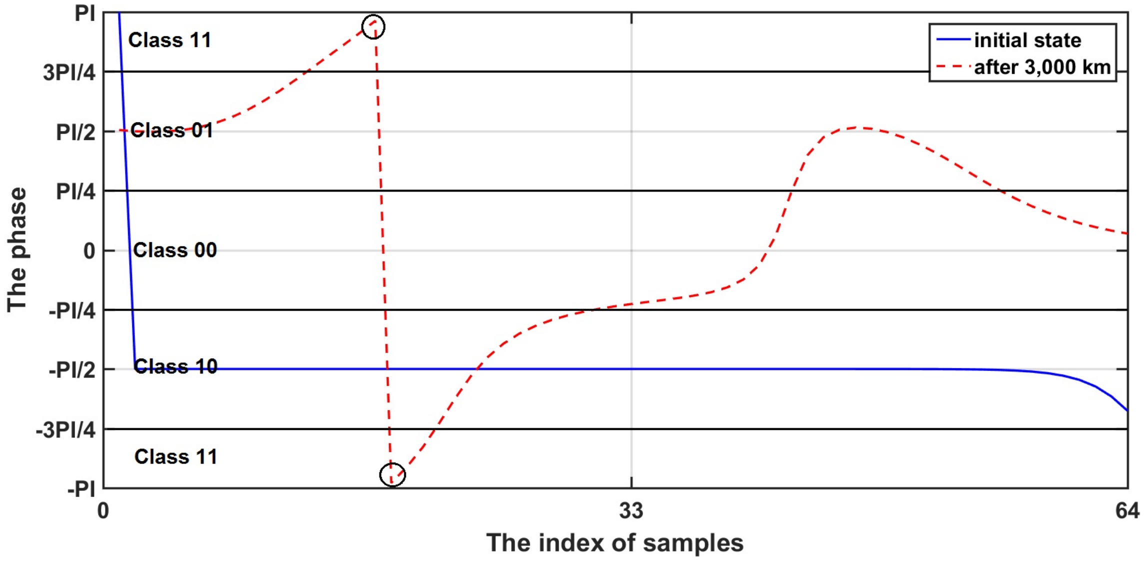


Figure 5.4: An optical transmission symbol from class “10” is classified correctly using the linear SVM where the complete target symbol’s set of samples are used as an input to the classifier, but incorrectly as class “00” using the threshold method, the symbol is received at the distance of 3,000 km. Note: Look carefully about the 33^{rd} sample, the mid-point in the red dotted line is just above the boundary so the threshold method classifies the result as class “00” . However the SVM trained on the complete set of samples of the symbol can classify it as class “10” . The phase values that are denoted by circles are adjacent, they have the same angle values but different signs (referenced in chapter 3, Section 3.3.1). Note that PI denotes π.

very similar results. Moreover, using the neighboring information does not improve the BER more that using just one symbol. This came as a surprise since I thought that further information from neighbouring symbols ought to add information.

Figure 5.4 shows an optical transmission symbol that is classified correctly by the linear SVM classifier using the complete set samples from one symbol, whereas it is mis-classified using the threshold method. As can be seen at the 33^{rd} sample, the mid-point in the red dotted line is just above the boundary. This kind of distortion makes the threshold mis-classify the symbol as class “00” when measuring the given phase value (class “10”) after it is distorted by distance travelled. Whereas the linear SVM is trained on the complete set of samples of the symbol has more information, which makes the SVM classifier able to detect the phase values correctly. Please note that the phase values that are denoted by circles are really adjacent, they have the same angle values but different signs (referenced in Chapter 3, Section 3.3.1). The way the signal is shown is just an artefact of the cyclic nature of phase angles. For completeness Figure 5.5 shows another optical transmission symbol that is mis-classified using both the linear SVM with neighbouring information and the threshold method. The optical transmission symbol in this situation is completely distorted, which can not be classified by the linear SVM classifier, this is called an irretrievable error.

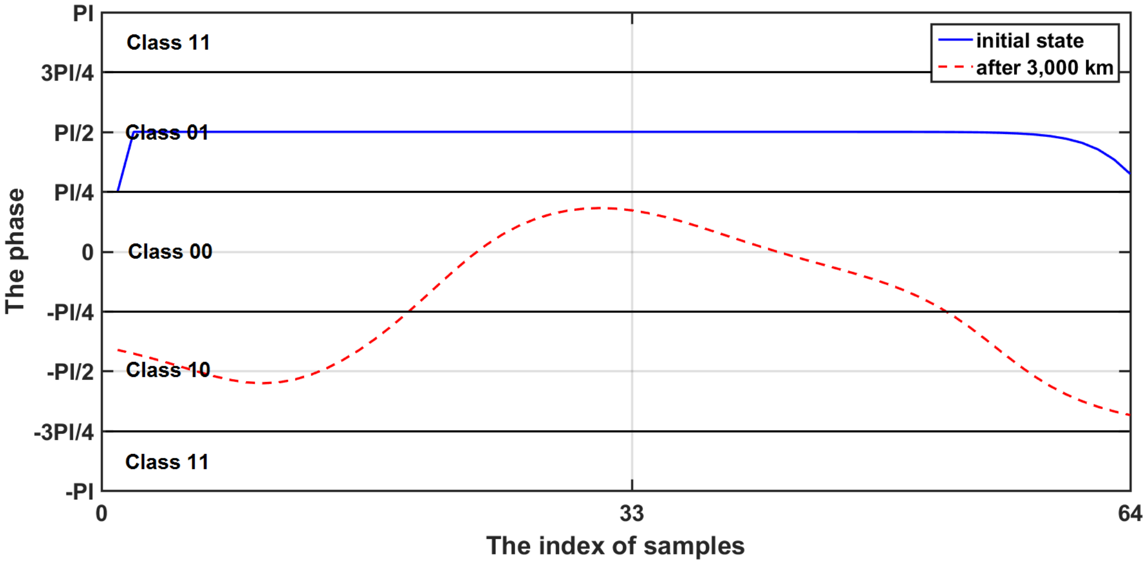


Figure 5.5: An optical transmission symbol from class "01" is classified incorrectly as class "00" using the linear SVM classifier with neighbouring information and also using the threshold method. The symbol is received at 3,000km (the red dotted line), and it is completely distorted, this is called an irretrievable error. Note that PI denotes π.

Having completed this initial investigation there is plenty of evidence that it is worth investigating further. In the next chapter the second type of data is investigated. This data has values at all distances out to 10,000 km and is generated using a more modern and better algorithm. The number of errors found by the threshold method is greatly reduced so the SVM will have a harder task to make such improvements as I have found here. Having found that including the amplitude value does not help I have decided not to investigate this further other evidence from further testing (not given here) justified this decision. The non-linear SVM was also found to have no benefit, so I have not made much further use of it. This is further motivated by wanting to create a method that could be incorporated into the hardware of the transmission line. A linear SVM can definitely be implemented in hardware (Metaxas et al., 2013). Finally I have pursued using values from neighbouring symbols, despite their mediocre performance in this chapter, since other evidence points to their possible ability.

Chapter 6

Results for the Second Type of Modulated Data

In this chapter I use the second type of optical transmission data, namely Non-Return-to-Zero (NRZ) modulated data. This represents probably the most realistic simulated data, which means that the BER is very low and better reflects the character of the real transmission line with the real BER that would be obtained with real world transmitted data ((Nasreen et al., 2018), (Agrawal, 2018) and (Antoniades et al., 2011)). The main focus of my experiments in this chapter is to find out which set of features gives the best results.

This second type of data consists of ten data sets for each distance, from zero km up to 10,000 km in 1,000 km steps, and each data set consists of 32,768 symbols. Again, as described in Chapter 5, to carry out my experiments each data set is divided into two-thirds (22,000 symbols) as a training set, and one-third (10,768 symbols) as a test set. The aim of the experiments in this chapter is to investigate whether or not a linear SVM classifier can help decode the received optical transmission signals (the second type of data) using different input vectors.

A summary of the experiments that are described in this chapter are shown in the following:

Experiment A: To check whether the improvement of the classification I was able to achieve with the first type of data was also achievable with the second type of data, Section 6.1.

Experiment B: Classifier with an input vector containing different number of input samples from each symbol being decoded, Section 6.2.

Experiment C: Classifier using an input vector containing different number of input samples from the target symbol and neighbouring symbols, Section 6.3.

Experiment D: Linear SVM using an input vector containing only the central sample from the target symbol and the central sample from a different number of neighbouring symbols either side, Section 6.4. Figure 6.1 shows an example of using the central sample from seven adjacent symbols as an input to the classifier, in which the red dots represent the central sample of each symbol.

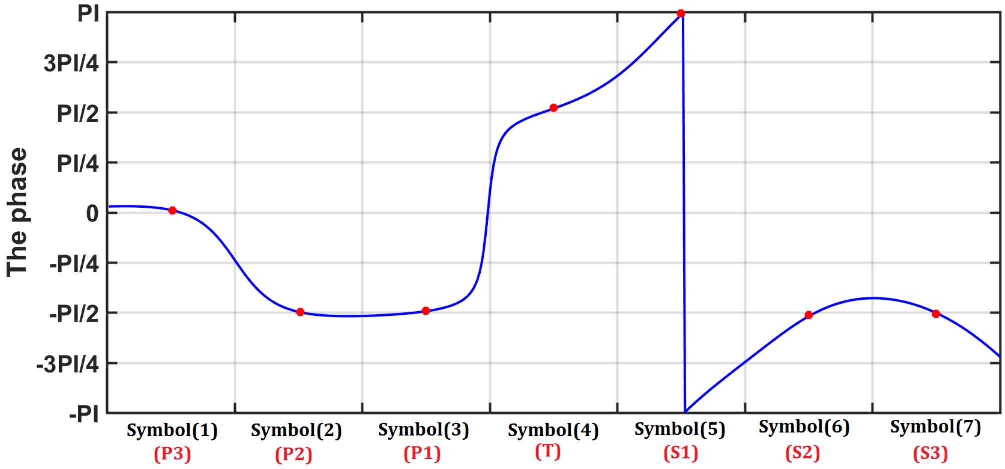


Figure 6.1: An example of using the central sample from seven adjacent symbols as an input to the classifier. The red dots represent the central sample of each symbol. T denotes the target symbol that is being decoded. P denotes the preceding symbol of the target symbol. S denotes the succeeding symbol of the target symbol. Note that PI denotes π.

Finally, Section 6.5 discusses and analyzes the results in the whole chapter. All results in these experiments are compared with the results obtained by applying the threshold method. They are averages over the ten data sets. The results that are obtained from the received symbols at 1,000 km are not shown in this chapter since they do not show any bit errors using both the threshold method and SVM methods.

Before starting with the experiments, I look into the results obtained on a class by class basis using the threshold method. Table 6.1 shows the number of predicted symbols in each class using the threshold method, at the distance of 8,000 km (8,000 km used so that I can compare the results to the third type of data, which only goes as far as 8,000 km). The results in this table are obtained from the first data set of the second type of data. It is shown that the number of symbols that are classified correctly for each class is very high. Also, most of the symbol errors come from predicting the symbol's class as one of the adjacent classes (See Figure 3.1), for example, predicting Class “00” as Class “01” or “10” but less likely to be Class “11”, which results in one-bit error for each mis-classified symbol. If the symbol that is from Class “00” is predicted as “11”, that means two-bit errors. Figure 6.2 shows that the BER is very low, and there are few symbol errors that have two-bit errors (in total 6 symbols).

6.1 Experiment A: Comparison between First and Second Type of Simulated Optical Data

In this experiment, an investigation is undertaken by applying a linear SVM classifier using the complete set of samples as inputs from each optical transmission symbol being decoded. As I said

Symbol	Predicted class			
class	00	01	10	11
00	2528	42	49	2
01	55	2606	1	48
10	61	1	2640	37
11	2	44	50	2602

Table 6.1: The number of predicted symbols in each class. The result is obtained from using the threshold method with the first data set of the second type of data, at a distance of 8,000 km. The test set consists of 10,768 symbols in total. Most of the symbol errors have a one-bit error, where usually the symbol's class is mistakenly determined as one of the adjacent classes. The number of symbol errors that have a two-bit error are very few (in red). The number of symbols that are classified correctly for each class is shown in blue.

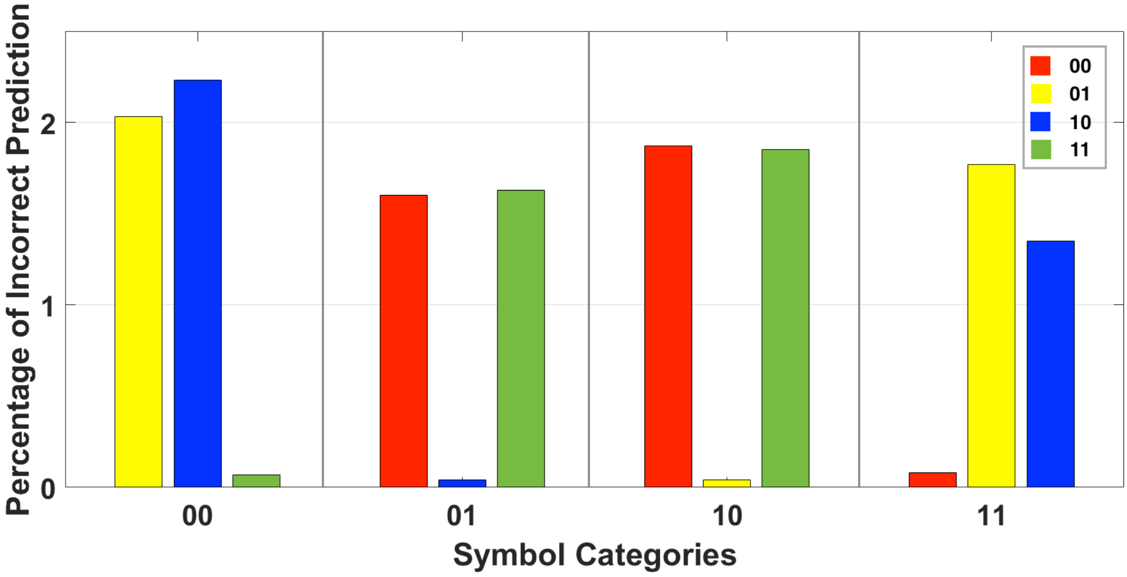


Figure 6.2: The percentage of symbol errors for each class. The result is obtained from using the threshold method with the first data set of the second type of data, at a distance of 8,000 km. The test set consists of 10,768 symbols. Table 6.1 shows the same information in numbers.

earlier, this is to check whether improving the classification I was able to achieve with the first type of data was also achievable with the second type of data. The best result for the first type of data was obtained using the complete set of samples (64 × 2) as input to the SVM. Hence the same method was used here on the second type of data. The only distance used for the first type of data was 3,000 km, so again the same distance was used here.

Table 6.2 shows the results including the Symbol Error Ratio (SER) and Bit Error Ratio (BER). Note that results for the first type of data shown in the table are averages over ten data sets only. Ten is used because I only have ten data sets for the second type of data (results in Table 5.1 are results over all 50 data sets). Also, the SVM model is trained using the C parameter value of 512, which provides the best BER with the first type of data (See Section 5.1). As described in Chapter 5, there is a large improvement over the threshold method when using the linear SVM classifier with the first type of data. As for the second type of data, I have the following three findings:

1. The value of BER is low for both the SVM and the threshold method. This is presumably because a higher quality transmission system is being modelled.

2. Using the linear SVM on both X- and Y- polarizations does not give any improvement over the threshold method.

3. Using a linear SVM with X- and Y- polarizations gives similar results.

These findings suggest that, it is much harder to improve the BER because there is much less room for improvement over such good BER values found using the threshold method. In addition, since the X- and Y-polarization data give similar classification results, so for time constraints reasons I shall focus on X-polarization data in the following study. But some results which are obtained using the Y-Polarization have been used to validate my work, see Appendix C.1.4.

6.2 Experiment B: Using a Different Number of Samples from One Symbol

The aim of experiments in this section is to investigate whether or not using a linear SVM with a specific number of samples from each symbol can help decode signals accurately at different distances starting with 2,000 km to 10,000 km. I use different input vectors to the linear SVM classifier: the central sample, the complete set of samples, and a different number of samples (three, five or thirty-two mid-samples) from the middle part of each symbol (see Figure 6.3).

Table 6.3 shows the SVM results using the second type of optical transmission data, received at the distances of 3,000 km, 5,000 km, 8,000 km and the maximum distance of 10,000 km. Note that I have shown results for four key distances only in this table. Results for all distances have been shown in Appendix C.1.1. The table presents values of SER and BER, which are averages over ten

Data set	**Method**	**SER** ($\times10^{-4}$)	**BER** ($\times10^{-4}$)
First data set	Linear SVM	55.07 ±14.05	27.63±7.1
	Threshold	120.91±28.66	60.64±14.56
Second data set (X-Pol)	Linear SVM	16.07±3	8.03±1.5
	Threshold	15.51±2.87	7.75±1.44
Second data set (Y-Pol)	Linear SVM	15.14±2.87	7.62±1.44
	Threshold	13.28±4.15	6.69±2.06

Table 6.2: A comparison between the improvement of the classification I achieved with the first and the second type of data, using the linear SVM and the threshold method. The complete set of samples of each symbol is used as an input to the classifier. Each optical transmission symbol is represented by $(\sin\theta,\cos\theta)$. The signal is received at the distance of 3,000 km. The SVM model is trained using the C parameter value of 512, which provides the best BER with the first type of data in Section 5.1. All results are averages over ten data sets. Ten data sets are selected randomly from 50 data sets of the first type of data. Using the SVM with the second type of data for both X- and Y- polarisation does not show any improvement over the threshold method as it has achieved with the first type of data. The BER obtained using the second type of data is low for both the SVM and the threshold method. X- and Y-polarization data give similar classification results.

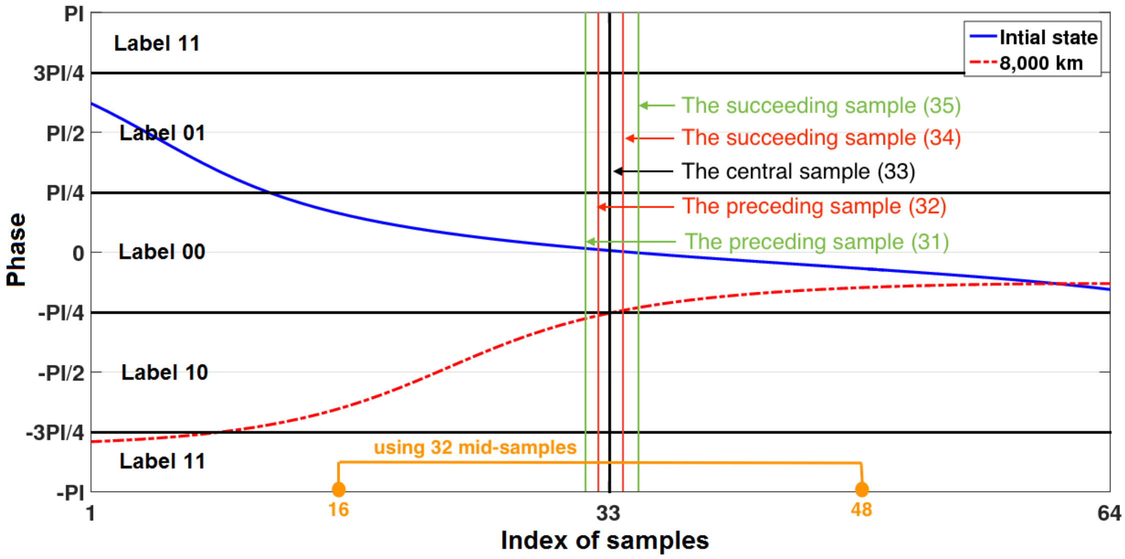


Figure 6.3: An explanation of using three, five or thirty-two mid-samples of each symbol as an input to the linear SVM classifier, see Table 6.3. Using three mid-samples of each symbol means that using the central sample (33^{rd}) and a sample either side (samples number (32^{nd}), (33^{rd}) and (34^{th})) as an input to the classifier, as shown as **the black line in the middle**, and the red lines on its sides. Using five mid-samples of each symbol means that using the central sample (33^{rd}) and two samples either side (samples number (31^{st}), (32^{nd}), (33^{rd}), (34^{th}) and (35^{th})) as an input to the classifier, as shown as **the black line in the middle**, and the red lines and the green lines on its sides. Using thirty-two mid-samples means that using the samples from number 16^{th} to 48^{th} of each symbol as an input to the classifier, look at the marks and text in orange. As can be seen, more samples give the classifier more information. Note that PI means π.

data sets. Also, it shows the p-value results obtained using the t-test. All results are compared with the threshold method. First, I focus on results obtained with the complete set of samples of the symbol. It can be seen that the errors increase as the distance travelled increase (see the values in red in Table 6.3).

Figure 6.4 shows the BER% of both the threshold method and the linear SVM using the complete set of samples of symbol over the distances from 2,000 km to 10,000 km. We can see that the BER obtained from the linear SVM is lower when compared with the threshold method from the distance of 4,000 km to 10,000 km, that is the SVM does constantly better than the threshold method over these distances. Figure 6.5 presents the improvement over the threshold method, that clearly shows an improvement starting at a distance of 3,000 km, though the improvement is minor there. At the distance of 2,000 km the linear classifier does not achieve any improvement, where the average of the BER using the threshold method is 1.07×10^{-4}, and the average of BER using the SVM is 1.49×10^{-4}. This means that there is about one error with the threshold method, and 2 errors with the SVM among the 10,768 test symbols.

Starting from the distance of 4,000 km till the maximum distance of 10,000 km, the results show that using the complete set of samples of each symbol as an input to the classifier gives the best BER, compared with the other input vectors and the result of the threshold method. The statistical results (p-value) of the t-test show that the results obtained at the distances of 2,000 km, 3,000 km and 5,000 km are likely to be due to the chance (p-value is equal to 0.12, 0.91 and 0.07, respectively). The classifier could not further improve the BER at the long distances of 9,000 km and 10,000 km so that the value of BER is less than (200×10^{-4}), which is the tolerable BER in the optical transmission field as mentioned in Chapter 1. For example, in Table 6.3d, the BER is (300.06×10^{-4}) at the distance of 10,000 km. The best result I have obtained so far is when using the complete set of samples of the symbol, compared with the other input vectors.

Now I discuss results obtained with other different numbers of samples from each symbol. As can be seen in Table 6.3, in general, SVM results using the central sample of the symbol are similar to the threshold results. Furthermore, using more samples from the symbol can improve the BER result, but not better than using the complete set of samples of the symbol. The reason might be because the signal distortion comes from many causes, and it is not just simple noise.

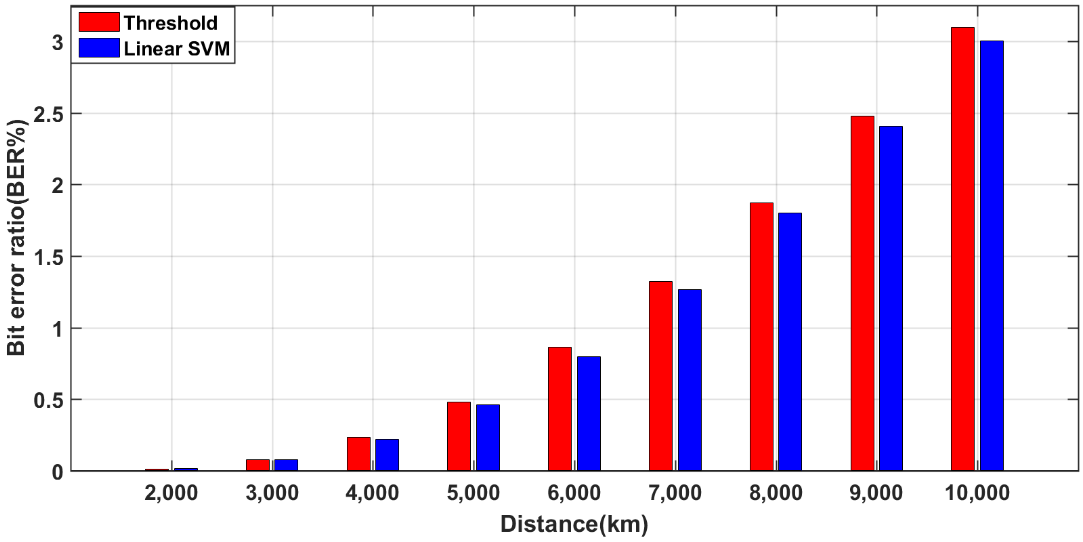


Figure 6.4: The BER (plotted in a linear bar graph) obtained using the complete set of samples of each optical transmission symbol as an input vector in the linear SVM classifier, over the distances from 2,000 km to 10,000 km. Comparing with the threshold method, the BER obtained from the linear SVM is lower, and the SVM constantly does better over the distances from 4,000 km to 10,000 km.

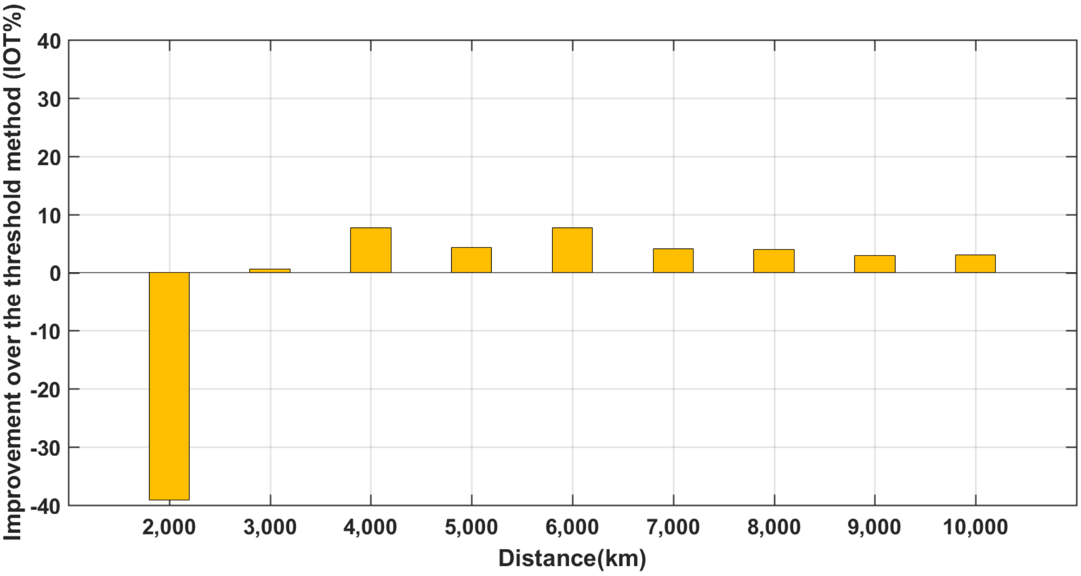


Figure 6.5: The improvement over the threshold method obtained using the complete set of samples of each optical transmission symbol as an input in the linear SVM classifier, over the distances from 2,000 km to 10,000 km. Note that the linear classifier does not achieve any improvement at the distance of 2,000 km, where the average of the BER using the threshold method is 1.07×10^{-4}, and the average of BER using the SVM is 1.49×10^{-4}.

Table 6.3: Using Different Samples from One Symbol

(a) Distance 3,000 km

Method	**No. of samples**	**No. of features**	**SER** ($\times 10^{-4}$)	**BER** ($\times 10^{-4}$)	**P-Value**
Threshold	Central	-	15.51±2.87	7.75±1.44	
SVM	Whole	64 × 2	15.42±3.87	7.71±1.94	0.91
	Central	1 × 2	15.14±3.61	7.57±1.81	0.42
	3 Mid	3 × 2	15.42±3.4	7.71±1.7	0.85
	5 Mid	5 × 2	15.32±2.98	7.66±1.49	0.64
	32 Mid	32 × 2	15.51±3.36	7.75±1.68	1

(b) Distance 5,000 km

Method	**No. of samples**	**No. of features**	**SER** ($\times 10^{-4}$)	**BER** ($\times 10^{-4}$)	**P-Value**
Threshold	Central	-	95.93±8.23	48.15±4.19	
SVM	Whole	64 × 2	91.75±10.85	46.06±5.47	0.07
	Central	1 × 2	95.56±8.03	47.97±4.06	0.58
	3 Mid	3 × 2	94.82±9.73	47.64±4.97	0.39
	5 Mid	5 × 2	95.84±10.28	48.06±5.2	0.88
	32 Mid	32 × 2	94.54±9.62	47.46±4.84	0.48

(c) Distance 8,000 km

Method	**No. of samples**	**No. of features**	**SER** ($\times 10^{-4}$)	**BER** ($\times 10^{-4}$)	**P-Value**
Threshold	Central	-	370.17±14.76	187.27±7.69	
SVM	Whole	64 × 2	356.24±13.74	179.84±7.36	0.001
	Central	1 × 2	370.54±16.23	187.41±8.36	0.85
	3 Mid	3 × 2	359.03±15.05	181.65±7.98	0.003
	5 Mid	5 × 2	359.68±16.22	181.93±8.59	0.01
	32 Mid	32 × 2	359.12±16.96	181.28±8.94	0.003

(d) Distance 10,000 km

Method	**No. of samples**	**No. of features**	**SER** ($\times 10^{-4}$)	**BER** ($\times 10^{-4}$)	**P-Value**
Threshold	Central	-	607.08±16.68	309.53±9.98	
SVM	Whole	64 × 2	588.5±23.71	300.06±13.05	0.01
	Central	1 × 2	610.33±19.01	311.2±10.95	0.06
	3 Mid	3 × 2	591.75±18.22	301.91±10.55	0.02
	5 Mid	5 × 2	589.34±17.75	300.66±10.02	0.004
	32 Mid	32 × 2	593.61±20.57	302.66±11.39	0.02

Table 6.3: The linear SVM results using different number of samples of each symbol at the distances of 3,000 km, 5,000 km, 8,000 km and the maximum 10,000 km, compared with the threshold method. SER and BER denote the average of Symbol and Bit Error Ratio over ten data sets, respectively. ($\times 2$) denotes that each sample in the input vector is represented by ($\sin\theta$,$\cos\theta$). Using only the central sample provides similar results to the threshold method (the text in blue), especially at the long distances such as 8,000 km and 10,000 km. In some cases, using more samples from the middle area of each symbol can improve the BER slightly. But it is not better than using the complete set of samples of each symbol, which provides the best BER at most of the distances. The text in red shows the best SVM results.

6.3 Experiment C: Using a Different Number of Samples from Neighboring Symbols

As mentioned in Section 3.3.2, my hypothesis is that the neighbouring symbols of the target symbol that is being decoded can have an effect on it. Although in Chapter 5 results show that using the neighbouring information does not improve the BER more than using just one symbol, I have decided to investigate further whether or not using a linear SVM classifier with different number of samples from the adjacent symbols can improve the BER on the second type of optical transmission data.

Table 6.4 shows the results obtained using the second type of optical transmission data, received at the distances of 3,000 km, 5,000 km, 8,000 km and the maximum 10,000 km. Again all other results for all other distances have been shown in Appendix C.1.2. The table presents average values of SER and BER over ten data sets. Also, it shows the statistical test results (p-value) of the t-test between the SVM and the threshold method. All results are compared with the results of the threshold method, and with the best linear SVM result I have obtained in the previous section 6.2, which uses the complete set of samples of each symbol.

In general, the best SVM results at most of the distances which are 3,000 km, 4,000 km, 5,000 km, 7,000 km and 8,000 km, are obtained when using the central sample, from the target symbol and symbol either side. The results show clearly that using more than one symbol as an input to the classifier improves the classification performance above that of the threshold method. In particular, using more than one symbol (adjacent symbols) gives better results than using only the symbol being decoded. However, when using just the adjacent succeeding symbol gives poorer results than using only the target symbol as an input to the classifier. This could be because the symbols are more affected by the preceding symbols, not succeeding symbols, see Table 6.4. Again, the classifier fails to further improve the BER at the long distances of 9,000 km and 10,000 km, where the tolerable BER should be less than 200×10^{-4} as has been mentioned earlier. For example, in Table 6.4d, the BER at the distance of 10,000 km is 297.14×10^{-4}.

Figure 6.6 shows BER% over the distances from 2,000 km to the maximum 10,000 km, of both the threshold method and linear SVM using the central sample of the target symbol and symbol either side. It can be seen clearly that the SVM outperforms the threshold method from a distance of 3,000 km to a distance of 10,000 km. Figure 6.7 presents the improvement over the threshold method. It shows that there is an improvement at the distance of 2,000 km which can not be seen easily in Figure 6.6.

In summary, so far the best improvement of the BER is obtained when using the linear SVM with the central sample from each of three adjacent symbols. The t-test results indicate that the difference between the threshold method and the SVM using this input vector across all distances is not due to the chance. The reason that using central samples work better might be because the

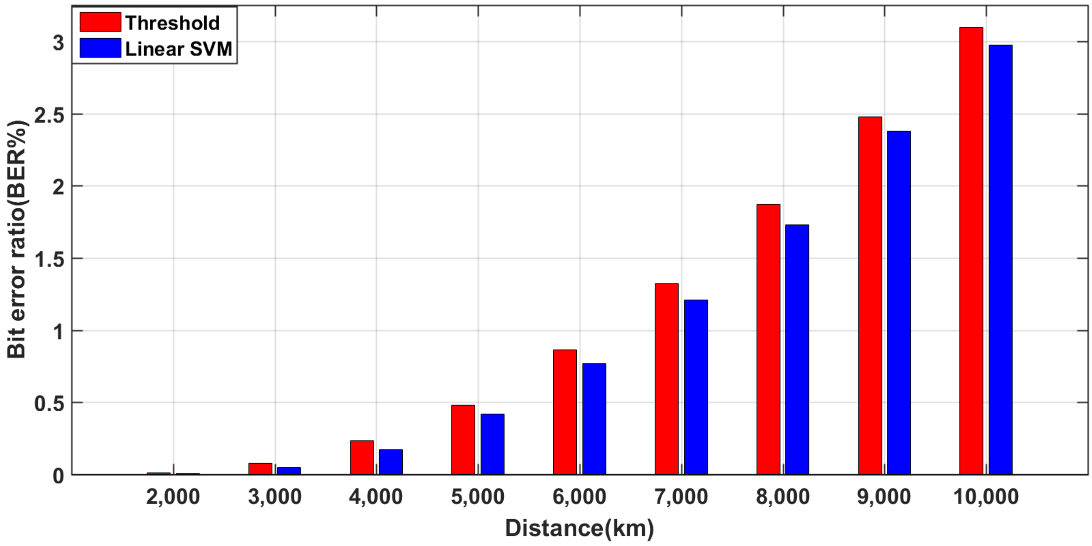


Figure 6.6: The BER (plotted in linear bar graph) obtained using the neighbouring information (the central sample of the target symbol and symbol either side of each optical transmission symbol) as an input in the linear SVM classifier, over the distances from 2,000 km to 10,000 km. From the comparison, the SVM clearly outperforms the threshold method over the distances from 3,000 km to 10,000 km.

symbol reaches the desired phase at the mid-point, and is affected less by the distortion than the other samples. Therefore, in the next section, I shall investigate whether or not using only the central samples from more neighbouring symbols as an input vector to the linear SVM classifier improves things further.

6.4 Experiment D: Using only the Central Sample from the Target Symbol and Symbols either Side

In this experiment, an investigation is undertaken by applying a linear SVM classifier using a different number of neighbour's information from symbols before and after the symbol being classified, that is, the central sample from five or seven adjacent symbols (from the target symbol (the symbol being decoded) and two or three symbols either side, respectively). As was mentioned in the introduction to this chapter, Figure 6.1 provides an example of using the central sample from seven adjacent symbols. Table 6.5 presents results which are averages of SER/BER over ten data sets at the distances of 3,000 km, 5,000 km, 8,000 km and the maximum 10,000 km. Again all other results for all other distances have been shown in Appendix C.1.3. Also, the table shows the statistical results (p-value) of the t-test between the linear SVM and threshold method. All the results are compared with the threshold method, and with the best result I have obtained so far from the previous sections 6.2 and 6.3, namely, when using the SVM classifier with the complete set of samples from each symbol, and with the central sample from the target symbol and symbol either side, respectively.

Table 6.4: Using Samples from Neighboring Symbols: the letters P, T and S denote that preceding, target and succeeding symbol respectively.

(a) Distance 3,000 km

Method	**No. of symbols and samples**	**No. of features**	**SER** $(\times 10^{-4})$	**BER** $(\times 10^{-4})$	**P-Value**
Threshold	1(Central/T)	-	15.51±2.87	7.75±1.44	
SVM	1(Whole/T)	64×2	15.42±3.87	7.71±1.94	0.91
SVM	3(Central/P,T,S)	3×2	9.85±1.87	4.92±0.93	0.00001
	3($\frac{1}{2}P$,Whole/T,$\frac{1}{2}S$)	128×2	13.65±3.64	6.83±1.82	0.11
	3(Whole/P,T,S)	192×2	12.54±3.31	6.27±1.66	0.01
	2(Central/P,T)	2×2	12.72±2.7	6.36±1.35	0.01
	2(Central/T,S)	2×2	12.91±2.95	6.45±1.48	0.001
	3(Central/2P,T)	3×2	11.98±2.82	5.99±1.41	0.0002
	2(Whole/P,T)	128×2	15.04±2.58	7.52±1.29	0.67

(b) Distance 5,000 km

Method	**No. of symbols and samples**	**No. of features**	**SER** $(\times 10^{-4})$	**BER** $(\times 10^{-4})$	**P-Value**
Threshold	1(Central/T)	-	95.93±8.23	48.15±4.19	
SVM	1(Whole/T)	64×2	91.75±10.85	46.06±5.47	0.07
SVM	3(Central/P,T,S)	3×2	83.13±8.83	41.75±4.41	0.00003
	3($\frac{1}{2}P$,Whole/T,$\frac{1}{2}S$)	128×2	87.12±9.36	43.7±4.65	0.001
	3(Whole/P,T,S)	192×2	85.45±10.58	42.91±5.34	0.002
	2(Central/P,T)	2×2	87.48±11.17	43.93±5.68	0.0004
	2(Central/T,S)	2×2	94.35±8.16	47.36±4.04	0.21
	3(Central/2P,T)	3×2	85.25±11.3	42.81±5.69	0.0002
	2(Whole/P,T)	128×2	87.76±7.21	44.07±3.68	0.0001

(c) Distance 8,000 km

Method	**No. of symbols and samples**	**No. of features**	**SER** $(\times 10^{-4})$	**BER** $(\times 10^{-4})$	**P-Value**
Threshold	1(Central/T)	-	370.17±14.76	187.27±7.69	
SVM	1(Whole/T)	64×2	356.24±13.74	179.84±7.36	0.001
SVM	3(Central/P,T,S)	3×2	342±13.65	173±7.43	0.0001
	3($\frac{1}{2}P$,Whole/T,$\frac{1}{2}S$)	128×2	343.46±8.8	173.49±4.79	0.00003
	3(Whole/P,T,S)	192×2	344.01±8.57	173.82±4.87	2.26E-05
	2(Central/P,T)	2×2	351.32±15.27	177.7±8.09	0.0002
	2(Central/T,S)	2×2	368.78±10.61	186.62±5.71	0.57
	3(Central/2P,T)	3×2	347.14±13.17	175.61±7.2	0.0001
	2(Whole/P,T)	128×2	349.18±12.16	176.45±6.57	0.00003

Table 6.4: Using Samples from Neighboring Symbols (continued)

(d) Distance 10,000 km

Method	**No. of symbols and samples**	**No. of features**	**SER** ($\times 10^{-4}$)	**BER** ($\times 10^{-4}$)	**P-Value**
Threshold	1(Central/T)	-	607.08±16.68	309.53±9.98	
SVM	1(Whole/T)	64 × 2	588.5±23.71	300.06±13.05	0.01
SVM	3(Central/P,T,S)	3 × 2	583.32±19.93	297.14±11.51	0.0004
	3($\frac{1}{2}P$,Whole/T,$\frac{1}{2}S$)	128 × 2	576.39±17.95	294.00±10.15	0.0001
	3(Whole/P,T,S)	192 × 2	576.58±21.04	293.95±11.50	0.0004
	2(Central/P,T)	2 × 2	588.97±23.48	300.33±13.41	0.005
	2(Central/T,S)	2 × 2	606.8±20.44	309.39±11.73	0.88
	3(Central/2P,T)	3 × 2	577.64±18.38	294.39±10.91	0.00003
	2(Whole/P,T)	128 × 2	584.6±28.43	297.97±15.4	0.004

Table 6.4: The results using information from neighboring symbols at the distances of 3,000 km, 5,000 km, 8,000 km and the maximum 10,000 km, compared with the threshold method. SER and BER show the average value over ten data sets, respectively. The letters P, T and S denote the preceding, target and succeeding symbol, respectively. In $M(X/NP, T, NS)$, M is the number of symbols, X is the position of the samples of each used symbol, and N is the number of symbols that are used from the preceding and succeeding symbols of the target symbol (T). ($\frac{1}{2}P$,Whole/T,$\frac{1}{2}S$) denotes that 32 samples (from sample 33_{rd} to sample 64_{th}) of the preceding symbol, the complete set of samples (64 samples) of the target symbol, and 32 samples (from 1_{st} sample to sample 32_{nd}) of the succeeding symbol are used as an input to the classifier. Note: using more than one symbol as an input to the classifier improves the BER over the threshold method (the text in blue). Also, it provides better results than using the complete set of samples of one symbol except when using the central sample of the target and only the succeeding symbol. Using the central sample of the target symbol and symbol either side gives the best SVM results at most of the distances. The text in red shows the best SVM results.

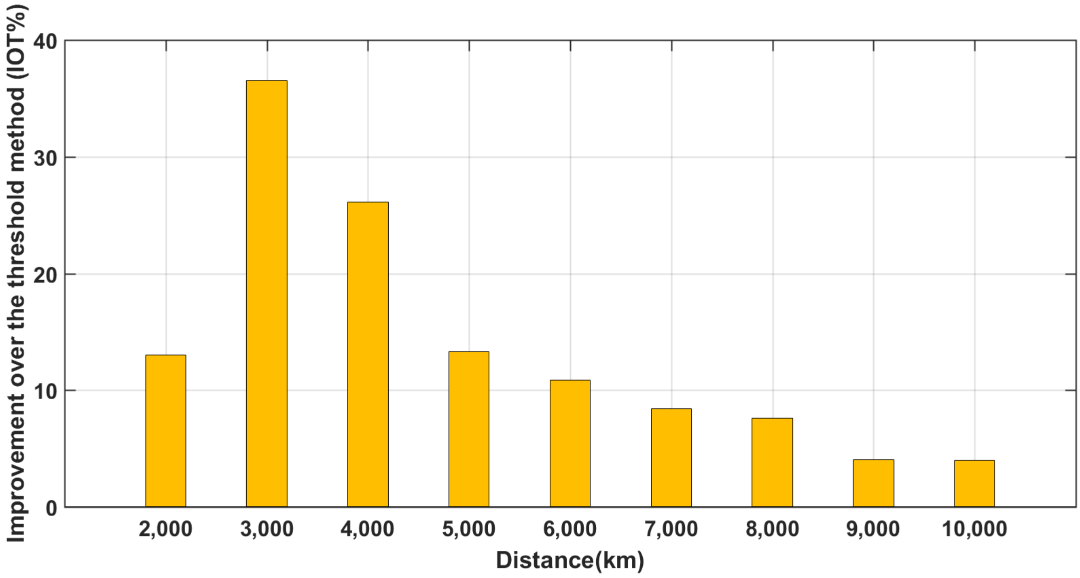


Figure 6.7: The improvement over the threshold method obtained using the neighbouring information (the central sample of the target symbol and symbol either side) as an input in the linear SVM classifier, over the distances from 2,000 km to 10,000 km. There is an improvement at the distance of 2,000 km which can not be seen easily in Figure 6.6.

The linear SVM using the input vector including the central sample from seven consecutive symbols (using three either side of the target symbol) provides the best results so far, especially, at the long distances from 6,000 km to 10,000 km. Moreover, using the central sample from three and five adjacent symbols gives similar results to the those obtained from seven neighbouring symbols. Furthermore, all the results obtained using the central sample from different number of adjacent symbols present an improvement over the threshold method, and the linear SVM using one symbol as an input to the classifier. For example, considering 8,000 km and more than 10,000 test symbols. There are about 172 errors using the central sample from seven adjacent symbols, which shows 15 errors less than the threshold method, and about eight errors less than using the complete set of samples of the symbol as an input to the classifier. T-test conducted between using different inputs, that is: the central sample from seven symbols and five symbols, and the central sample from seven symbols and three symbols gives p-value of 0.1 and 0.49, respectively. This suggests that there is no statistically significant difference between using these input vectors.

The statistical results of t-test for distances from 3,000km to 10,000 km provide evidence that the improvement over the threshold method is not likely to be due to the chance, since all p-values are less or equal to 0.01. However, the classifier still could not improve the BER at the long distances of 9,000 km and 10,000 km, to the tolerable threshold 200×10^{-4}. It can be seen in Table 6.5d, the BER value at the distance of 10,000 km is 292.85×10^{-4}.

Figure 6.8 shows the BER over the nine distances for both the threshold method and the SVM, using the seven central samples from the target symbol and three symbols either side. Also, we can

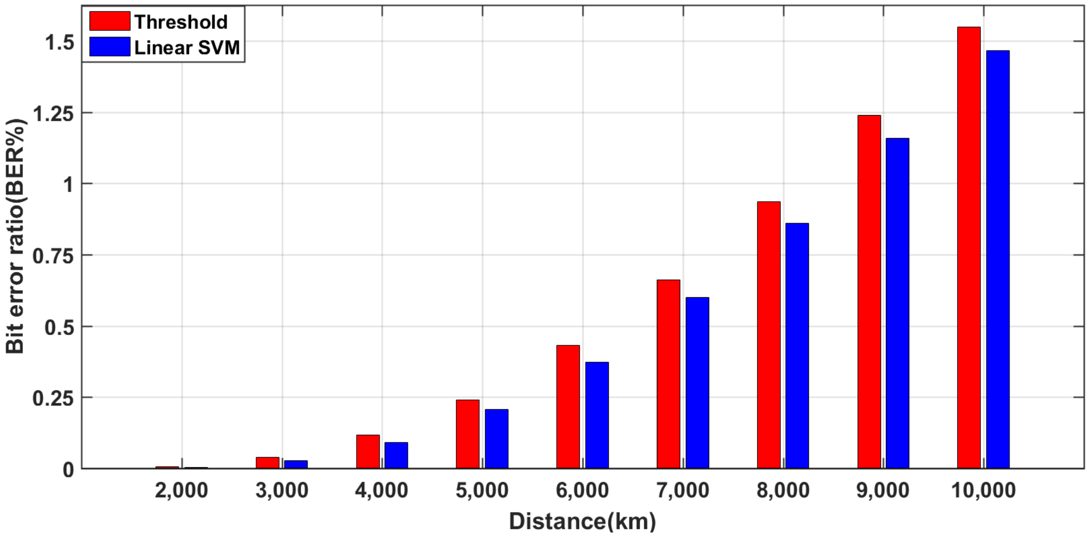


Figure 6.8: The BER (plotted in linear bar graph) obtained using the central sample from seven adjacent symbols (the central sample from the target symbol and three symbols either side of each optical transmission symbol) as an input in the linear SVM classifier, over the distances from 2,000 km to 10,000 km. The improvement of BER is clear compared with the threshold method.

see the improvement over the threshold method clearly along all the distances in Figure 6.9.

In summary, using the central sample from a different number of adjacent symbols as an input vector to the linear SVM classifier improves the BER compared with using the threshold method. Although in general using seven symbols gives slightly better results than using either three or five symbols, the performance of using a different number of neighbouring symbols is very similar. Considering the fact that the bigger an input vector is, the more computational time is needed, I have decided to stop using more neighbouring symbols. Furthermore, using neighboring symbols does not improve the BER to the allowable BER value in the optical transmission, that is less than 0.02 at the distances of 9,000 km and 10,000 km.

6.5 Conclusion

Results in this chapter show that, interestingly, using information from immediate preceding symbols definitely helps successful decoding. Moreover, when information from the succeeding symbols is also involved, the BER can be further improved. In this chapter, the best linear SVM results I have got so far is the result obtained using the central sample from the target symbol and three symbols either side. But the BER could not yet be improved to be less than 200×10^{-4} at the distances of 9,000 km and 10,000 km. Figure 6.10 shows a comparison of the improvement over the threshold method (IOT) when using three different input vectors to the classifier: 1) the central sample of the target symbol only. 2) the central sample from the target symbol and symbol either side. 3) the central sample from the target symbol and three symbols either side. It shows a decreasing trend

Table 6.5: Using Central Samples from Neighboring Information

(a) Distance 3,000 km

Method	No. of symbols and samples	No. of features	SER ($\times10^{-4}$)	BER ($\times10^{-4}$)	P-Value
Threshold	1(Central/T)	-	15.51±2.87	7.75±1.44	
SVM	1(Whole/T)	64×2	15.42±3.87	7.71±1.94	0.91
	3(Central/P,T,S)	3×2	9.85±1.87	4.92±0.93	0.00001
SVM	5(Central/2P,T,2S)	5×2	10.4±2.96	5.2±1.48	0.000002
	7(Central/3P,T,3S)	7×2	10.87±2.81	5.43±1.4	0.0002

(b) Distance 5,000 km

Method	No. of symbols and samples	No. of features	SER ($\times10^{-4}$)	BER ($\times10^{-4}$)	P-Value
Threshold	1(Central/T)	-	95.93±8.23	48.15±4.19	
SVM	1(Whole/T)	64×2	91.75±10.85	46.06±5.47	0.07
	3(Central/P,T,S)	3×2	83.13±8.83	41.75±4.41	0.00003
SVM	5(Central/2P,T,2S)	5×2	82.2±9.09	41.29±4.59	0.00002
	7(Central/3P,T,3S)	7×2	82.58±8.85	41.48±4.4	0.0001

(c) Distance 8,000 km

Method	No. of symbols and samples	No. of features	SER ($\times10^{-4}$)	BER ($\times10^{-4}$)	P-Value
Threshold	1(Central/T)	-	370.17±14.76	187.27±7.69	
SVM	1(Whole/T)	64×2	356.24±13.74	179.84±7.36	0.001
	3(Central/P,T,S)	3×2	342±13.65	173±7.43	0.0001
SVM	5(Central/2P,T,2S)	5×2	347.02±17.05	175.51±8.8	0.002
	7(Central/3P,T,3S)	7×2	339.9±9.8	171.95±5.41	0.000004

(d) Distance 10,000 km

Method	No. of symbols and samples	No. of features	SER ($\times10^{-4}$)	BER ($\times10^{-4}$)	P-Value
Threshold	1(Central/T)	-	607.08±16.68	309.53±9.98	
SVM	1(Whole/T)	64×2	588.5±23.71	300.06±13.05	0.01
	3(Central/P,T,S)	3×2	583.32±19.93	297.14±11.51	0.0004
SVM	5(Central/2P,T,2S)	5×2	576.07±17.1	293.7±10.08	0.00001
	7(Central/3P,T,3S)	7×2	574.27±15.81	292.85±9.561	0.000004

Table 6.5: The linear SVM results using the central sample from different numbers of adjacent symbols at the distances of 3,000 km, 5,000 km, 8,000 km and the maximum 10,000 km, compared with the threshold method. SER and BER show the average value over ten data sets. Note: Using the central sample from the neighbouring symbols improves the BER, compared with the threshold method (the text in blue). Using the central sample from seven neighbouring symbols gives slightly better results than using either three or five symbols, but the performance of using a different number of neighbouring symbols is very similar. The text in red presents the best SVM results.

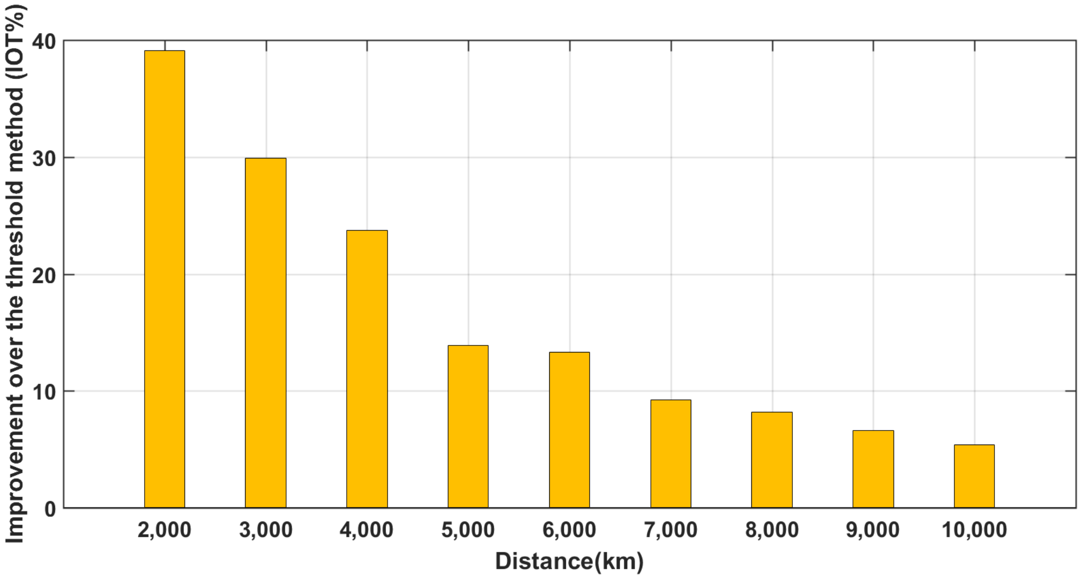


Figure 6.9: The improvement over the threshold method using the central sample from seven adjacent symbols (the central sample from the target symbol and three symbols either side of each optical transmission symbol) as an input in the linear SVM classifier, over the distances from 2,000 km to 10,000 km. The IOT decreases as the distance travelled by the optical transmission data increases.

of IOT when the distance travelled by the information is increased. Also, using more neighbouring information gives better results than using only one symbol either side except for the distances of 3,000 km and 4,000 km.

Now I investigate prediction errors with each of the four classes. Table 6.6 shows the number of predicted symbols in each class. The results are obtained from the first data set of the second type of data, using an SVM at a distance of 8,000 km. The input vector that is used in the classifier is the central sample of the target symbol and three symbols either side. As has been shown, most of the symbol errors have just one-bit errors (the numbers in blue), that is, the symbol class is mis-classified as one of the adjacent classes. Comparing with Table 6.1, one can see that some of these errors are corrected using the SVM compared with the threshold method. But the symbol errors that have two-bit errors were not corrected using the classifier (the numbers in red).

The final two figures show the information from Table 6.6 and the comparison with Table 6.1 diagrammatically. Figure 6.12 visualizes the number of errors in each class using a grouped-bar plot. The highest two percentage values of mis-classified patterns within each group are the two corresponding adjacent classes of each class. Furthermore, Figure 6.11 presents four bar graphs of incorrectly predicted symbols for each class with two methods, the threshold and the best SVM model, namely. As can be seen, the SVM can gives a slightly better results over all four classes at the distance of 8,000 km on one-bit errors; while both methods perform the same on two-bit errors.

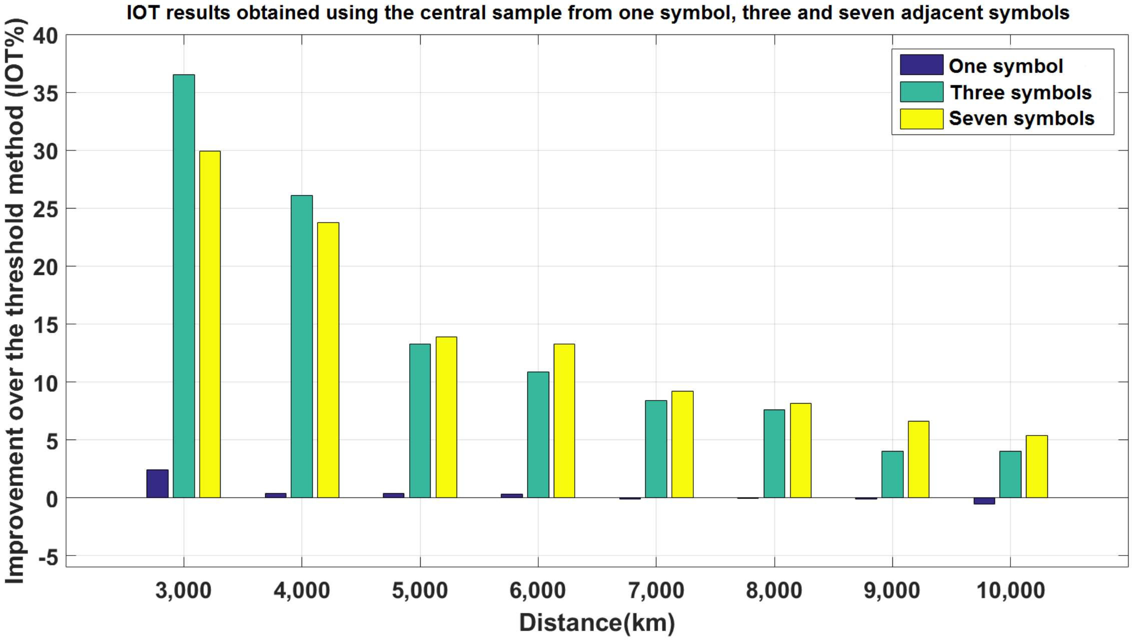


Figure 6.10: A comparison of the improvement over the threshold method results, when using the central sample from one symbol, three and seven adjacent symbols. There is a trend of decreasing the IOT whenever the distance is increased.

Symbol	**Predicted class**			
class	**00**	**01**	**10**	**11**
00	2536	39	42	2
01	53	2610	1	46
10	56	1	2650	31
11	2	38	49	2609

Table 6.6: The number of predicted symbols in each class. The result is obtained from using the linear SVM with the first data set of the second type of data, at a distance of 8,000 km. The test set consists of 10,765 symbols in total. The input vector that is used in the SVM classifier of each symbol being decoded includes the central sample of the target symbol and three symbols either side. As it has shown, most of the symbol errors have a one-bit error. Comparing with Table 6.1, one can see that some of these errors are corrected using SVM compared with the threshold method. But the symbol errors that have a two-bit error are not corrected using the classifier (in red). The number of symbols that are classified correctly for each class is shown in blue

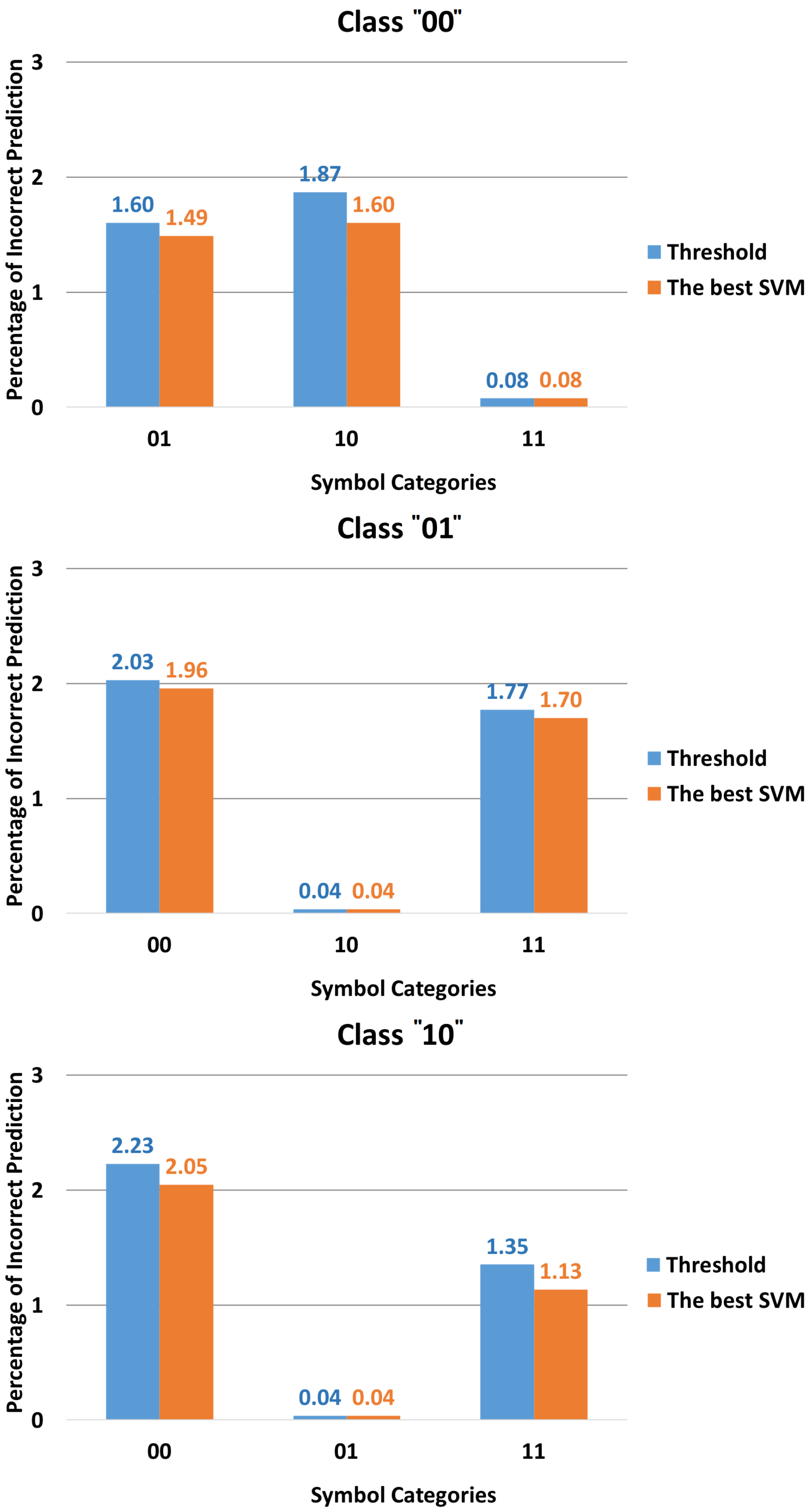


Figure 6.11: Four bar graphs on the percentage of incorrect prediction for each class (continued over page).

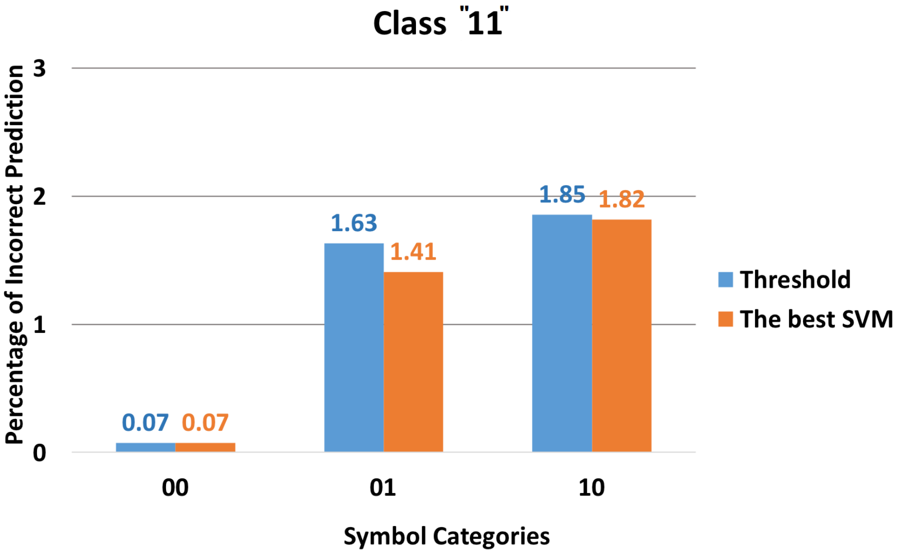


Figure 6.11: Four bar graphs on the percentage of incorrect predictions for each class. The results are obtained using the threshold method and the SVM with the first data set of the second type of data, at a distance of 8,000 km. The input vector that is used in the SVM classifier of each symbol being decoded includes the central sample of the target symbol and three symbols either side. Most of symbol errors have only a one-bit error, which results by predicting the symbol's class as one of the adjacent classes. Some of these errors are corrected using SVM compared with the threshold method. But the symbol errors that have a two-bit error are the same for both the threshold method (see Table 6.1) and the SVM classifier.

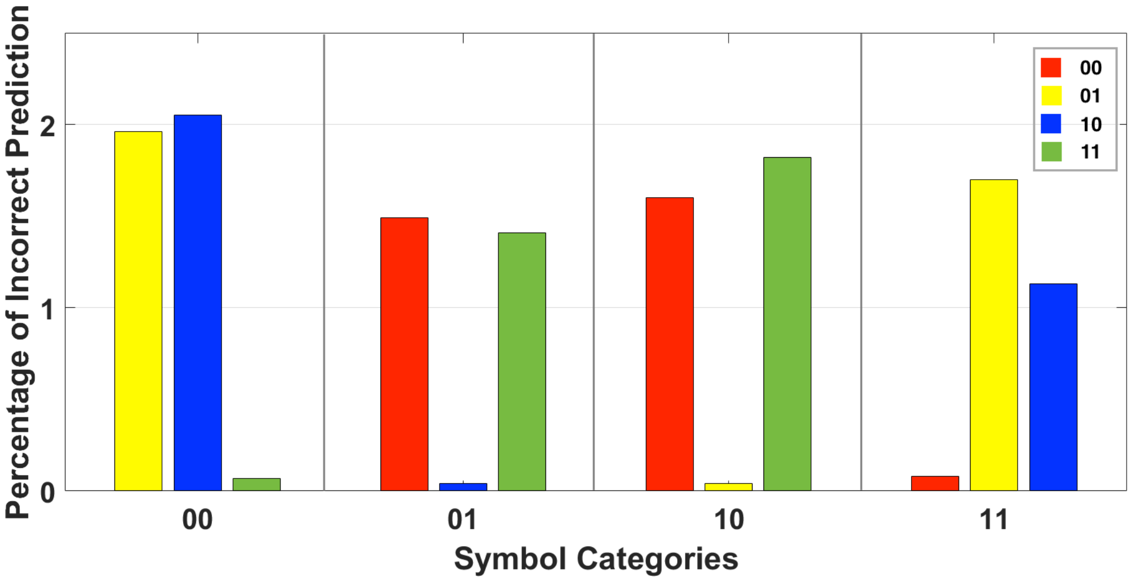


Figure 6.12: The percentage of incorrectly predicted symbols in each class. The result is obtained using the linear SVM classifier with the first data set of the second type of data, at a distance of 8,000 km. The test set consists of 10,765 symbols. The input vector that is used in the SVM classifier of each symbol being decoded includes the central sample of the target symbol and three symbols either side. Table 6.6 shows the same information in numbers.

Chapter 7

Experiments and Results using an SVM with Wavelets

Results in this chapter except for Table 7.4b have been published in the proceedings of ESANN 2017 (Binjumah et al., 2017b) and ICPRAM 2017 (Binjumah et al., 2017a).

Currently, in many areas such as digital processing and, in particular, image processing, wavelet transforms are a widely used as feature extraction method (Yadav et al., 2015). At a conference prior to the one that this work was presented several researchers extolled the virtues of wavelet transforms and were of the opinion that they would aid my work considerably. This chapter is outcome of those discussions.

In this chapter I shall investigate the effect of applying the wavelet transforms (WT) prior to using an SVM in an attempt to improve the BER of the second type of optical data at the distances of 8,000 km, 9,000 km and 10,000 km.

The wavelet transform method has been described in Chapter 3, Section 3.3.3. There are two parts to this chapter. First, in Section 7.1 I investigate the use of wavelets by analyzing how effective wavelet transformation would be on a general sinusoidal signal that has Gaussian noise added to the amplitude (It is the signal noise ($\mathcal{E}_s$) that has been mentioned in Eq. 3.4 and Eq. 3.7), frequency ($\mathcal{E}_f$ in Eq. 3.5) or to the phase ($\mathcal{E}_\varphi$ in Eq. 3.8), more details can be found in Section 3.2. This investigation was carried out to familiarize myself with the use of wavelets and to investigate which sort of signals wavelets deal with best. Then, in Section 7.2 I apply the wavelet transformation to the actual simulated optical transmission data.

A summary of the experiments that are described in this chapter is shown in the following:
Experiment A: Simple sinusoidal waves with frequency noise and phase noise, Section 7.1.
Experiment B: The linear SVM using wavelet transformations as inputs on the optical transmission data, Section 7.2.

7.1 Experiment A: Wavelet Transform with Sinusoidal Waves

7.1.1 Results on Sinusoidal Waves with Frequency Noise

The aim of this section is to investigate whether using wavelet transforms can enable the SVM to better distinguish data from two different wave classes containing frequency noise than without using the wavelet transform (WT). A description on how these data classes are generated is shown in Section 3.2.1. The data sets herein consist of different combinations of two classes of data; they are AC, AB, AD and BD. As has been described in Chapter 3, each letter represents one class and each class has a different value of signal frequency (f). For example, AC is a combination of the two classes of data A and C, and so on. Each pair of classes have different distances between their signal frequencies (see Table 3.2) and so represent a different level of difficulty when attempting to classify the noisy data. The 1,000 data points/waves (500 from each class) are randomly selected to give 700 waves that are used to train the model, and the rest of the data (300 waves) are used as a test set.

The method of generating the individual classes A,B,C and D is by adding frequency noise to the basic wave, so effectively we already have frequency noise ($\mathcal{E}_f$ in Eq. 3.5). We now investigate also adding Amplitude noise (It is the signal noise ($\mathcal{E}_s$) that has been mentioned in Eq. 3.4). Six tests are made for each data set: the waves with no added amplitude 'noise', without and with two types of wavelet transforms; the waves with added amplitude 'noise', without and with two types of wavelet transforms. The two wavelet transforms are: Haar and DB4 wavelet transforms. I have tried at level 1, level 2 and level 3, but level 2 has given me the best results. Therefore, I have shown results only at level 2. Then, the results are compared with each other to see the effect of using wavelet transforms.

Table 7.1 shows the results, also the table presents the type and level of the wavelet transforms that are used, and the accuracy (%) of the classification using the linear SVM. The following observations can be seen from Table 7.1:

1. The difference between the values of the signal frequency for class A and C is quite high (a difference of 10) and consequently the data could be partitioned with 98.67% accuracy. Using the wavelet transforms on the test set AC with or without amplitude noise ($\mathcal{E}_s$) does not give any improvement. This can be explained by the data distribution. As it has been shown in Figure 3.12, 1.33% signals from the two classes are overlapped, which are exactly those misclassified signals.

2. It can be seen that when two classes have closer values of signal frequency, the data is more overlapped and the accuracy rate is further reduced. For example, the difference between two values of signal frequency of the classes A and B is 5, and the accuracy rate is 91% on this dataset with only frequency noise ($\mathcal{E}_f$) and without using WT (see Table 7.1b). When

the difference becomes smaller, for instance, a value of 3 (the combination of BD) or 2 (the combination of AD), the corresponding accuracy rate is reduced to 79% (see Table 7.1c) or 69% (see Table 7.1d), respectively. This is obvious and just reflects the degree of separation of the data.

3. The use of wavelets does not have any effect on the data with just frequency noise in any of the tests. For example, the accuracy rate is always 91% with or without using WT on the dataset AB (see Table 7.1b).

4. Once the amplitude noise (signal noise ($\mathcal{E}_s$)) is added the use of wavelets does improve the accuracy back towards the values obtained with the Frequency noise ($\mathcal{E}_f$) only version. For instance, as shown in Table 7.1b, with classes A and B the wavelet transform on waves nearly brought the fully noisy wave performance up to that of the frequency only noisy wave (from 84.33% to 90.67%, which is very close to the 91% frequency only-noisy version). This is the biggest improvement I have obtained using WT over four groups of data.

These observations suggest that using wavelet transforms (WT) could not solve the problem of frequency noise ($\mathcal{E}_f$) in the optical transmission data, since they did not improve the classification accuracy with the simple data that has only frequency noise ($\mathcal{E}_f$). However, the wavelet transforms did improve the performance of the data with added Amplitude/signal noise ($\mathcal{E}_s$) .

7.1.2 Results on Sinusoidal Waves with Phase Noise

The aim of these experiments is to investigate whether or not using wavelet transforms can affect the signal decoding that have phase noise. The data set used in this experiment consists of 1,000 data points (waves/signals), and 640 equally spaced sample values for each data point. Half of the data set has a phase of zero, and the other half has a phase of 90. A description on how the data is generated can be found in Section 3.2.2. In this section, 600 signals are used to train the linear SVM model, and the rest of the data (400 signals) are used as a test set. There are four types of test set: the signals with no amplitude noise (It is the signal noise ($\mathcal{E}_s$) that has been mentioned in Eq. 3.7), those with amplitude noise (noisy signals), and signals with and without using wavelet transforms (extracted signals).

Figure 7.1 shows two signals: the red and blue lines, and also their noisy version, and ten signals of each class after adding the random phase ($\mathcal{E}_\varphi$ in Eq. 3.8) and amplitude noise ($\mathcal{E}_s$ in Eq. 3.7). Figure 7.2, shows ten extracted approximations from Haar wavelet transform at level 2. Since the average of differences between the original (Figure 7.1) and extracted signal (the Approximation part after WT, Figure 7.2) gets bigger after increasing the level of wavelet transforms. I also tried to normalize the extracted signals to the range of $[-1, 1]$ (as shown in the original waves (Figure 7.3)) to see if that would help in improving the classification process or not. In this section:

Table 7.1: sinusoidal signals with the frequency noise

(a) Group of data AC (10)

Type of noise	Type of (WT)	Level of (WT)	Accuracy%
$\mathcal{E}_f$	No	0	98.67
	Haar wavelet	2	98.67
	db4 wavelet	2	98.67
$\mathcal{E}_f\, with\, \mathcal{E}_s$	No	0	98.67
	Haar wavelet	2	98.67
	db4 wavelet	2	98.67

(b) Group of data AB (5)

Type of noise	Type of (WT)	Level of (WT)	Accuracy%
$\mathcal{E}_f$	No	0	91
	Haar wavelet	2	91
	db4 wavelet	2	91
$\mathcal{E}_f\, with\, \mathcal{E}_s$	**No**	**0**	**84.33**
	Haar wavelet	**2**	**90**
	db4 wavelet	**2**	**90.67**

(c) Group of data BD (3)

Type of noise	Type of (WT)	Level of (WT)	Accuracy%
$\mathcal{E}_f$	No	0	79
	Haar wavelet	2	79
	db4 wavelet	2	79
$\mathcal{E}_f\, with\, \mathcal{E}_s$	No	0	73
	Haar wavelet	2	77.33
	db4 wavelet	2	76.67

(d) Group of data AD (2)

Type of noise	Type of (WT)	Level of (WT)	Accuracy%
$\mathcal{E}_f$	No	0	69
	Haar wavelet	2	69
	db4 wavelet	2	69
$\mathcal{E}_f\, with\, \mathcal{E}_s$	No	0	65.33
	Haar wavelet	2	67.67
	db4 wavelet	2	67

Table 7.1: Linear SVM results on four different data sets of noisy sinusoidal signals. Note: The number in the brackets beside the name of group of data is the difference between the values of the signal frequency (f). $\mathcal{E}_f$ denotes frequency noise only is added to the signal. $\mathcal{E}_f\, with\, \mathcal{E}_s$ denote both Frequency and Amplitude (signal) noise are added to the signal. The text in blue refers to the results that are obtained without applying wavelets. The text in red refers to the best wavelet results. **The text in bold refers to the data set that gives the best result absolutely in section 7.1.1**. Note: the high difference in the values of the signal frequency in data set AC allows the partition of its signals easily, except the waves that are ambiguous, and can't be classified even with using the WT. The data with only frequency noise ($\mathcal{E}_f$) does not give improved results using (WT). It gives an expectation that (WT) may not solve the problem of frequency noise in any other type of data, but it does improve the classification accuracy in those cases where amplitude noise ($\mathcal{E}_s$) is also added.

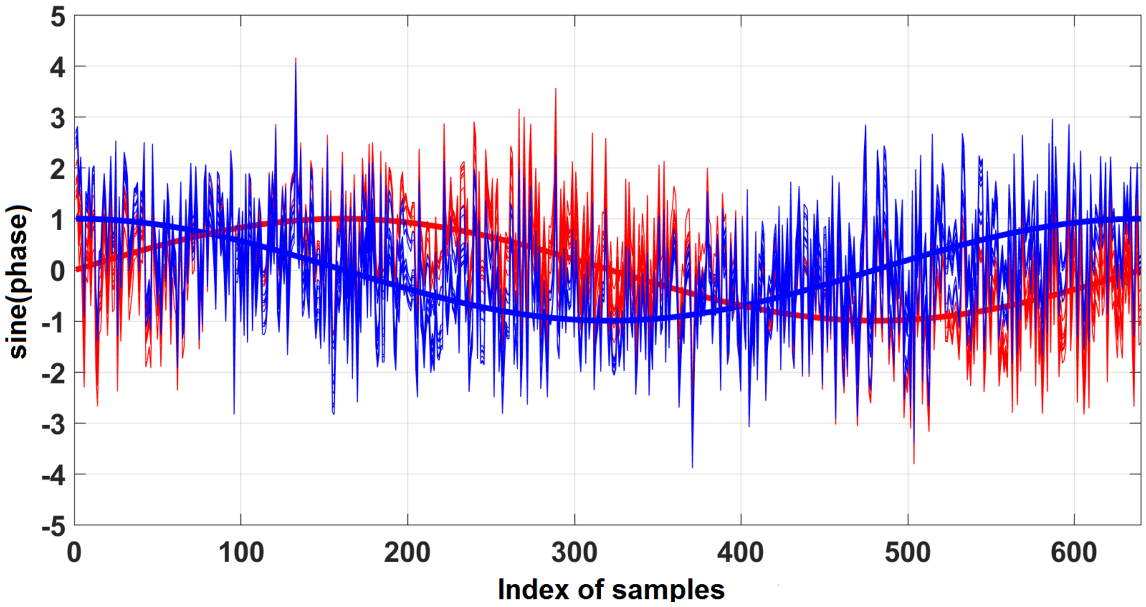


Figure 7.1: Ten Sinusoidal waves with phase ($\mathcal{E}_\varphi$) and amplitude ($\mathcal{E}_s$) noise compared with non-noisy waves (smoothed solid lines). Blue waves have a phase of 90 degree and red waves have phase of zero degree. Note that the range of the samples of the noisy sinusoidal wave is bigger than [-1, 1] because of the noise.

1. Haar wavelet transform at different levels from 1 to 5, and db4 wavelet transform at level 2 are implemented on noisy waves first.
2. The best wavelet representation from Point 1 is selected to be implemented on the waves with no amplitude noise (waves that have phase noise ($\mathcal{E}_\varphi$) only).
3. Two types of input to the classifier have been used. The first type is the complete set of samples extracted using wavelet transforms. The second one is the central sample of the extracted waves.

7.1.2.1 Linear SVM Results using Extracted Waves without Normalization

Table 7.2 presents the accuracy of prediction on the non-normalized extracted waves. Table 7.2a shows the results using the complete set of samples of the non-normalized extracted signals. The results in Table 7.2a do not show a noticeable improvement, where the accuracy rate before using the wavelet transform (using noisy signals) is 92.5%, and after using the wavelet transform is improved to 92.75% using Haar wavelets at levels 1, 2 and 5. Table 7.2b shows the SVM results using just the central sample of the non-normalized extracted waves. With less input information the classification results from the Haar wavelet transform are worse than those in Table 7.2a. Interestingly, the best result obtained is using the db4 wavelet transform at level 2, which is 93% and 1.25 % higher than the result obtained on the noisy signal without a wavelet transformation. In summary, using WT

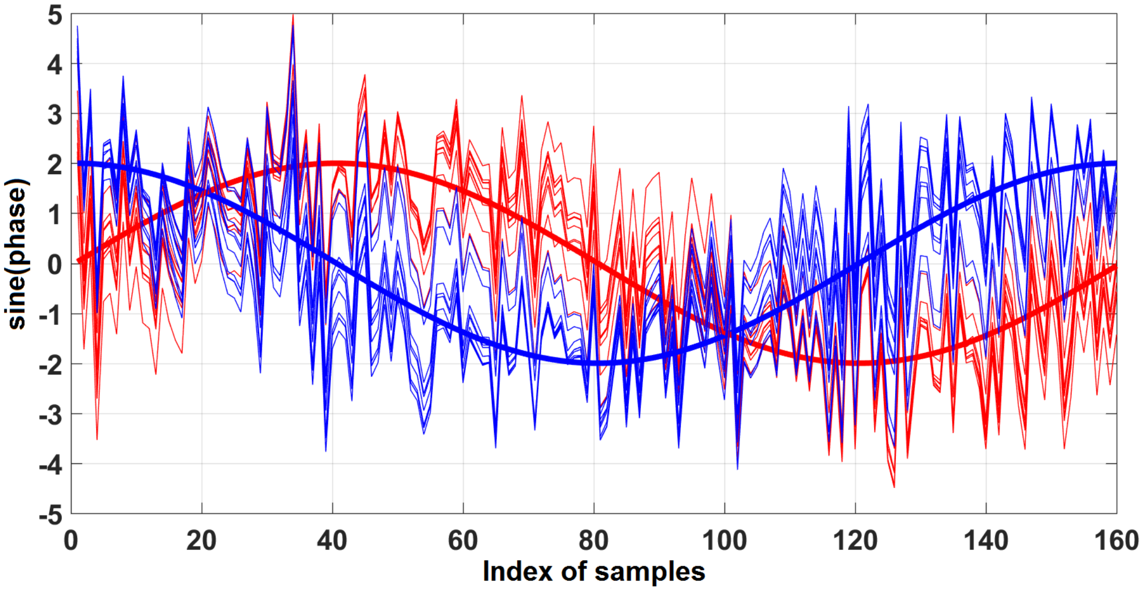


Figure 7.2: Ten extracted waves using Haar wavelet transform, at level 2 (Approximation part), compared with non-noisy extracted waves (smoothed solid lines). Blue waves have a phase of 90 degree and red waves have a phase of zero degree. The waves became smoother, but the average of difference between the original noisy wave (Figure 7.1) and extracted wave gets bigger after increasing the level of wavelet transforms (WT).

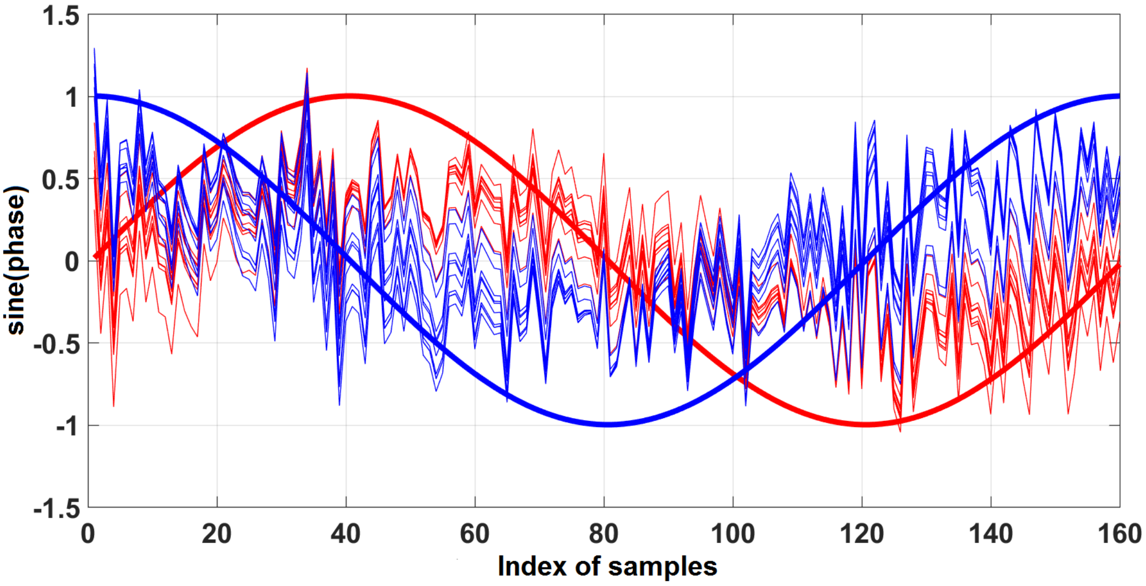


Figure 7.3: Ten extracted normalized waves using Haar wavelet transform, at level 2 (Approximation part), compared with non-noisy extracted waves (smoothed solid lines). Blue waves have a phase of 90 degree and red waves have a phase of zero degree. The range of the extracted waves is re-scaled to [-1, 1], similar to the original waves.

Table 7.2

(a) The complete set of samples of the extracted signal are used as an input to the classifier.

Type of noise	**Type of (WT)**	**Level of (WT)**	**Accuracy%**
$\mathcal{E}_\varphi + \mathcal{E}_s$	No	0	92.5
	Haar	**1**	**92.75**
	Haar	**2**	**92.75**
	Haar	3	92.5
	Haar	4	92.5
	Haar	**5**	**92.75**
	Db4	2	92.25
$\mathcal{E}_\varphi$	No	0	92.5
	Haar	2	91.5

(b) The central sample of the extracted signal are used as an input to the classifier.

Type of noise	**Type of (WT)**	**Level of (WT)**	**Accuracy%**
$\mathcal{E}_\varphi + \mathcal{E}_s$	No	0	91.75
	Haar	1	91.75
	Haar	2	91
	Haar	3	91
	Haar	4	90
	Haar	5	87.25
	Db4	**2**	**93**
$\mathcal{E}_\varphi$	No	0	91.75
	Db4	2	91.75

Table 7.2: A comparison between linear SVM results using noisy sinusoidal waves and the extracted waves. Note: $\mathcal{E}_\varphi + \mathcal{E}_s$ denote both Phase and amplitude/signal noise are added to the signal. $\mathcal{E}_\varphi$ denotes only phase noise is added to the signal. The text in blue refers to the results that are obtained without applying wavelets. The text in red refers to the best wavelet results. Note that using wavelet transforms does not improve the classification accuracy with the data that has only phase noise ($\mathcal{E}_\varphi$).

does not show any improvement with the data that has only phase noise with both types on input vectors: the complete set of samples and the central sample of the wave.

7.1.2.2 Linear SVM Results using Normalized Extracted Signals

Table 7.3 presents the accuracy of prediction on the normalized extracted signals. Table 7.3a shows the results using the complete set of samples of the extracted normalized signals. Again, the SVM results do not show much improvement, where the accuracy is only improved from 92.5% to 92.75% after using WT. Table 7.3b shows the results using the central sample of the extracted normalized wave. Here the wavelet transformations do have an effect. The best result is obtained using the DB4 wavelet transform at level 2, where the accuracy is improved to 93.75% (from 91.75%). Even with normalizing the waves, using WT does not show any improvement with the data that has only phase noise ($\mathcal{E}_\varphi$) with both types on input vectors: the complete set of extracted samples and the central sample of the extracted wave.

Comparing Tables 7.2 and 7.3, we see that in general, using the normalization can slightly improve the results, especially when using the central sample of the signals as inputs. However the overall results show the difficulty that wavelets have with phase distorted data. The results in this section suggests that wavelet transformation (WT) might not deal perfectly with the phase noise in the optical transmission data, because it did not improve the classification accuracy with the simple data that has only phase noise ($\mathcal{E}_\varphi$).

Table 7.3

(a) The complete set of samples of the extracted normalized signal are used as an input to the classifier.

Type of noise	**Type of (WT)**	**Level of (WT)**	**Accuracy%**
$\mathcal{E}_\varphi + \mathcal{E}_s$	No	0	92.5
	Haar	1	92.5
	Haar	2	92.5
	Haar	**3**	**92.75**
	Haar	4	92.5
	Haar	5	92.5
	Db4	**2**	**92.75**
$\mathcal{E}_\varphi$	No	0	92.5
	Haar	3	92.5

(b) The central sample of the extracted normalized signal are used as an input to the classifier.

Type of noise	**Type of (WT)**	**Level of (WT)**	**Accuracy%**
$\mathcal{E}_\varphi + \mathcal{E}_s$	No	0	91.75
	Haar	1	93.5
	Haar	2	93.5
	Haar	3	90.5
	Haar	4	93
	Haar	5	88
	Db4	**2**	**93.75**
$\mathcal{E}_\varphi$	No	0	91.75
	Db4	2	91.75

Table 7.3: A comparison between linear SVM results using noisy signal and the extracted normalized signals. Note: $\mathcal{E}_\varphi + \mathcal{E}_s$ denote both phase and amplitude/signal noise are added to the signal. $\mathcal{E}_\varphi$ denotes only phase noise is added to the signal. The text in blue refers to the results that are obtained without applying wavelets. The text in red refers to the best wavelet results. Note that using wavelet transforms does not improve the classification accuracy with the data that has only phase noise ($\mathcal{E}_\varphi$).

7.2 Experiment B: The linear SVM using wavelet transformations as inputs on the phase coded optical transmission data

Having familiarized myself with the use of wavelet transform and analysed how well they work on some simple artificial data, I will now carried out experiments on the simulated optical transmission data, namely the second type of data. The purpose of these experiments is to check whether or not using WT can affect decoding the distorted optical transmission signals.

In this section, a linear SVM is applied using different input vectors: just the central sample, the complete set of samples of each symbol (all 64 values) and neighbouring information from symbols before and after the symbol being classified. I focus on using the central sample from seven adjacent symbols (from the target symbol and three symbols either side), since it gives the best results in Chapter 6. A selection of different wavelet transformations are employed, from Haar level 1 and 2 to db4 level 2 wavelets. The wavelets are applied on each original whole symbol to a certain level, then either the whole approximation part or the central value of the approximation part is used as WT features. For example, consider the use of Haar wavelets to level 2. First, I have used Haar wavelets on the 64 samples of each symbol. This process generates an approximation part including 32 values at level 1, and a further approximation part including 16 values at level 2. For the whole approximation, I use all sixteen values. For the central value, I have selected the 9^{th} feature from each extracted symbol (either just the target symbol or the target and three symbols either side) to be used as an input to the classifier. More details in Appendix C.2.2.

Figure 7.4 shows an example of using three neighbouring symbols (the target symbol and symbol either side), at 8,000 km, as an input to the wavelet transformation. The top panel shows the original signal. The bottom panel shows the approximation parts of the same symbols. In the case of using only the central sample of three adjacent symbols after using WT, the final input vector to the classifier is the set of central values of approximation coefficients from each symbol (the green dots in the bottom panel), which are (0.8162, 0.2604, -3.0982)).

Table 7.4 shows the results at the distances of 8,000 km and 10,000 km with and without using wavelet transform. These results are compared with the results obtained by the threshold method. It presents averages of SER$\times 10^{-4}$ and BER$\times 10^{-4}$ over ten data sets. Also, it shows the statistical test results (p-value) that measures the probability under the assumption of no difference between the two methods' performance: the threshold method and the support vector machine method. The results at the distance of 9,000 km have been shown in Appendix C.2.1 C.2. For the purpose of comparison, results at distances of 8,000 km and 10,000 km are shown again in the same appendix. Looking at Table 7.4, we can see:

1. The best result I have got so far is the SVM result using the central sample from seven adjacent symbols with a Haar wavelet transform at level 2, which gives BER equal to 168.32×10^{-4} and 289.13×10^{-4} at the distance of 8,000 km and 10,000 km, respectively. P-values of these best results give a strong evidence that the performances of the classifier over the threshold method are not obtained by chance.

2. Using the extracted symbols obtained from db4 wavelet transform does not improve the classification result, and the p-value of the t-test confirms that the results obtained using db4 wavelet transforms (WT) are unlikely to be due to the chance.

3. Among results obtained without using WT, samples from seven consecutive symbols give the best result, even though they only used the central value of each of the seven symbols. This is the results I have discussed in Chapter 6 (Table 6.5c and 6.5d).

Figures 7.5, 7.6 and 7.7 show an example of a single optical transmission symbol at the initial state (blue solid line), and at 8,000 km (red dotted line), respectively. The mid-point of the symbol is the 33^{rd} sample, where the phase is measured, because that represents the highest power level. These figures are selected from the best results using the Haar wavelet at level 2, from the target symbol and three symbols either side.

Figure 7.5 displays an optical transmission symbol that has been mis-classified as class "01" using both SVM and the threshold method, while it belongs to class "00". As we can see from this figure, the middle sample (33^{rd}) is a little bit above the boundary of class "00". Even after applying

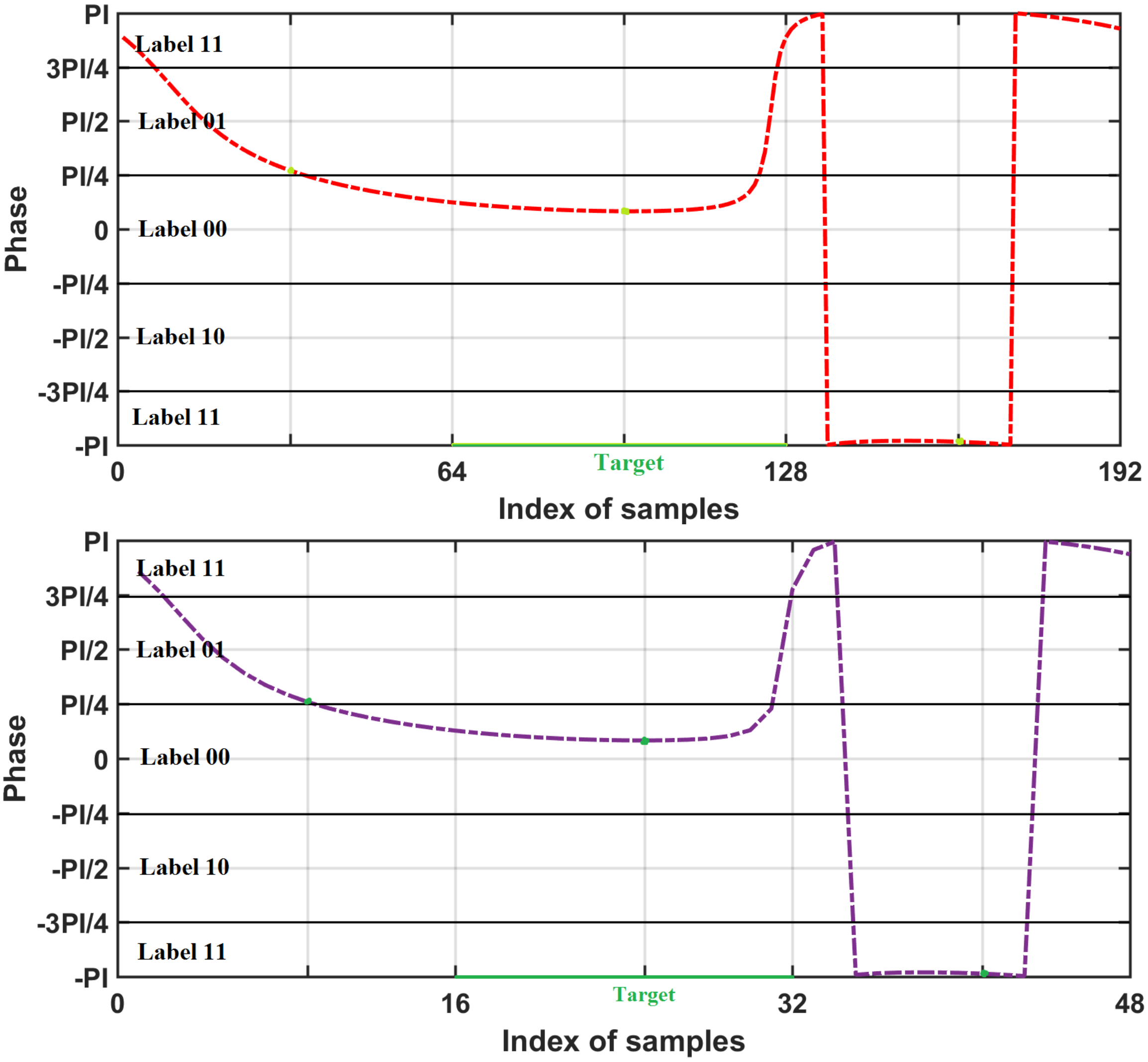


Figure 7.4: The top panel shows an example of using three neighbouring symbols (the target symbol and symbol either side), at 8,000 km, as an input to the classifier. The bottom panel shows the approximation parts of the same symbols. In the case of using only the central sample of three adjacent symbols, the input vector of the target symbol will be the green dots from the top panel (which are (0.8821, 0.2609, -3.0908), at 8,000 km). The central values of approximation coefficients from each symbol (the green dots in the bottom panel) will be used as inputs when using the neighbouring information with wavelet transforms which are (0.8162, 0.2604, -3.0982)). Note: PI denotes π.

Table 7.4: Results of linear SVM with WT on the second type of data

(a) Distance 8,000 km

Method	**No. of symbols and sample**	**No. of features**	**Type and level of (WT)**	**SER** ($\times 10^{-4}$)	**BER** ($\times 10^{-4}$)	**P-Value**
Threshold	1(Central/T)	-	No/0	370.17±14.76	187.27±7.69	
SVM	1(Central/T)	1 × 2	No/0	370.54±16.23	187.41±8.36	0.85
			Haar/1	363.67±16.13	183.97±8.48	0.01
			Haar/2	358.56±15.93	181.51±8.55	0.004
			db4/2	605.40±46.58	307.35±23.55	5.42E-08
	1(Whole/T)	64 × 2	No/0	356.24±13.74	179.84±7.36	0.001
		32 × 2	Haar/1	356.24±13.21	179.93±7.15	0.003
		16 × 2	Haar/2	354.94±12.87	179.33±7.02	0.0005
	7(Central/3P,T,3S)	7 × 2	No/0	339.9±9.8	171.95±5.41	0.000004
			Haar/2	332.65±11.34	168.32±6.14	2.08E-07
			db4/2	425.08±21.79	216.12±11.84	7.23E-06

(b) Distance 10,000 km

Method	**No. of symbols and sample**	**No. of features**	**Type and level of (WT)**	**SER** ($\times 10^{-4}$)	**BER** ($\times 10^{-4}$)	**P-Value**
Threshold	1(Central/T)	-	No/0	607.08±16.68	309.53±9.98	
SVM	1(Central/T)	1 × 2	No/0	610.33±19.01	311.2±10.95	0.06
			Haar/1	604.01±18.35	307.67±10.67	0.1
			Haar/2	593.8±18.47	302.75±10.63	0.01
			db4/2	871.19±25.19	447.39±13.22	2.36E-11
	1(Whole/T)	64 × 2	No/0	588.5±23.71	300.06±13.05	0.01
		32 × 2	Haar/1	586.27±24.62	298.89±13.62	0.005
		16 × 2	Haar/2	584.88±23.95	298.2±13.14	0.002
	7(Central/3P,T,3S)	7 × 2	No/0	574.27±15.81	292.85±9.561	0.000004
			Haar/2	566.84±14.84	289.13±9.05	2.47E-06
			db4/2	694.38±29.46	356.85±15.57	3.43E-08

Table 7.4: The linear SVM results using different input vectors of optical transmission symbols before and after using wavelet transforms at the distances of 8,000 km and the maximum 10,000 km, compared with the threshold method result. SER and BER denote the average of Symbol and Bit Error Ratio over ten data sets. (×2) denotes that each an optical transmission symbol input is represented by ($\sin\theta$,$\cos\theta$). The letters P, T and S denote that preceding, target and succeeding symbol respectively. In $M(X/NP, T, NS)$, M is the number of symbols, X is the position of the samples from each used symbol, and N is the number of symbols that are used from the preceding (P) and succeeding (S) of the target symbol (T). The text in blue refers to the results that are obtained without applying wavelets. The text in red refers to the best wavelet results. Using Haar wavelets at level 2 to represent the signal provides the best SVM results, whereas using db4 gives worse results than the threshold method.

WT, the symbol has not yet been classified correctly. This is one of the cases that using central samples of neighboring symbols does not decode it correctly. However, the same distortion may be correctly classified using the complete set of samples of the symbol, as we have seen in Figure 5.4

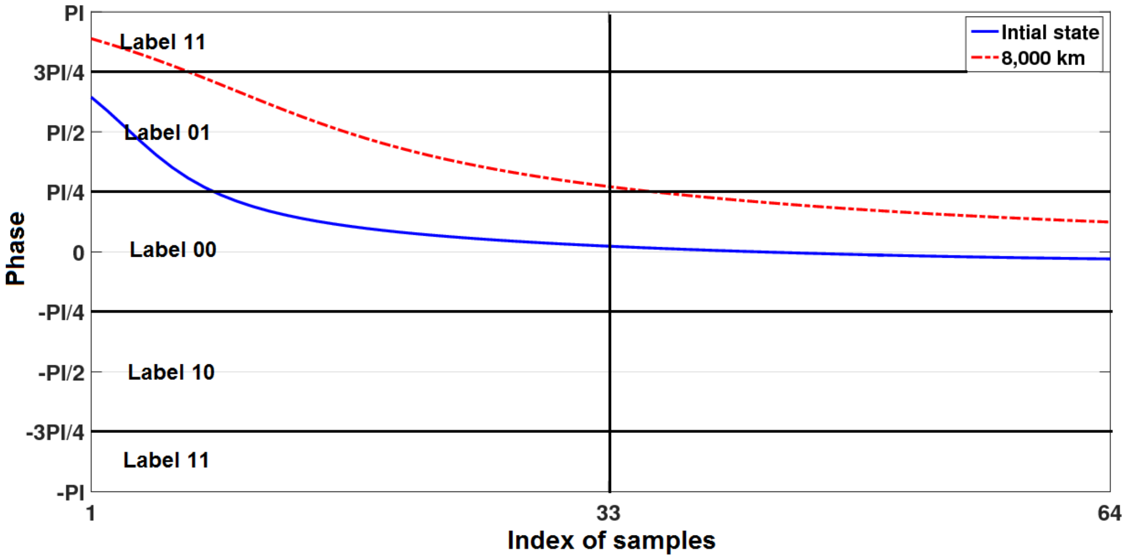


Figure 7.5: An optical transmission symbol from class “00” is classified incorrectly using both linear SVM with the central sample from seven adjacent symbols with Haar transform (level 2), and the threshold method (the first data set in the second type of data). The symbol is received at 8,000 km (the red dotted line), the middle sample (33^{rd}) is a little bit above the boundary of class “00”. This is one of the cases that using central samples of neighboring waves does not solve, whereas it was solved using the complete set of samples of the wave (see Figure 5.4, chapter 5). Note that PI means π.

(Chapter 5).

Figure 7.6 presents another optical transmission symbol that has been detected correctly as class “00”, using both SVM and the threshold method. It can be seen clearly that the symbol did not distorted so much compared with the initial state (the blue solid line).

Figure 7.7 shows one more optical transmission symbol that has been detected correctly using the classifier, with WT as features but incorrectly using the threshold method. As we can see, the symbol should belong to the class “00”, but was mis-classified as class “10” by the threshold method, at the distance of 8,000 km. We can see that the mid-point (33^{rd}) sample is on the boundary between class “00” and “10”, which makes the threshold classifier fail to classifying it. Conversely the linear SVM, which was trained on these types of distortion, classifies this symbol successfully.

Furthermore, Figures 7.8 and 7.10 show an example of optical transmission symbols that is mis-classified using the central sample from seven adjacent symbols, but is classified correctly after applying the WT, respectively. On the other hand, Figures 7.12 and 7.14 show two optical transmission symbols that classified correctly using the neighbouring information, but are mis-classified when using WT, respectively.

Interestingly, the wavelet transform gives a lossless representation for the original signal as shown in Figures 7.9, 7.11, 7.13 and 7.15. In particular using the combination of neighbourhood information and wavelets give much better results than using the threshold method.

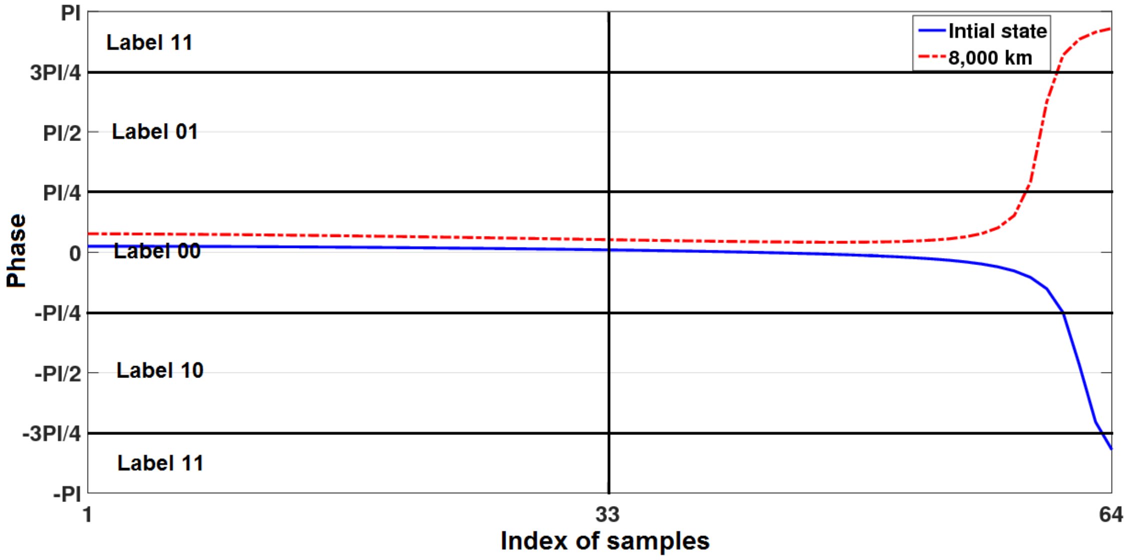


Figure 7.6: An optical transmission symbol has been classified correctly using both linear SVM using the central samples from seven adjacent symbols with Haar transforms at level 2, and the threshold method (the first data set in the second type of data). The symbol is received at 8,000 km (the red dotted line). It is clear that the symbol did not distorted so much compared with the initial state (the blue solid line). Note: PI denotes π.

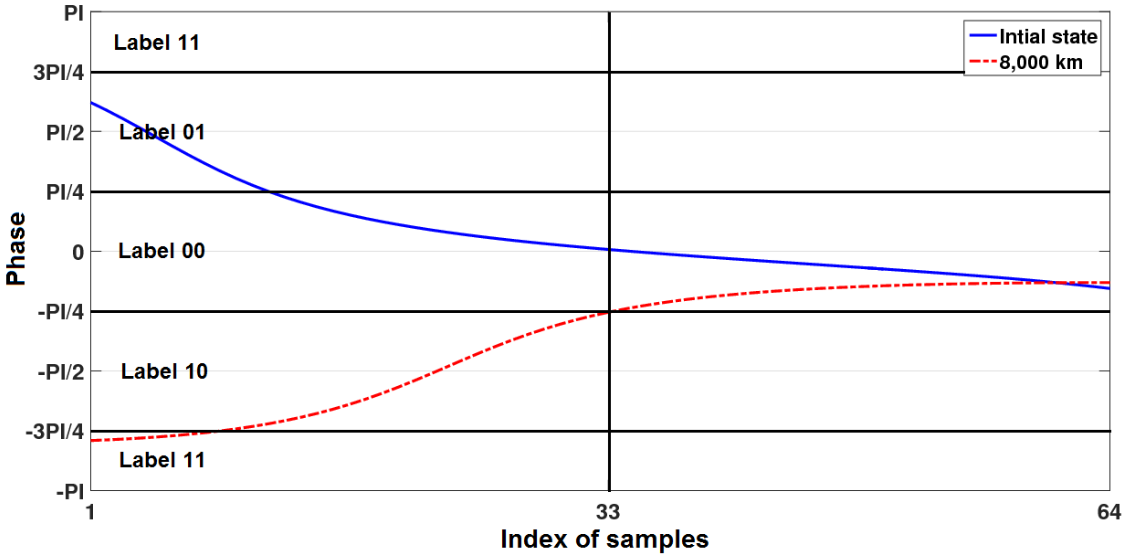


Figure 7.7: An optical transmission symbol has been classified correctly using linear SVM with the central sample from seven adjacent symbols with Haar transforms at level 2, and mis-classified using the threshold method (the first data set in the second type of data). The symbol is received at 8,000 km (the red dotted line). Look carefully at the mid-point 33^{rd} sample, it is on the boundary between class "00" and "10", which makes the threshold classifier fail to classifying it. Conversely the linear SVM, which was trained on these types of distortion classifies this symbol successfully. Note that PI means π.

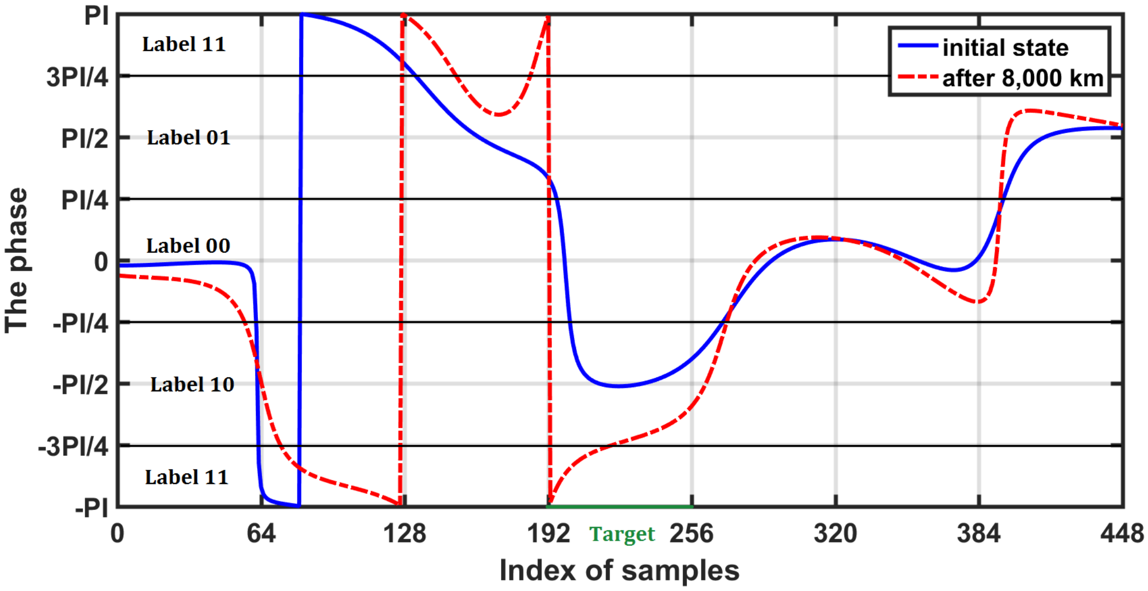


Figure 7.8: An optical transmission symbol from class "10" is classified correctly, using linear SVM with the extracted features with Haar transform (level 2), but mis-classified by using seven adjacent symbols only as class "11". The symbol is extracted from the first data set in the second type of data. The symbol is received at 8,000 km (the red dotted line). The figure shows seven symbols, the target symbol that being decoded is marked in green. Note that PI means π.

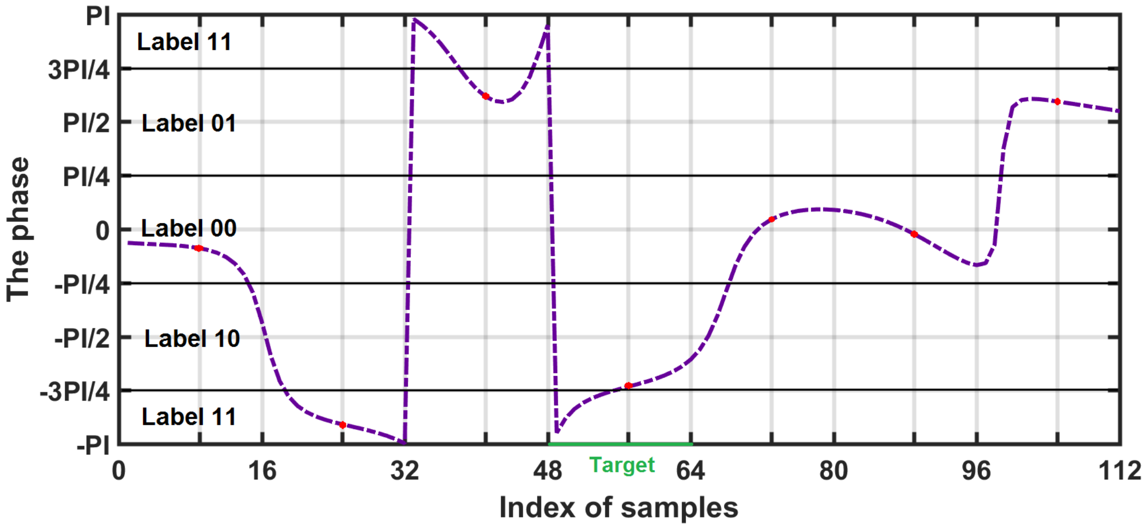


Figure 7.9: The extracted approximation part of seven adjacent symbols (the dotted line in Figure 7.8). The target symbol from class "10" is classified correctly, using linear SVM with the extracted features with Haar transform (level 2), but mis-classified by using seven adjacent symbols only as class "11". The symbol is extracted from the first data set in the second type of data. The symbol is received at 8,000 km (the red dotted line). The figure shows seven symbols, the target symbol that being decoded is marked in green. The input vector consists of the central sample of the target symbol and three symbols either side (the red dots on the mid-point of each sample). Note that PI means π.

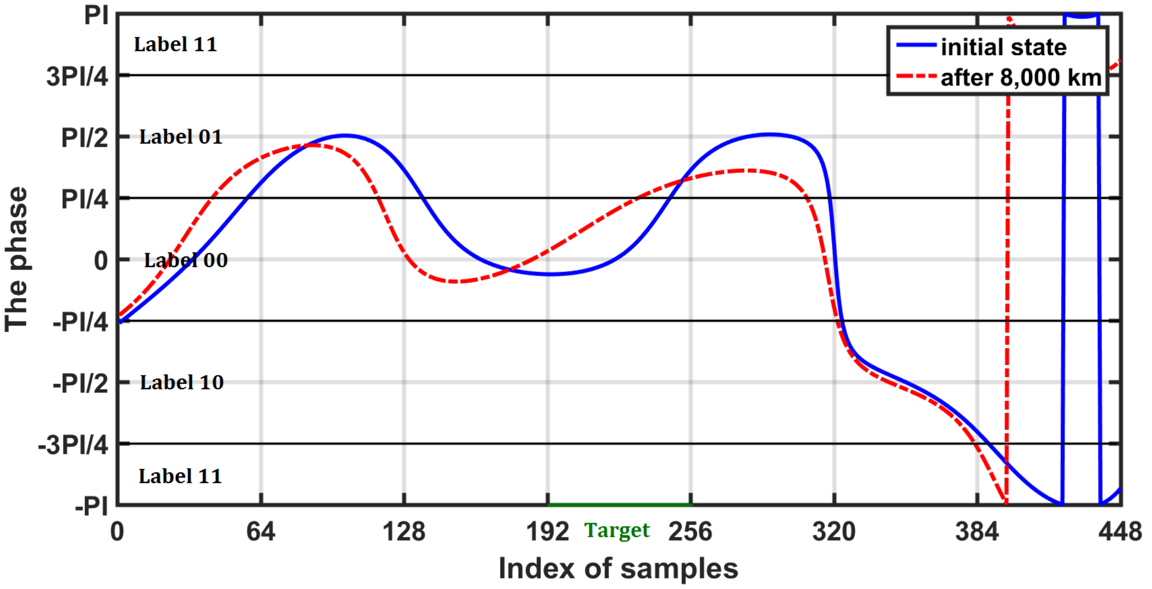


Figure 7.10: An optical transmission symbol from class "00" is classified correctly, using linear SVM with the extracted features with Haar transform (level 2), but mis-classified by using seven adjacent symbols only as class "01". The symbol is extracted from the first data set in the second type of data. The symbol is received at 8,000 km (the red dotted line). The figure shows seven symbols, the target symbol that being decoded is marked in green. Note that PI means π.

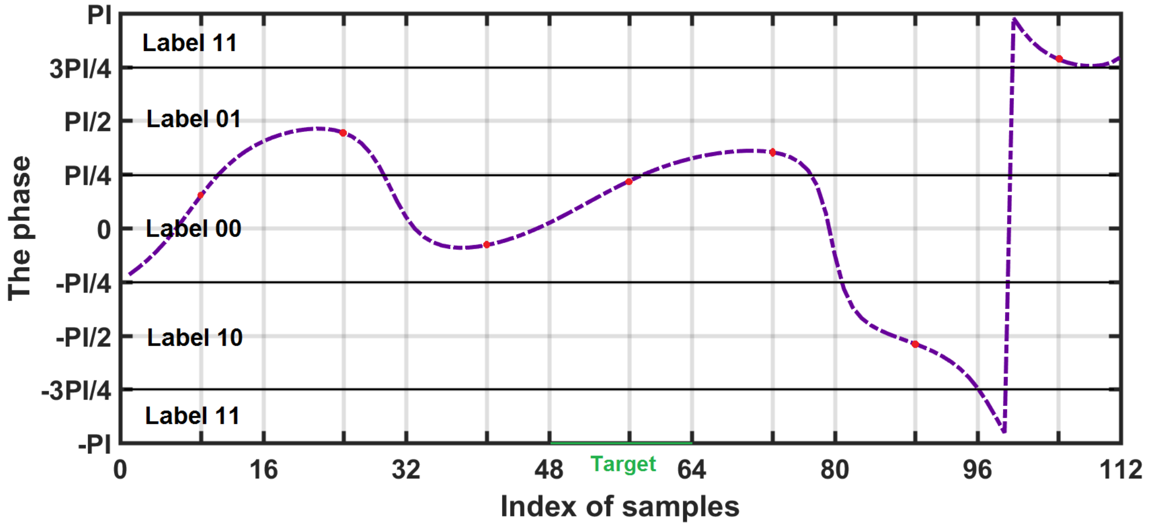


Figure 7.11: The extracted approximation part of 7 adjacent symbols (the dotted line in Figure 7.10). The target symbol from class "00" is classified correctly, using linear SVM with the extracted features with Haar transform (level 2), but mis-classified by using seven adjacent symbols only as class "01". The symbol is extracted from the first data set in the second type of data. The symbol is received at 8,000 km (the red dotted line). The figure shows seven symbols, the target symbol that being decoded is marked in green. The input vector consists of the central sample of the target symbol and three symbols either side (the red dots on the mid-point of each sample). Note that PI means π.

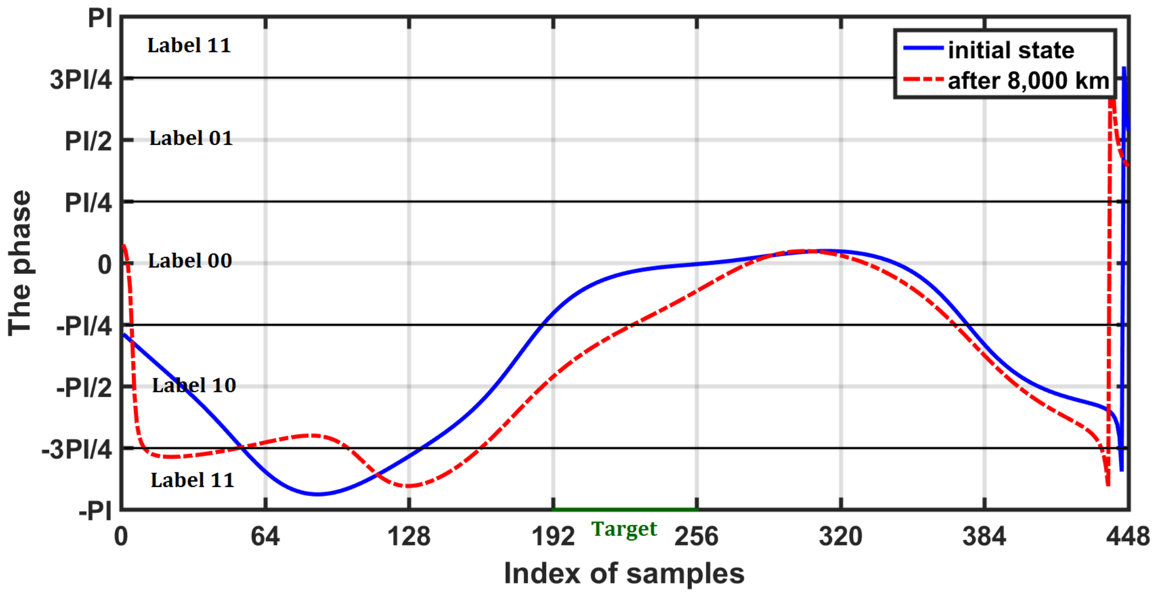


Figure 7.12: An optical transmission symbol from class “00” is classified correctly, using linear SVM with seven adjacent symbols only, but mis-classified by using the extracted features with Haar transform (level 2) as class “10”. The symbol is extracted from the first data set in the second type of data. The symbol is received at 8,000 km (the red dotted line). The figure shows seven symbols, the target symbol that being decoded is marked in green. Note that PI means π.

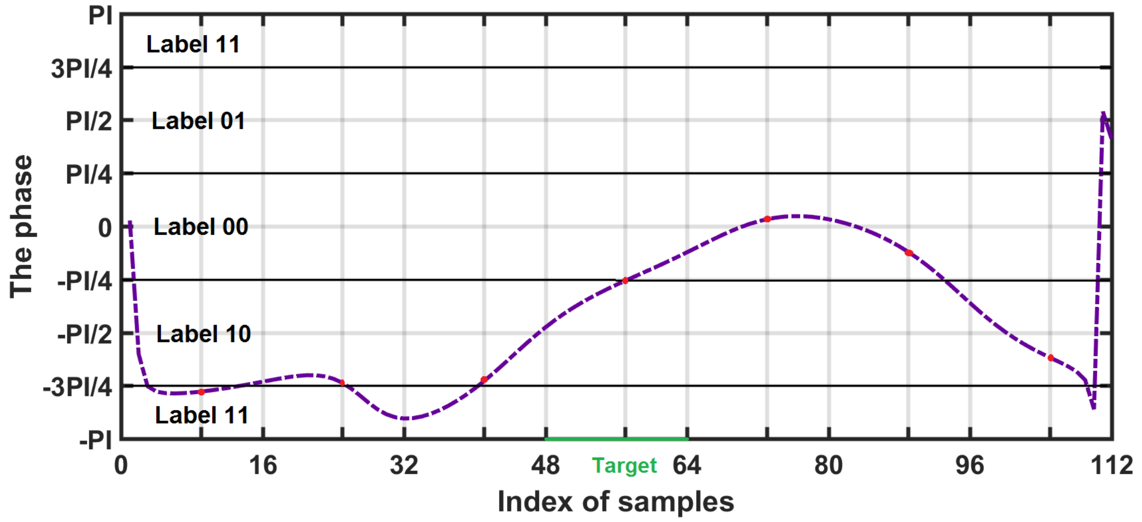


Figure 7.13: The extracted approximation part of seven adjacent symbols (the dotted line in Figure 7.12). The target symbol from class “00” is classified correctly, using linear SVM with seven adjacent symbols only, but mis-classified by using the extracted features with Haar transform (level 2) as class “10”. The symbol is extracted from the first data set in the second type of data. The symbol is received at 8,000 km (the red dotted line). The figure shows seven symbols, the target symbol that being decoded is marked in green. The input vector consists of the central sample of the target symbol and three symbols either side (the red dots on the mid-point of each sample). Note that PI means π.

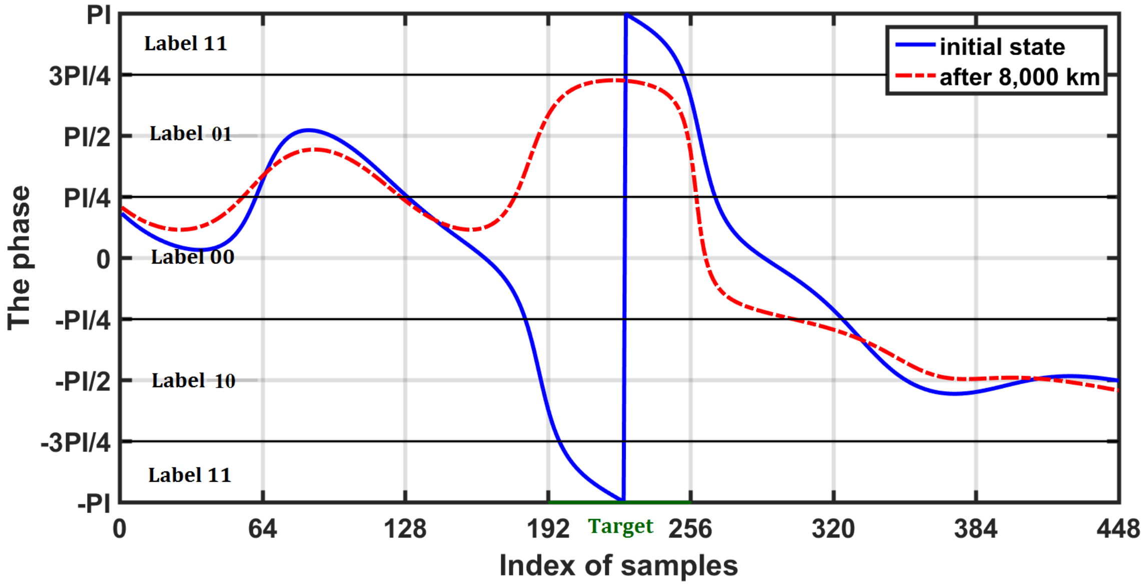


Figure 7.14: An optical transmission symbol from class “11” is classified correctly, using linear SVM with seven adjacent symbols only, but mis-classified by using the extracted features with Haar transform (level 2) as class “01”. The symbol is extracted from the first data set in the second type of data. The symbol is received at 8,000 km (the red dotted line). The figure shows seven symbols, the target symbol that being decoded is marked by green. Note that PI means π.

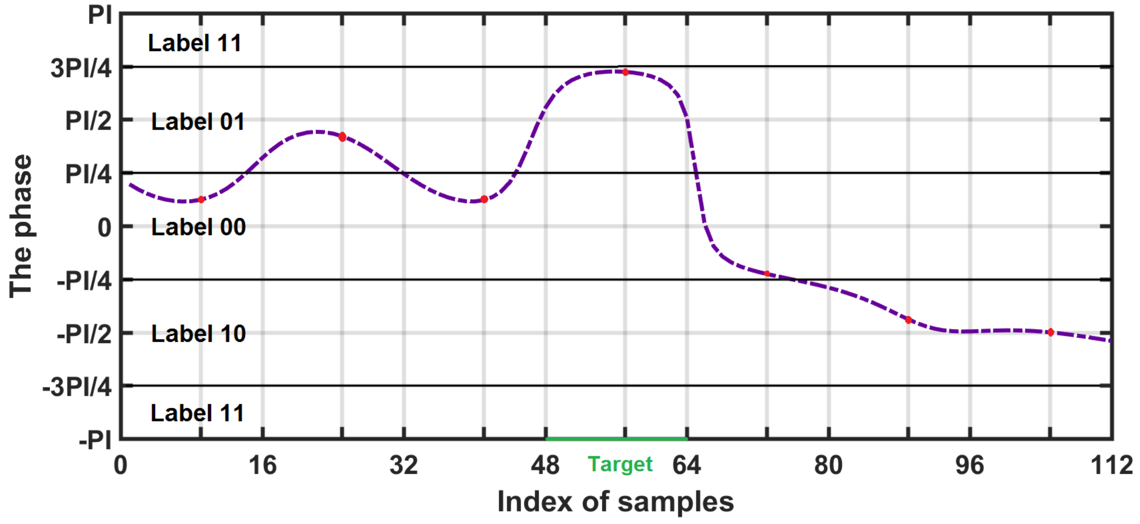


Figure 7.15: The extracted approximation part of seven adjacent symbols (the dotted line in Figure 7.14). The target symbol from class “11” is classified correctly, using linear SVM with seven adjacent symbols only, but mis-classified by using the extracted features with Haar transform (level 2) as class “01”. The symbol is extracted from the first data set in the second type of data. The symbol is received at 8,000 km (the red dotted line). The figure shows seven symbols, the target symbol that being decoded is marked in green. The input vector consists of the central sample of the target symbol and three symbols either side (the red dots on the mid-point of each sample). Note that PI means π.

7.3 Discussion and Conclusion

Results in this chapter show that wavelets are more beneficial with the amplitude distorted data than with the frequency and phase distorted data. From the results obtained using the simple data with frequency noise in Table 7.1, we can see that the use of wavelets does not have any effect on the data with just frequency noise. However, when the amplitude noise is added to signals, wavelets do improve the classification accuracy, compared to results without using WT. Regarding the results obtained using the simple data with phase noise in Tables 7.2 and 7.3, WT also does not show any improvement on signals with only phase noise. However, it provides a slight improvement when applied on signals with both phase and amplitude noise.

When working on simulated optical transmission data, wavelet transforms do have a small effect on the accuracy (for example, see Figure 7.7), and in this field small effects can be worth a lot. However, the best results for distances of 9,000 km and 10,000 km are still worse than the tolerable BER (0.02) for optical transmission system. Although using a Haar wavelet at level 2 brings the mean value of BER$\times 10^{-4}$ at the distance of $10,000$ down from 292.85 to 289.13, we can see there is no big improvement when using WT on this type of data.

Table 7.5 shows the prediction results. The results are obtained from a set of signals in the second type of data that are decoded after travelling 8,000km. The input vector used to the classifier are extracted features (the central sample of the target symbol and three symbols either side), using Haar wavelets at level 2. Comparing with the threshold method (Chapter 6, Table 6.1), the number of two bit errors are increased by one in this experiment when Class “00” was predicted as Class “11” incorrectly (the numbers in red). Figure 7.16 visualises numbers in Table 7.5 in a grouped-bar plot. It shows that the majority of errors are those one-bit errors.

Symbol	Predicted class			
class	00	01	10	11
00	2540	36	40	3
01	56	2610	1	43
10	57	1	2653	27
11	2	39	46	2611

Table 7.5: The number of predicted symbols in each class. The result is obtained from using the linear SVM with the first data set of the second type of data, at a distance of 8,000 km. The test set consists of 10,765 symbols in total. The input vector that is used in the SVM classifier of each symbol being decoded includes the extracted features (the central feature of the target symbol and three symbols either side after applying Haar wavelets at level 2). However there is a small increase in the number of symbols that are classified correctly in each class (in blue) compared with the threshold method results in Chapter 6, Table 6.1, the number of symbol errors that have a two-bit error have increased by one in this experiment (in red). All the improvement comes from correcting the symbol errors that have only a one-bit error.

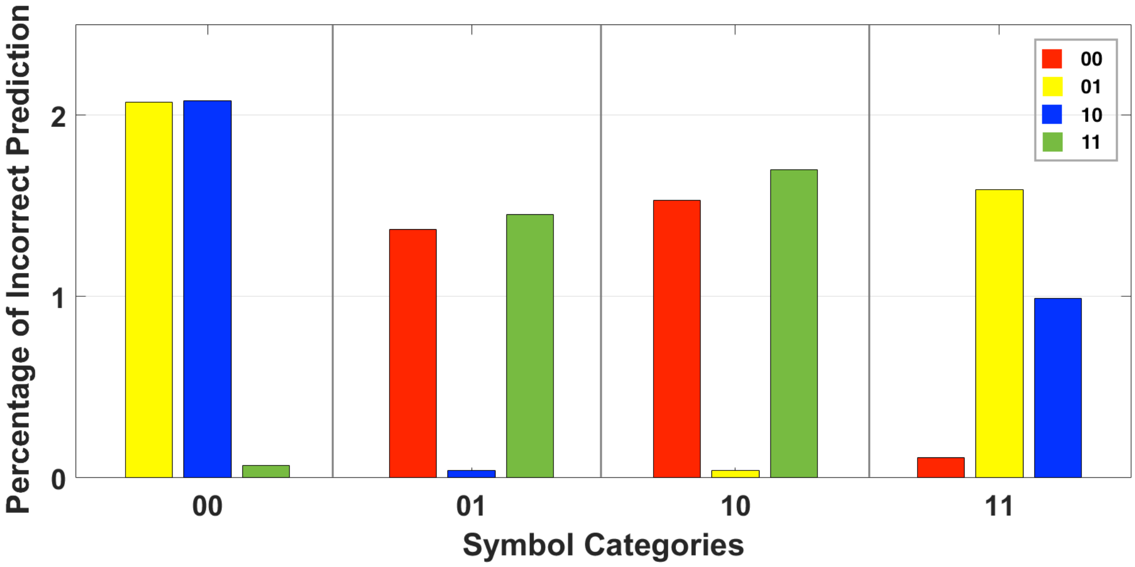


Figure 7.16: The percentage of incorrectly predicted symbols in each class. The result is obtained from using the linear SVM with the first data set of the second type of data, at a distance of 8,000 km. The test set consists of 10,765 symbols. The input vector that is used in the classifier of each symbol being decoded includes the extracted features (the central feature of the target symbol and three symbols either side after applying Haar wavelets at level 2). Table 7.5 shows the same information in numbers.

Figure 7.17 shows the error ratio for each symbol. This is a comparison of the best result in this chapter with the threshold method. Looking at one-bit errors (for example, “01” or “10” in Class “00”), in general, using wavelet transforms gives a slight improvement. The possible reason that wavelet transformation does not work well on the phase modulated data could be because the phase modulated signal does not have a periodic nature, and it is not a wave such as the one shown in Chapter 2 on the bottom panel of Figure 2.7.

Overall, this chapter shows that wavelet transforms can help a little with the noise on phase modulated optical transmission data, however the method does not bring the sort of improvements that the proponents of wavelets led me to believe.

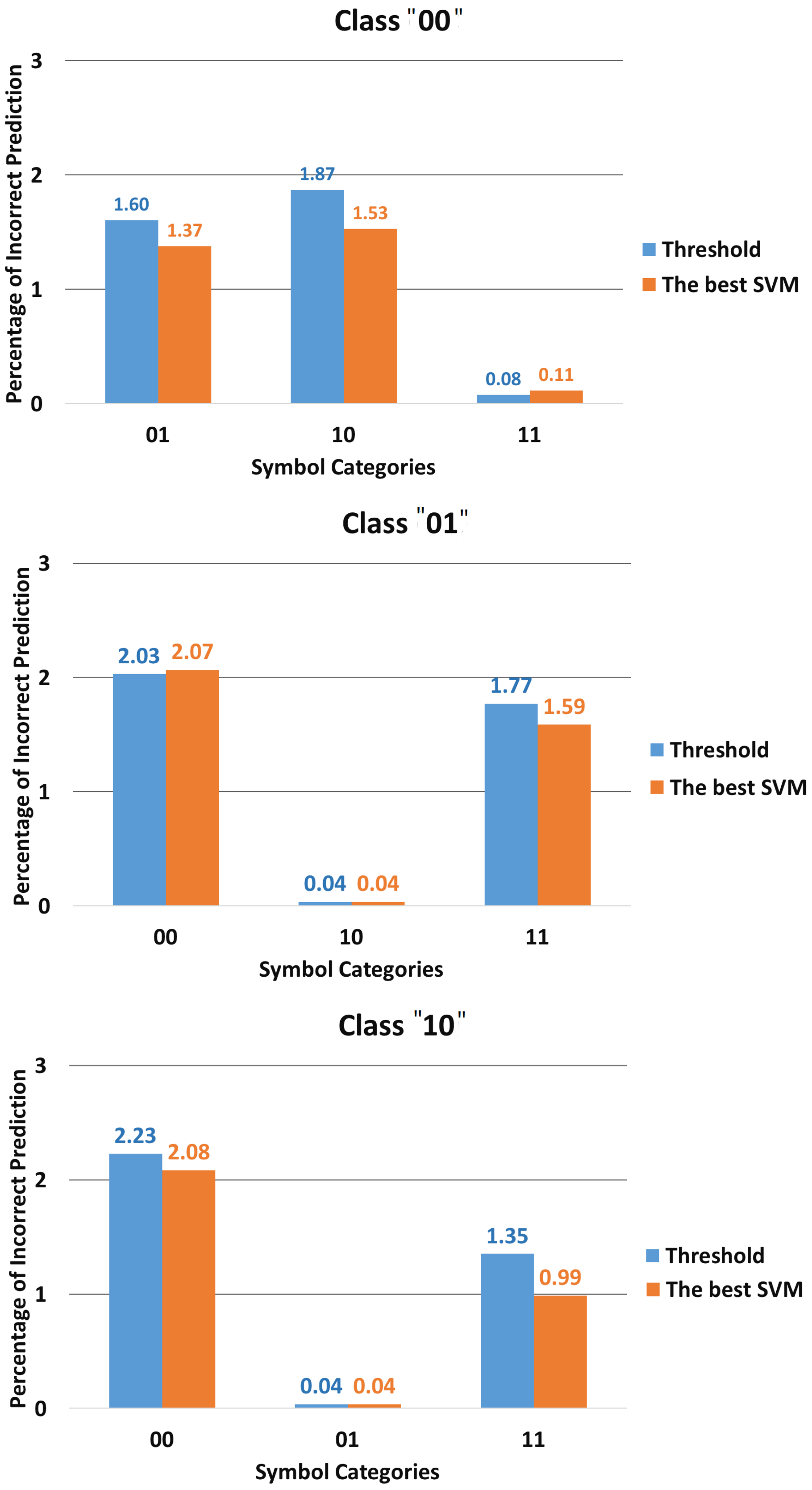


Figure 7.17: Four bar graphs on the percentage of incorrect prediction for each class (continued over page).

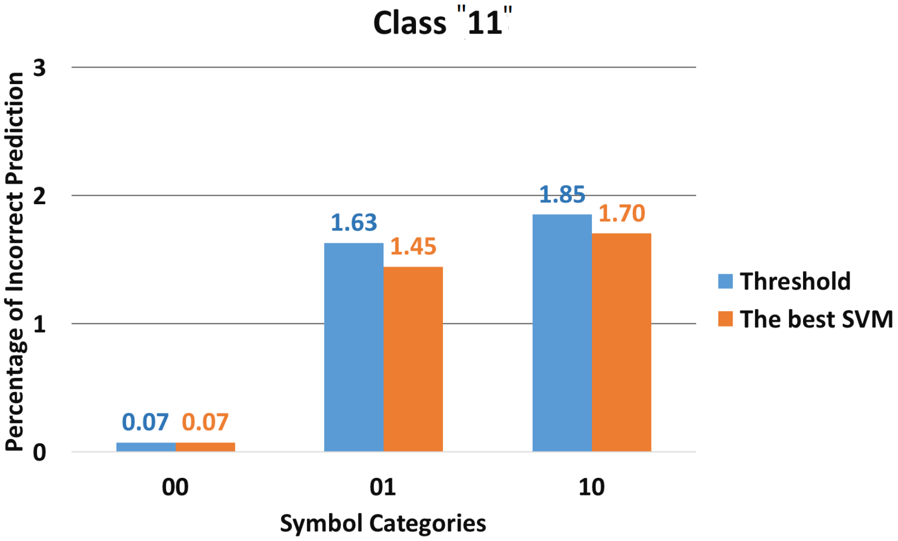


Figure 7.17: Four bar graphs on the percentage of incorrect predicted symbols for each class. The results are obtained using the threshold method (from Chapter 6, Figure 6.2) and the SVM with the first data set of the second type of data, at the distance of 8,000 km. The input vector that is used in the SVM classifier of each symbol being decoded includes the central sample of the target symbol and three symbols either side after extracting them using Haar wavelets, at level 2. Most of symbol errors have only a one-bit error, which resulted from the predicted symbol's class class being predicted as one of the adjacent classes. Some of these errors are corrected using SVM compared with the threshold method. But the symbol errors that have a two-bit error are increased using the SVM classifier compared with the threshold method in Chapter 6.

Chapter 8

Results using Data based on Meaningful Text

In Chapter 7, wavelets do help a little on improving the decoding on the second type of optical transmission data, although this improvement is not very significant. As mentioned in Chapter 3, both the second and third type of data are generated by the same simulated optical link. The second data type consists of a random series of binary bits (0's and 1's) and represents the best attempt to simulate real data traffic, whereas the third type consists of a series of binary bits that is based on meaningful (English) text (see Chapter 3, Section 3.1.2.3). It was seen as an interesting diversion to test my methods on this sort of restricted data traffic. Certainly the use of neighbouring information to train the classifier might prove to be even more effective due to the connection between consecutive letters in a word. Therefore, the aim of this chapter is to investigate how effective a SVM classifier and wavelet transforms might be when working on the optical transmission data that is simulated based on a meaningful text.

As can be found in Chapter 3 (Section 3.1.2.3), the third type of data consists of nine data sets for each distance up to 8,000km, and each data set consists of 16,384 data points. As before each data set is divided into a training set including two-thirds (10,923) symbols, and a test set including the rest of (5,461) symbols. A summary of the experiments that are described in this chapter is shown in the following:

Experiment A: Representing symbols as the central sample or with neighbouring symbols, Section 8.1.

Experiment B: The linear SVM with inputs extracted using wavelet transforms, Section 8.2.

Experiment C: A comparison between the non-linear SVM and the linear SVM with same inputs, Section 8.3.

Experiment D: A comparison of the effect on using the nonlinear classifier with the second type and the third type of data, Section 8.4.

Symbol class	**Predicted class**			
	00	**01**	**10**	**11**
00	1156	36	99	1
01	79	1654	10	192
10	162	11	1074	77
11	0	103	20	787

Table 8.1: The number of predicted symbols in each class. The result is obtained from using the threshold method with the first data set in the third type of data, at a distance of 8,000 km. The test set consists of 5461 symbols in total. The number of predicted symbols that are classified correctly in each class (in blue) is much higher than the number of symbol errors. Most of the symbol errors have a one-bit error. The number of symbol errors that have a two-bit error for each class is shown in red.

Experiment E: An investigation on the relation between symbol errors and the number of training examples, Section 8.5.

As shown in previous chapters, all tables showing experimental results present both the Symbol and Bit Error Ratio (SER and BER). In this chapter, each of these values is an average over nine data sets. Again, each table shows the statistical test results (p-value) measuring the probability under the assumption of no performance difference between the threshold method and the SVM method.

Before I show the main results in this chapter, first I look into the results obtained using the threshold method. Table 8.1 shows these results, and displays the number of predicted symbols in each class. This result is obtained using the first data set at the distance of 8,000 km from the third type of data. We can see that:

- The number of symbols that are classified correctly (the number in blue) for each class is much higher than the number of errors.

- Most of the symbol errors come from predicting the symbol's class as one of the adjacent classes (See Figure 3.1), which is a one bit error. For example, predicting Class “00” as either “01” or “10”, but less likely to be Class “11”.

Figure 8.1 shows the number of mis-classified symbols within each class in a grouped-bar plot. Comparing with Figure 6.2, we can see that the percentage of errors in the third type of data is much higher than those in the second type of data (where all percentages are less than 3%).

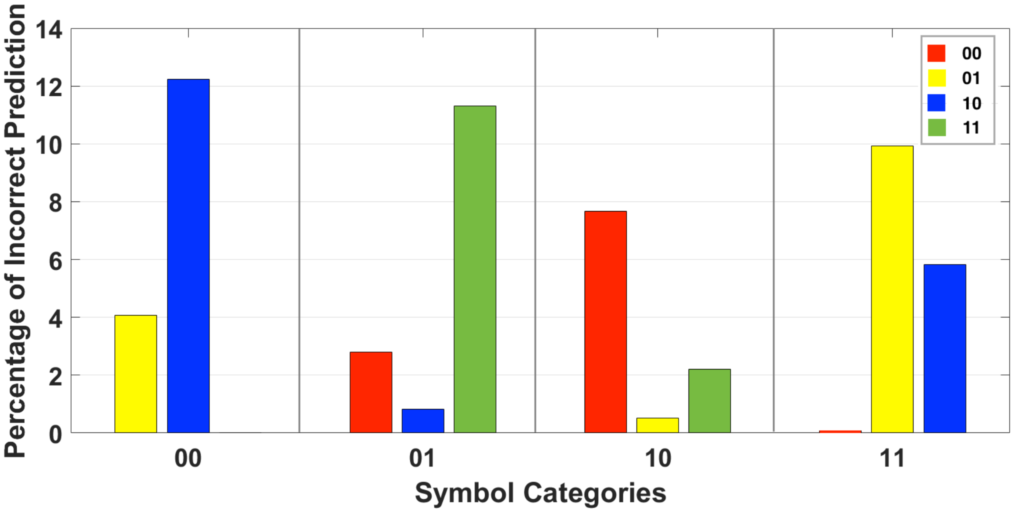


Figure 8.1: The percentage of incorrectly predicted symbols in each class. The result is obtained from using the threshold method with the first data set of the third type of data, at a distance of 8,000 km. The test set consists of 5461 symbols. Table 8.1 shows the same information in numbers.

8.1 Experiment A: Representing Symbols as the Central Sample or with Neighbouring Symbols

The aim of this section is to investigate whether or not different representations of the data can improve the BER with the third type of optical transmission data. Since the best result obtained on the second type of optical transmission data is when using the central sample from the target symbol and three adjacent neighbouring symbols either side, I have decided to investigate the effect on decoding by using the central sample from the target symbol together with the central sample of the neighbouring symbols.

Now the decision that I need to make is the maximum number of neighbouring symbols I should use in this experiment. I have to make a trade-off between getting a good BER and staying with a low computational cost. On the one hand, the classification process may be slowed down due to the need to use a long buffer on the input. On the other hand, the BER may be improved as the number of neighbouring symbols is increased. I have decided that the maximum number of adjacent symbols I would like to consider is twenty-one due to the fact that the average number of letters in a word in English is about five letters (in fact, it is 5.1 letters according to (Bochkarev et al., 2015)), which is equal to twenty symbols (Note: each character consists of eight bits, which are four symbols. So, the average number of bits in a word in English is forty bits, which are twenty symbols).

The input vectors that I use herein are: either the complete set of samples of each symbol, or the central sample from the target symbol and from one to six and ten adjacent neighbouring symbols either side, respectively. All these results are compared with those obtained using the threshold method.

Table 8.2 shows the results at the distances of 3,000 km, 5,000 km and the maximum distance of 8,000 km. Note that all results at distances of 1,000 km to 8,000 km are shown together in Appendix C.3.1. Looking at the table, findings can be summarized as follows:

1. Using only the central sample of each symbol (in green) as an input vector gives a similar result to the threshold method (in blue) over all distances. The statistical t-test provides evidence that the performance difference between these two methods are likely obtained by chance, since the p-values are higher than 0.05 at the distance of either 5,000 km or 8,000 km.

2. Using the complete set of samples of each symbol gives a much better result compared with the threshold method over all three distances. For example, at the distance of 8,000km, the mean value of BER is decreased from 706.12×10^{-4} (using the threshold method) to 599.81×10^{-4} (using SVM with the complete set of samples).

3. Furthermore, in general the BER has improved as the number of the neighbouring symbols is increased. It might be because increasing the number of neighbouring symbols to train a SVM model allows the classifier to learn the frequency with which one letter follows another, for example 'h' often follows 't'. For instance, a classifier that has learned a three-letter word's structure, such as, 'the', when it tries to predict the middle letter between the preceding letter (that is, t) and the succeeding letter (that is, e), it will predict that the letter should be 'h', and hence it will predict the mid-bits correctly quite often. In natural language, the order with which letters occur is not random. This result is as I would have predicted predicted before analysing the data, namely that the SVM can additionally learn letter order when given text based data.

4. The best result I have obtained so far is when using the classifier with twenty-one central samples (the central sample from the target symbol and ten symbols either side). In general, using the central sample from twenty-one adjacent symbols shows a considerable improvement on the BER compared with using the threshold method, despite of the fact that the BER is still greater than the tolerable threshold that is 200×10^{-4} at the long distances like 8,000 km (see Table 8.2c). For example, comparing with the threshold method, the BER at the distance of 5,000 km (Table 8.2b) is improved from 386.99×10^{-4} to 127.3×10^{-4} whereas the BER at the distance of 8,000 km (Table 8.2c) is improved from 706.12×10^{-4} to 338.37×10^{-4} when using the SVM with twenty-one central samples.

Table 8.2: Results using a Linear SVM

(a) Distance 3,000 km

Method	**No. of symbols and samples**	**No. of features**	**SER** ($\times 10^{-4}$)	**BER** ($\times 10^{-4}$)	**P-Value**
Threshold	1(Central/T)	-	350.77±39.35	176±19.75	
SVM	1(Whole/T)	64 × 2	256.36±28.76	129.81±14.81	9.96E-06
	1(Central/T)	1 × 2	355.04±37.17	178.13±18.65	0.02
	3(Central/P,T,S)	3 × 2	225.27±18.72	113.35±9.29	4.53E-07
	5(Central/2P,T,2S)	5 × 2	203.33±20.32	102.28±10.22	5.13E-07
	7(Central/3P,T,3S)	7 × 2	156.55±21.82	78.99±11.09	1.15E-08
	9(Central/4P,T,4S)	9 × 2	127.46±22.95	64.34±11.9	8.66E-08
	11(Central/5P,T,5S)	11 × 2	133.8±16.19	67.41±7.9	7.05E-08
	13(Central/6P,T,6S)	13 × 2	104.7±16.43	52.86±8.12	3.1E-08
	21(Central/10P,T,10S)	21 × 2	73.38±20.67	37.51±10.39	9.32E-09

(b) Distance 5,000 km

Threshold	1(Central/T)	-	762.38±39.66	386.99±20.88	**P-Value**
SVM	1(Whole/T)	64 × 2	568.68±30.15	297.46±18.01	5.52E-08
	1(Central/T)	1 × 2	767.06±35.1	389.22±18.55	0.25
	3(Central/P,T,S)	3 × 2	555.35±43.19	283.37±23.68	5.64E-08
	5(Central/2P,T,2S)	5 × 2	514.95±47.38	264.09±25.64	3.22E-08
	7(Central/3P,T,3S)	7 × 2	434.43±39.57	224.24±22.42	1.33E-09
	9(Central/4P,T,4S)	9 × 2	357.95±37.06	185.29±20.57	9.47E-11
	11(Central/5P,T,5S)	11 × 2	364.33±40.24	189.6±22.23	3.25E-11
	13(Central/6P,T,6S)	13 × 2	316.94±26.7	165.5±14.97	1.27E-10
	21(Central/10P,T,10S)	21 × 2	244.2±32.37	127.3±18.36	2.59E-11

(c) Distance 8,000 km

Threshold	1(Central/T)	-	1367.47±55.89	706.12±28.67	**P-Value**
SVM	1(Whole/T)	64 × 2	1112.54±51.17	599.81±29.68	9.77E-08
	1(Central/T)	1 × 2	1370.12±61.46	707.44±31.93	0.66
	3(Central/P,T,S)	3 × 2	1130.65±55.58	590.05±30.99	2.81E-08
	5(Central/2P,T,2S)	5 × 2	1075.7±57.17	563.9±30.41	1.39E-08
	7(Central/3P,T,3S)	7 × 2	961.08±60.37	507.1±32.26	2.22E-10
	9(Central/4P,T,4S)	9 × 2	843.56±51.75	446.11±28.05	1.09E-10
	11(Central/5P,T,5S)	11 × 2	843.52±52.33	445.89±27.3	2.29E-11
	13(Central/6P,T,6S)	13 × 2	765.05±54.64	406.46±29.23	7.17E-11
	21(Central/10P,T,10S)	21 × 2	634.54±53.54	338.37±29.88	2.17E-10

Table 8.2: The linear SVM results at the distances from 3,000 km, 5,000 km and the maximum 8,000 km, compared with the threshold method result. SER and BER denote the average of Symbol and Bit Error Ratio over nine data sets, respectively. ($\times 2$) denotes that each symbol is represented by ($\sin\theta$,$\cos\theta$). The letters P, T and S denote the preceding, target and succeeding symbol, respectively. In $M(X/NP,T,NS)$, M is the number of symbols, X is the position of the samples from each used symbol, and N is the number of symbols that are used from the preceding and succeeding of the target symbol T. The text in blue refers to the results that are obtained using the threshold method. The text in green refers to the results that are obtained using the SVM with only the central sample of the target symbol. The text in red refers to the best results. Using only the central sample of the target symbol does not show any BER improvement over the threshold method. Using more than one sample from the target symbol (e.g the complete set of samples) improves the BER. However increasing the number of the neighbouring central samples improves the BER even more. Looking at the second and the fourth (BER) columns, each time I increase the neighbouring central samples, the BER is further improved.

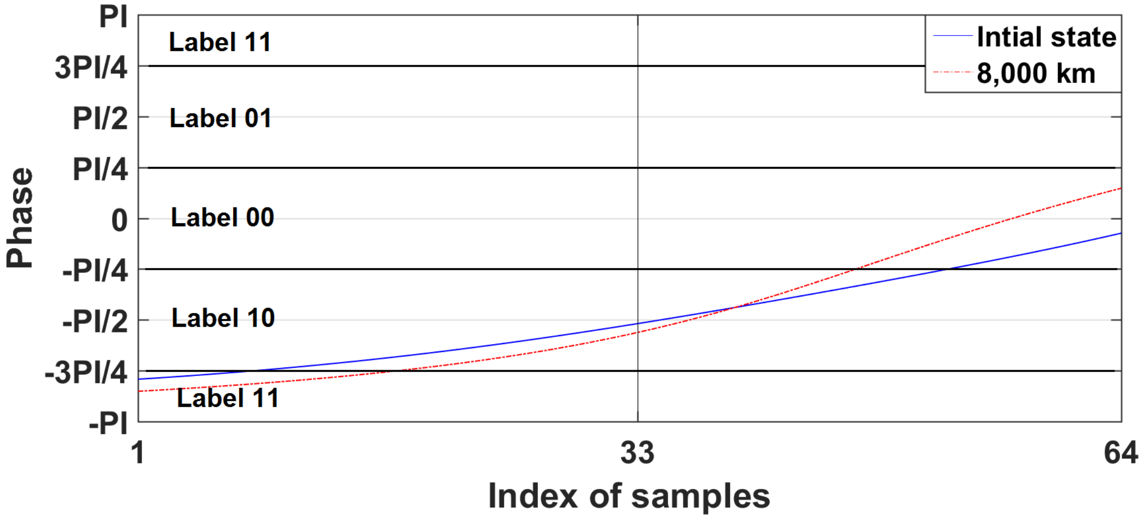


Figure 8.2: An optical transmission symbol from class “10” is classified correctly using either linear SVM with the central sample from twenty-one adjacent symbols and the threshold method. As we can see the middle sample of the symbol after 8,000 km (the red dotted line) is not distorted too much. Note that PI is π.

In the following I shall show three symbol examples in Figures 8.2, 8.3 and 8.4. The classifier has correctly classified two of them, but not the third one.

Figure 8.2 shows an symbol where the middle sample after 8,000 km (the red dotted line) is not distorted too much. This symbol belongs to class “10”, and is classified correctly using either the SVM with the central sample from the target symbol and ten adjacent symbols either side, or the threshold method.

Figure 8.3 shows another symbol from class “10” that is classified correctly using the linear SVM with the central sample from the target symbol and ten adjacent symbols either side, but mis-classified using the threshold method. As we can see from this figure, the target symbol is completely distorted to class “00” rather than the actual class “10”. As a result, the threshold method can not classify it correctly by just measuring the phase of the mid-point of the symbol, which is clearly distorted and pulled up to class “00”. However, the classifier has learned with this kind of noise from using neighbouring information of the target symbol and has been able to predict the symbol’s class correctly.

Figure 8.4 presents the third example of a symbol from class “01”. The middle sample, at 8,000 km (the red dotted line), is slightly below the boundary of class “00” and is in the class of “10”. This kind of distortions is tricky, and can make an irretrievable error. It is classified incorrectly using either the SVM with the central sample from the target symbol and ten adjacent symbols either side, or the threshold method. The threshold method has classified the symbol as class “10”, which is a two-bit error. As for the classifier, because it has learned from training examples having more neighbouring information involved, where the preceding and succeeding symbols have an effect

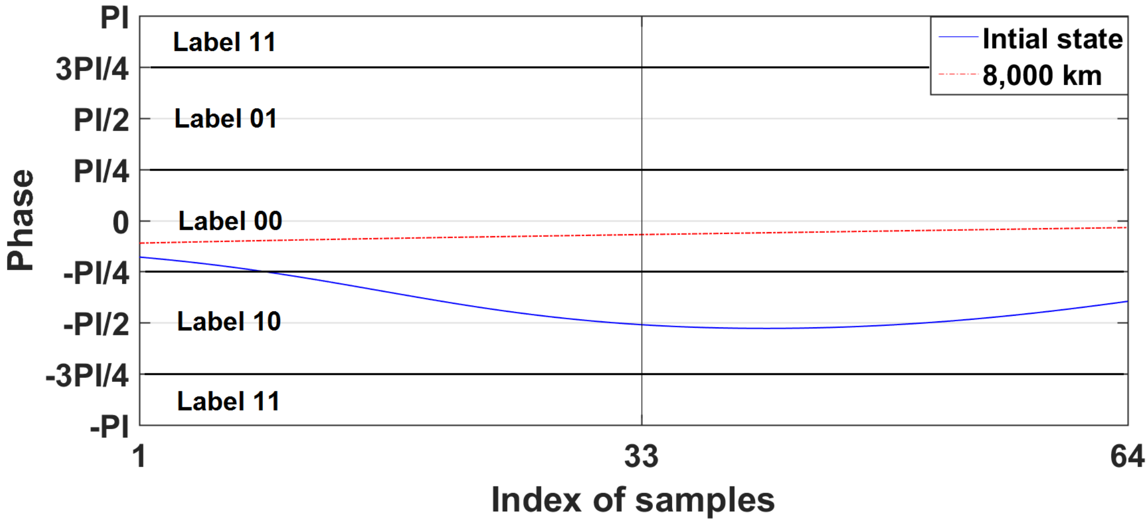


Figure 8.3: An optical transmission symbol from class "10" is classified correctly using the linear SVM with the central sample from the target symbol and ten symbols either side, but mis-classified using the threshold method. The symbol is received at 8,000 km (the red dotted line). Note that PI is π.

on the target symbol, it predicted the symbol as class "00", which is just a one-bit error.

Figures 8.5 presents the BER obtained using either the threshold method, and the classifier with the central sample from the target symbol and ten symbols either side as an input, at the distances from 1,000 km to 8,000 km. The figure shows: 1) The BER increases as the distance travelled by optical transmission data increases. 2) The SVM gives a considerable improvement over the threshold method. Figure 8.6 further shows that the improvements of the classifier over the threshold method for all distances is greater than 50 %.

In summary, using the linear SVM classifier provides a large improvement over the threshold method when the neighbouring information is increased as an input to the classifier, although the BER can not be improved to be less than 0.02 (the tolerable BER in the optical transmission system) at the long distances 7,000km and 8,000km so far. In next section, I will investigate if using the wavelets can improve further the BER or not.

8.2 Experiment B: The linear SVM with Inputs Extracted using Wavelet Transforms

So far, the best BER value is still greater than the tolerable threshold (0.02) at the distance of either 7,000 km or 8,000 km. In this section I focus on the maximum distance 8,000 km only. The aim is to investigate whether or not using wavelet transforms can further improve the BER. First, I use wavelet transforms on three types of signals: just the target symbol, the target symbol with one symbol either side and the target symbol with three symbols either side. I have set the level of WT to 2, hence each symbol is reduced to sixteen approximation coefficients. Then, the central value

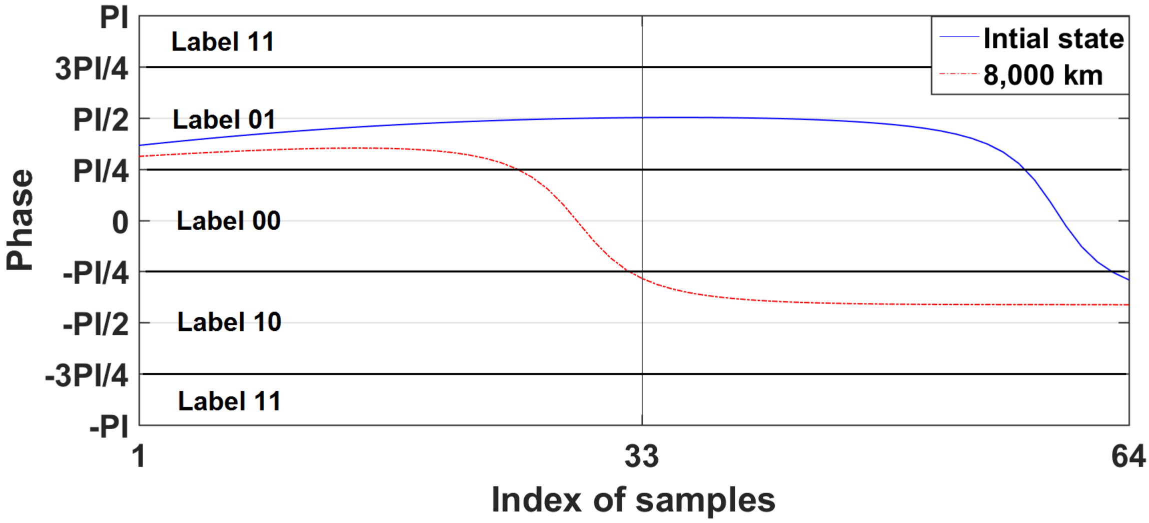


Figure 8.4: An optical transmission symbol from class “01” is classified incorrectly using both the linear SVM with the central sample from twenty-one adjacent symbols and the threshold method. The middle sample, at 8,000 km (the red dotted line), is slightly below the boundary of class “00”. That makes the threshold method classify the symbol as class “10” by measuring just the phase value at this mid-point. This kind of distortions is a tricky, and gives rise to an irretrievable error. The linear SVM classifier, because it has been trained with neighbouring information, classifies it as belonging to the class “00”, representing just a one bit error. Note that PI is π.

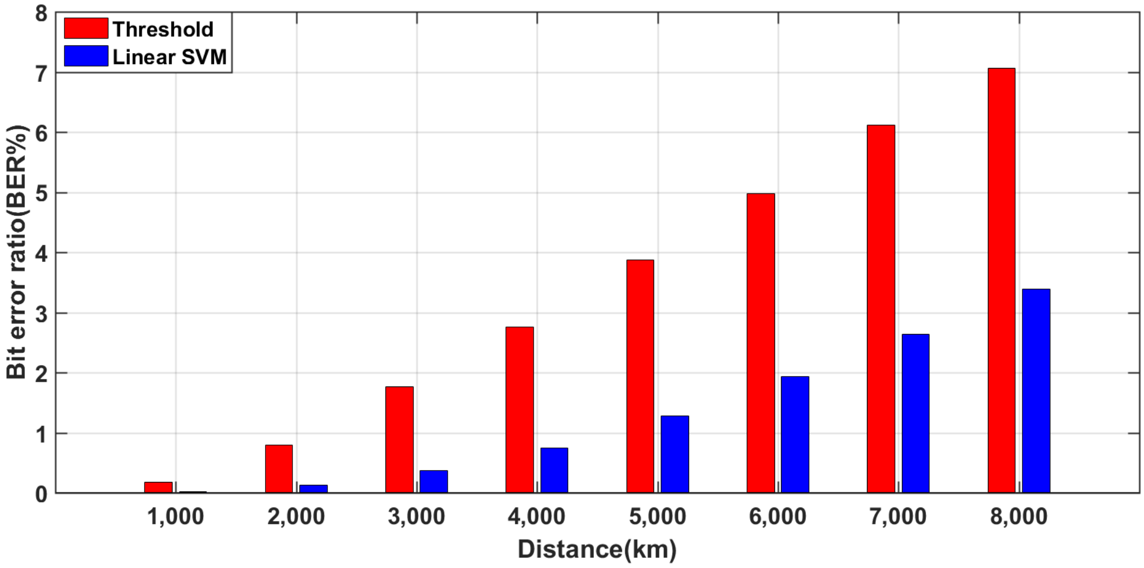


Figure 8.5: The BER (plotted in the bar graph) obtained using the neighbouring information (the central sample of the target symbol and ten symbols either side of each optical transmission symbol) as an input in the linear SVM classifier, over the distances from 1,000 km to 8,000 km. Comparing with the threshold method, the SVM gives a considerable improvement.

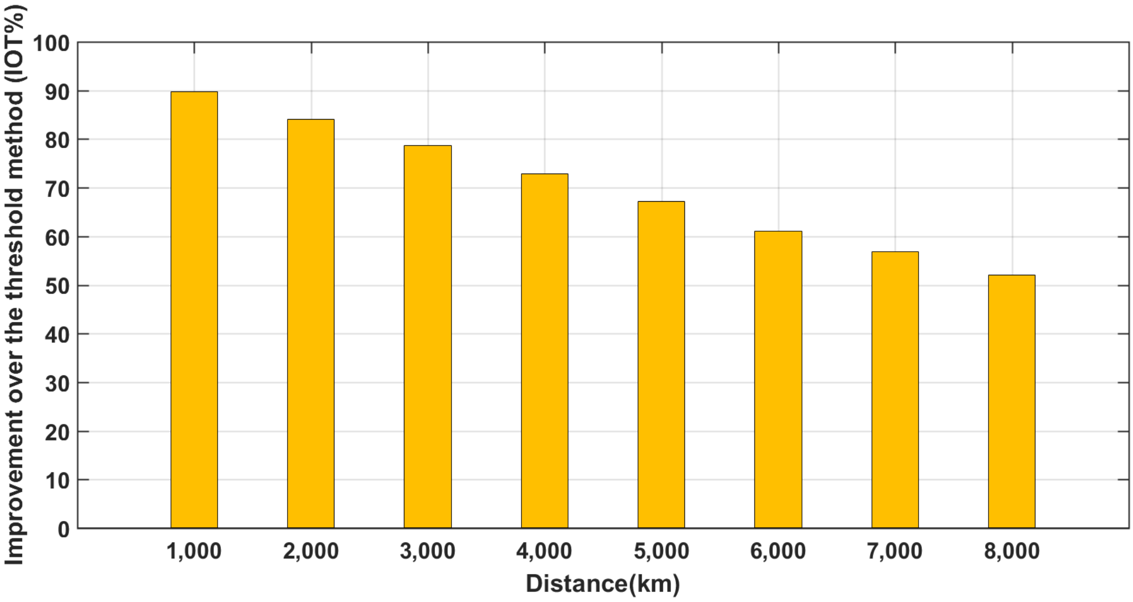


Figure 8.6: The improvement over the threshold method obtained using the central sample of the target symbol and ten symbols either side of each optical transmission symbol as an input in the linear SVM classifier, over the distances from 1,000 km to 8,000 km. The improvement of the SVM classifier over the threshold method is greater than 50% for all the distances.

of the corresponding approximation part for each symbol is used. For example, when applying a level-2 Haar wavelet transforms on a target symbol with three adjacent symbols either side, sixteen approximation coefficients at level 2 for each symbol are obtained. Then, the ninth coefficient (that is the central value) is selected from the target and the three symbols either side. Hence the final wavelet feature vector is of size seven, each element of it is the selected central value from each corresponding approximation part. A selection of different wavelet transformations are tried, which are: original symbol, Haar and db4 at level 2.

Table 8.3 shows the SVM results before and after using the WT at the distance of 8,000 km, compared with the threshold method. I have summarized the following results from the table:

1. The best wavelet result (in red colour) is obtained using the extracted information, where the Haar wavelet is applied on the target symbol and three adjacent symbols either side. These symbols are encoded into Haar wavelet transform at level 2. It can be seen that, the BER is slightly improved from 507.1×10^{-4} (without applying wavelet transforms) to 498.96×10^{-4}. The statistical t-test (with the p-value is equal to 1.33E-10) suggests that using the extracted approximation from WT as inputs to the classifier may be important.

2. Using wavelets on each symbol , that is, without using the neighbouring information, provides much worse results than using the threshold method. The reason for this might be that wavelet transforms extract irrelevant information from the signal in some cases which makes the classification process more complicated.

Method	**No. of symbols and sample**	**No. of features**	**Type and level of (WT)**	**SER** ($\times 10^{-4}$)	**BER** ($\times 10^{-4}$)	**P-Value**
Threshold	1(Central/T)	-	No/0	1367.47±55.89	706.12±28.67	
SVM	1(Central/T)	1×2	No/0	1370.12±61.46	707.44±31.93	0.66
			Haar/2	3380.33±386.99	2821.62±219	1.31E-09
			db4/2	3415.13±58.97	2788.97±58.15	2.44E-13
	3(Central/P,T,S)	3×2	No/0	1130.65±55.58	590.05±30.99	2.81E-08
			Haar/2	1105.41±46.68	576.72±26.27	2.92E-08
			db4/2	1287.55±57.18	678.27±30.59	0.001
	7(Central/3P,T,3S)	7×2	No/0	961.08±60.37	507.1±32.26	2.22E-10
			Haar/2	948.66±49.01	498.96±26.68	1.33E-10
			db4/2	1103.99±62.97	588.74±33.02	2.22E-06

Table 8.3: The linear SVM results using different input vectors of optical transmission symbols before and after using wavelet transforms at the distance of 8,000 km, compared with the threshold method result. The text in blue refers to the results that are obtained without applying wavelets. The text in red refers to the best wavelet results. Using SVM with only the central sample of the target symbol does not show any BER improvement over the threshold method. Using Haar wavelets with the neighbouring information, and increasing the neighbouring information (from three to seven adjacent symbols) helps to improve the BER slightly. Using *db*4 wavelets shows worse results compared with the results obtained without using wavelets, and the threshold method.

The results of the t-test presented in Table 8.3 show that the difference between the threshold and SVM, is unlikely due to the chance, since all p-values but the first one are less than 0.05.

8.3 Experiment C: A comparison between the Non-linear SVM and the Linear SVM with the same Inputs

In this section, I investigate whether or not using a non-linear SVM classifier with the RBF kernel can further improve the BER obtained from the linear SVM. I focus on the maximum distance 8,000 km again.

Table 8.4 presents the results, where for the purpose of comparison, I have shown corresponding linear SVM results again, which have previously been displayed in Table 8.2c. From this table we can see that using the non-linear SVM classifier provides a considerable improvement over the threshold method. In addition, it shows that the BER decreases as the number of neighbouring symbols used increases, especially when the number of neighbours increases from **three** to **nine**. Moreover, the BER drops from 338.37×10^{-4} (in green) by using a linear SVM with the central sample of the target symbol and 10 symbols either side, to 154.3×10^{-4} (in red) by using a non-linear SVM with the same inputs. This BER improvement is the highest I have got so far, it provides a BER of less than 0.02 at a long distance of 8,000 km. Furthermore, the t-test provides a strong indication (where the p-values are less than 0.05) that the presented results obtained in this section are far less likely to be due to chance.

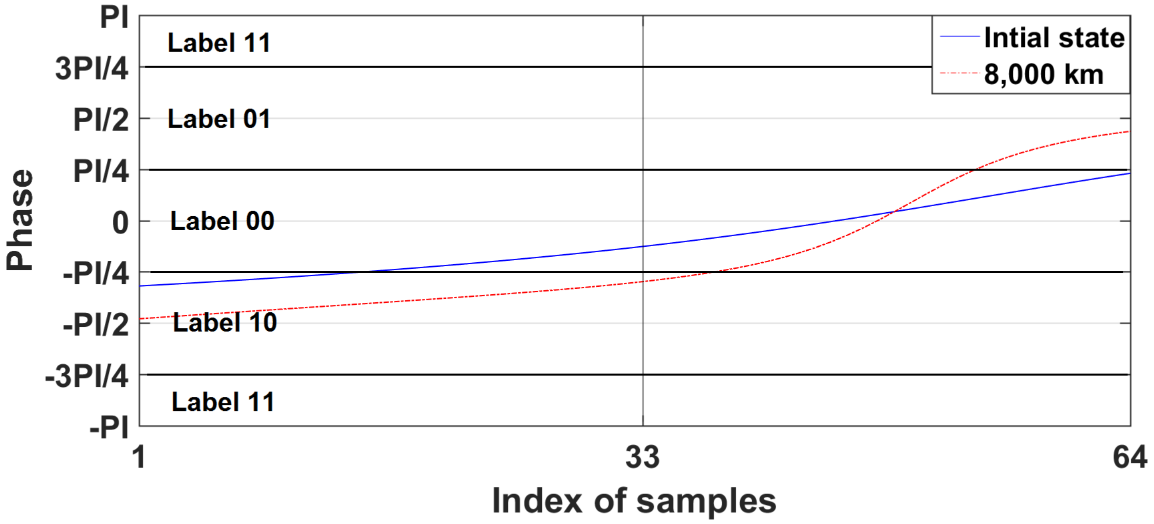


Figure 8.7: An optical transmission symbol from the class “00” is classified correctly using the non-linear SVM with the central sample from twenty-one adjacent symbols, but mis-clasified by the linear SVM and the threshold method. Look at the middle sample, at the distance of 8,000 km (the red dotted line), it is clear that the symbol was pulled down into the area of the class “10” because of the distortion. Note that PI denotes π.

Method	No. of symbols and samples	No. of features	SER ($\times 10^{-4}$)	BER ($\times 10^{-4}$)	P-Value
Threshold	1(Central/T)	-	1367.47±55.89	706.12±28.67	
SVM					
Linear	3(Central/P,T,S)	3 × 2	1130.65±55.58	590.05±30.99	2.81E-08
	9(Central/4P,T,4S)	9 × 2	843.56±51.75	446.11±28.05	1.09E-10
	21(Central/10P,T,10S)	21 × 2	634.54±53.54	338.37±29.88	2.17E-10
RBF	3(Central/P,T,S)	3 × 2	951.77±50.6	500.71±24.15	2.27281E-09
	9(Central/4P,T,4S)	9 × 2	411.91±45.23	221.33±24.22	5.86452E-11
	21(Central/10P,T,10S)	21 × 2	295.15±38.9	154.3±22.37	8.36412E-12

Table 8.4: The linear and non-linear SVM results using the central sample from the target symbol and one, four and ten symbols either side, compared with the threshold method result, at the distance of 8,000 km. The text in blue refers to the result that is obtained using the threshold method. The text in green refers to the best linear SVM result. The text in red refers to the best non-linear SVM result. The non-linear SVM shows a large BER improvement compared with the linear SVM and the threshold method, which improved the BER to be less than (200×10^{-4}). It should be noted that the non-linear SVM provides similar results to the linear SVM with the first type of data in the initial study (see Chapter 5, Section 5.2).

Figure 8.7 shows an optical transmission symbol from class “00” that is classified correctly using the non-linear SVM with the central sample from twenty-one adjacent symbols, but mis-clasified by the linear SVM and the threshold method. Looking at the middle sample in the figure (the red dotted line), it is clear that the symbol was pulled down to the area of class “10” because of the distortion.

Symbol	**Predicted class**			
class	**00**	**01**	**10**	**11**
00	1248	24	15	1
01	25	1889	2	16
10	26	1	1269	26
11	0	32	13	864

Table 8.5: The number of predicted symbols in each class. The result is obtained from using the non-linear SVM with the first data set of the third type of data, at a distance of 8,000 km. The test set consists of 5,451 symbols in total. The input vector that is used in the SVM classifier of each symbol being decoded includes the central sample of the target symbol and ten symbols either side. It can be seen clearly that the number of symbol errors has been decreased in each class compared with the threshold method (see Table 8.1). Note: the numbers in blue, namely the correct ones, are increased when compared to those in Table 8.1. The number of symbol errors that have a two-bit error for each class is shown in red.

Table 8.5 shows the number of predicted symbols in each class. The results are obtained from the first data set of the third type of data, using the non-linear SVM at the distance of 8,000 km. The test set consists of 5451 symbols. It can be seen clearly that the number of symbol errors has been decreased in each class compared with the threshold method (see Table 8.1).

Figure 8.8 visualises the information in Table 8.5 in a grouped-bar plot, where it shows the percentage of incorrect prediction in each class. We can see that the large percentage of incorrectly classified patterns are those one-bit errors, as discussed in Chapter 7.

Furthermore, Figure 8.9 shows a comparison between using the threshold method and the non-linear SVM, at a distance of 8,000 km. It shows four bar graphs of the percentage of incorrect prediction for each class, obtained from the first data set of the third type of data. The input vector that is used to the classifier is the central sample of the target symbol and ten symbols either side. The figure shows the following: 1) The non-linear SVM classifier outperforms the threshold method over all classes. 2) Most of two bit errors given by the threshold method in either Class "01" or "10" are corrected by the non-linear classifier.

8.4 Experiment D: A comparison of the Effect on using the Nonlinear Classifier with the Second Type and the Third Type of Data

Now I shall compare the effect of using a nonlinear classier on the second type of data (random simulated data) and the third type of data (meaningful text). Figure 8.10 shows two bar graphs. The left panel presents a comparison of the BER between the threshold method and the non-linear

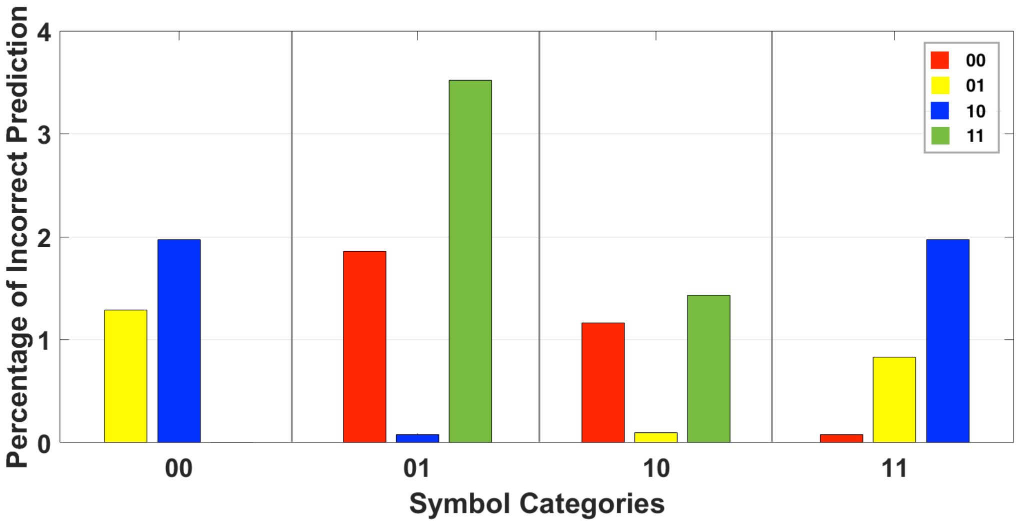


Figure 8.8: The percentage of symbol errors in each class. The result is obtained from using the non-linear SVM with the first data set of the third type of data, at a distance of 8,000 km. The test set consists of 5451 symbols. The input vector that is used in the SVM classifier of each symbol being decoded includes the central sample of the target symbol and ten symbols either side. Table 8.5 gives the same information in numbers.

SVM with both the second and third type of optical transmission data. The input vector to the classifier is the central sample from the target symbol and ten symbols either side. It shows that using the non-linear SVM gives a much better result with the third type of optical transmission data. By contrast, using the non-linear SVM with the same input vector does not show a large improvement over the threshold method with the second type of data. This can be see clearly from the right panel of Figure 8.10, where it shows the improvement over the threshold method (IOT) using both types of data. The IOT obtained using the third type of data is about 60% greater than the one obtained using the second type of the data. For the purpose of comparison, I have shown the detailed results on SER and BER obtained using the second type of data in Table 8.6.

8.5 Experiment E: An Investigation on the Relation between Symbol Errors and the Number of Training Examples

In this section, I shall convert symbol errors to character errors. Alphanumeric characters are used to make words and strings. They include uppercase and lowercase letters, digital numbers from “0” to “9”, and punctuation like “?” and “!”. Table 8.7 shows the number of each case including “spaces” in the training and test set separately. This information is obtained from the first data set belonging to the third type of data. In total, there are 2, 728 characters in the training set and 1, 362 in the test set. As can be seen, the majority of characters are the lower case letters, that is about 80% of the both training set and test set. Seven out of eight numbers of digits have been mis-classified, this is very likely caused by the fact that the number of training examples for digits

Method	No. of symbols and samples	No. of features	SER ($\times 10^{-4}$)	BER ($\times 10^{-4}$)	P-Value
Threshold	1(Central/T)	-	370.17±14.76	187.27±7.69	-
Non-linear SVM, RBF					
SVM	3(Central/P,T,S)	3×2	340.76±11.06	172.56±6.09	1.53E-05
	7(Central/3P,T,3S)	7×2	341.2±12.49	172.64±6.85	1.73E-06
	9(Central/4P,T,4S)	9×2	371.05±57.9	187.71±29.08	0.96
	21(Central/10P,T,10S)	21×2	340.3±11.46	172.34±6.05	1.57E-06

Table 8.6: The non-linear SVM results using different number of central samples from different number of adjacent symbols from the 2^{nd} type of optical transmission data at 8,000 km, compared with the threshold method result. SER and BER are averages over ten data sets. The text in blue refers to the result that is obtained using the threshold method. The text in red refers to the best SVM result. Look carefully at the text in red: the non-linear SVM does not show a good BER improvement over the threshold method (in blue) with the second type of data as when using the third type of data (see Table 8.4).

is too low, that is about $(15/2,728) = 0.05\%$ of the training set. On the contrary, "Space" has the lowest error ratio. This may be because the number of training examples for "Space" is relative high $378/2,728 = 14\%$. Although the total percentage of errors of the lower case letters is not so small as the one for "Space", this must be because there are 26 different lower case letters.

The information in Table 8.7 suggests that increasing the number of training example for each individual alphanumeric character may be of help to reduce the BER.

8.6 Conclusion

Results in this chapter show that importantly using neighbouring information provides a large improvement over the threshold method on the meaningful-text dataset. The BER can be further improved slightly by applying Haar wavelet transform. Moreover, using a non-linear SVM with neighbouring information can dramatically drop down the value of BER. The non-linear SVM decreases the BER less than 0.02 at the long distance of 8,000 km, which is considered to be a significant improvement. However, this may require more hardware support than is currently available. Also, my results suggest that increasing the number of training examples for various characters may improve the BER.

Finally, some overall observations on the non-linear SVM results. This chapter shows that the non-linear SVM dramatically improves the BER on the third type of data (the meaningful text). However, it should be pointed out that the same is not true for the second type of data (random simulated data). This can be seen by comparing the best results at 8,000 km using seven central samples that is contained in Table 6.5, Table 7.4 and Table 8.6. The BER for the linear SVM (Table 6.5) is 171.95×10^{-4}, the BER for the linear SVM with the best Wavelet Transform (Table 7.4) is

Character	**Training set**	**Test set**	**Errors in test set**	**Percentage of errors %**
Upper case	42	32	12	37.50
Lower case	2211	1088	135	12.41
Digits	15	8	7	87.50
Punctuation	72	28	5	17.86
Space	378	206	9	4.37
Total	2728	1362	168	12.33

Table 8.7: The table presents: 1) the total number of the upper, lower case letters, numbers, punctuation and spaces in the training and test sets for each pattern. 2) the total number and percentage of errors for each pattern in the test that mis-classified using the non-linear SVM model (this model gives the best result in this chapter). The result obtained from the first data set belonging to the third type of data. The big percentage of the errors in the test set comes from mis-classifying the numbers patterns which have less frequency in the training set distributed between zero to two and nine. While the "space" pattern has the smallest percentage of errors, but has the biggest frequency for an individual symbol in the training set. Although the total percentage of errors of the lower case letters is not so small as the one for "Space", this must be because there are 26 different lower case letters.

168.32×10^{-4} and the BER for the non-linear SVM (Table 8.6) is 172.64×10^{-4}, so there is little difference in any of the results. Hence the non-linear SVM can extract more information from the training on meaningful data than on random data. This is perhaps not surprising since there are connections between the letters in words in the meaningful data that the SVM can work on. Note also that even increasing the number of central samples to 21, while this helped the meaningful data results it did not help the random data results, as Table 8.6 shows.

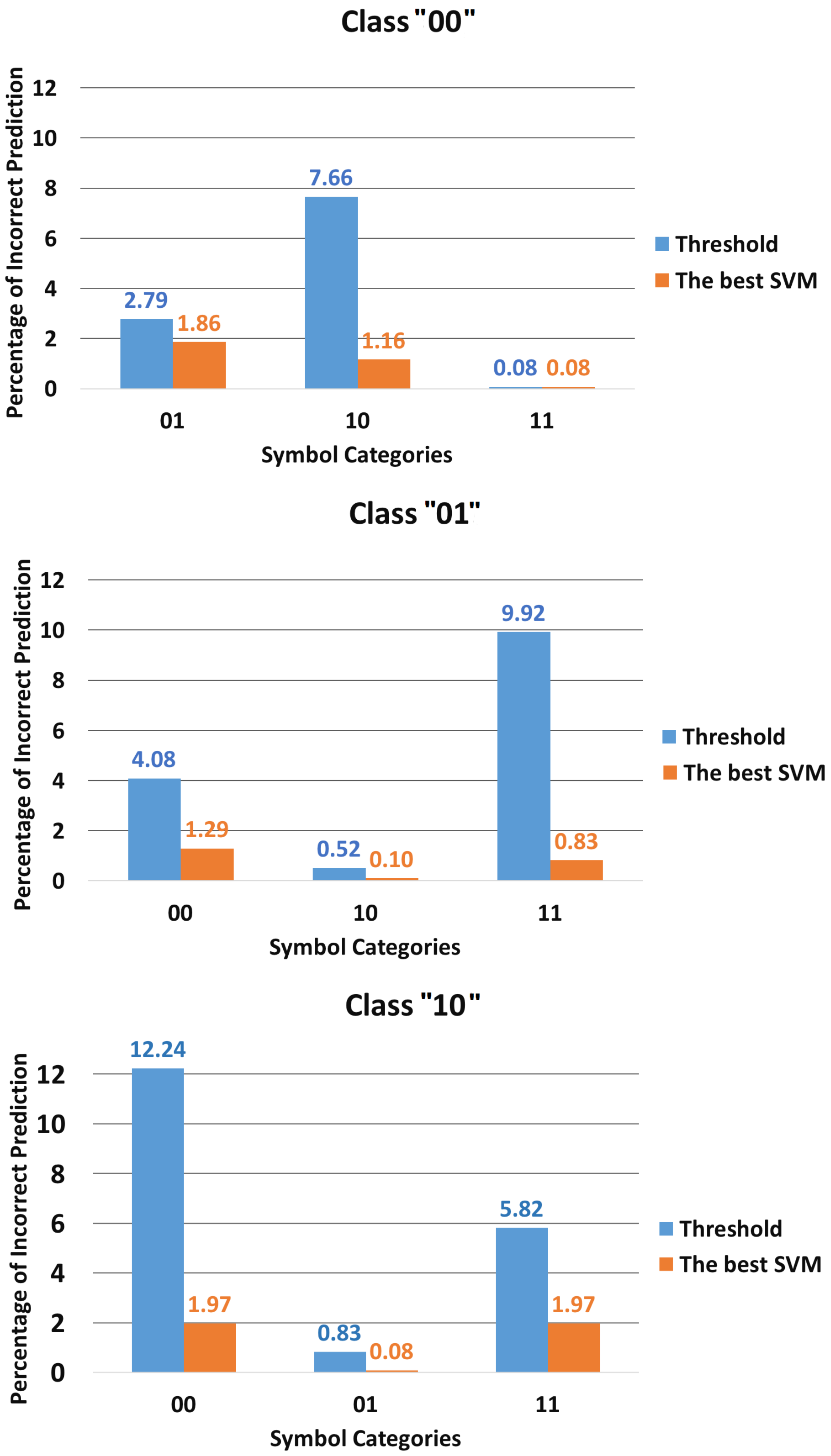


Figure 8.9: Four bar graphs on the percentage of incorrect prediction for each class (continued over page).

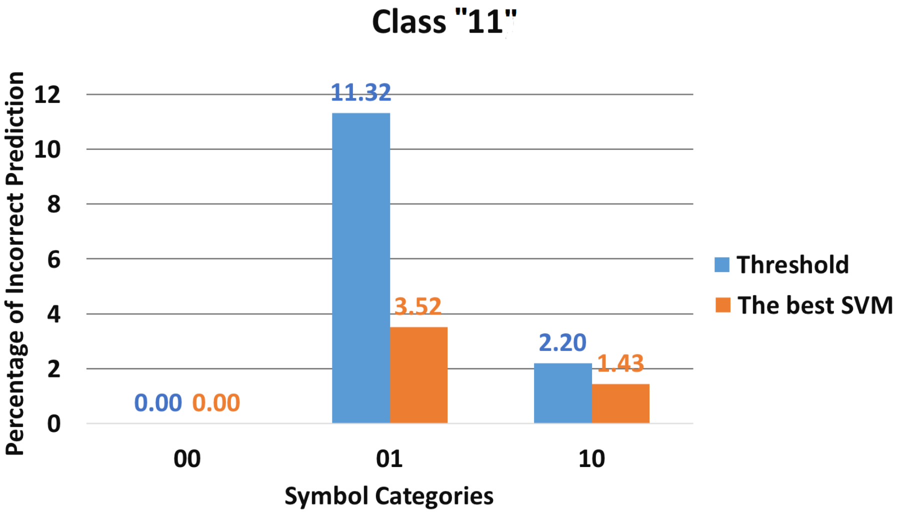


Figure 8.9: Four bar graphs on the percentage of incorrect prediction for each class. The results are obtained using the threshold method and the non-linear SVM with the first data set of the third type of data, at a distance of 8,000 km. The input vector that is used in the SVM classifier of each symbol being decoded includes the central sample of the target symbol and ten symbols either side. The non-linear SVM classifier provides a very good improvement for the symbol error ratio compared with the threshold method. Most of the symbol errors given by the threshold method from Class “01” and “10” are corrected using the SVM classifier.

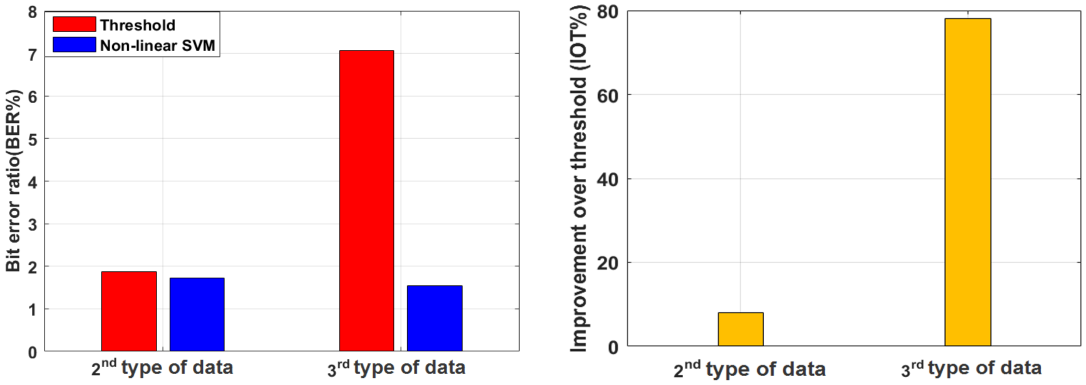


Figure 8.10: The BER and the IOT obtained using both the non-linear SVM and the threshold method with the second and third type of data, at the distance of 8,000 km (plotted in linear bar graph). The input vector that is used in the classifier of each symbol being decoded includes the central sample of the target symbol and ten symbols either side.

Chapter 9

Conclusion

In this chapter I shall summarise the major findings and contributions of my work. In addition, I shall discuss some potential future work. It is worth mentioning that the work shown in this thesis is an interdisciplinary research between Computer Science and fibre optics and optical communications. Undertaking research on both sides has been a challenging experience for me.

9.1 Chapter Summary

This section summaries the main points in each chapter in the thesis.

Chapter 2 is in two sections: The first section gives a brief background of optical transmission communication including a brief history of data transmission and basic information about optical fibres and cables, amplifiers, multiplexers and modulation types that are used in optical communication systems. The second part of **Chapter 2** investigates some of the most relevant papers to my research. It discusses using machine learning techniques (ML) in applications of optical communication systems. The literature review shows that artificial neural networks and support vector machines (SVM) are the most common machine learning methods that have been used in applications of optical communication systems. As shown in (Sun et al., 2008), (Hunt et al., 2008) and (Hunt et al., 2010), an artificial neural network succeeds in reducing the bit error ratio; and the best results were obtained when using one bit either side of the target symbol as an input (neighbouring information). Metaxas et al. (2013) proposed using a linear SVM to classify the optical transmission signals, since the linear separator should be able to be implemented easily in hardware and it offers the high speed required of a de-modulator.

Chapter 3 gives a description of all the types of data that were used for the experiments in this thesis. Also, the chapter introduces the data representation and data pre-processing of the optical transmission data. The optical transmission data is of three types, and was provided by one of my supervisors Dr. Alexey Redyuk from the Institute of Computational Technologies in Novosibirsk, Russia. In addition, I have generated two different types of sinusoidal waves in this study to examine the effect of wavelet transformation on signals with phase or frequency distortion.

In **Chapter 4**, all the methods used in my research are introduced. I present the threshold method, which is currently used in hardware in real-world applications. It is considered as the benchmark method in my research. This chapter also shows the principal component analysis (PCA) that I used to visualize the data. Support vector machines (SVM) are described in this chapter with its two toolboxes. Finally, the performance measurements that evaluate the efficiency of the SVM classifier are presented in this chapter as well.

Chapter 5 provides an initial investigation of the methodology that was chosen to be used in the rest of my research. It illustrates all initial experiments using the SVM on the first type of optical transmission data and is used as a test-bed to indicate what methods to explore further in later work. The main findings are shown as follows:

1. Using an SVM outperforms the threshold method, which is currently used in hardware, and improves the BER significantly.

2. The results confirm that using a linear SVM classifier with only the central sample of the symbol gives a very similar result to the threshold method despite the fact that the SVM is more flexible than the threshold method.

3. Using extra information in the training vector, from the symbol being decode, to the SVM classifier gives more significant improvements.

4. Using the non-linear (the Gaussian kernel (RBF)) and the linear SVM classifier gives very similar results.

5. Moreover, using neighbouring information does not improve the BER more than using just one symbol as I expected from the result in point two.

6. Including the amplitude information to the training vector does not help in improving the result.

I decided at this point not to investigate the use of the non-linear SVM further, since there appears to be no benefit and this is encouraging me to create a method that can easily be built in hardware. Also, I have decided to not use the amplitude information, since using it does not help in improving the BER result. Finally I have pursued using values from neighbouring symbols, despite their mediocre performance in this chapter, since other evidence points to their possible ability.

Chapter 6 presents all the results obtained using the SVM on the second type of the optical transmission data, which is the main and most realistic simulated data. I focus on finding the set of features that give best results. The main findings include:

1. Using information from immediate preceding symbols definitely helps successful decoding. Moreover, when information from the succeeding symbols is also involved, the BER can be further improved.

2. The best linear SVM results I have got in this chapter is the result obtained using the central sample from the target symbol and three symbols either side. However the BER could not be improved to be less than 200×10^{-4} at the distances of 9,000 km and 10,000 km that was desired.

3. There is a decreasing trend of the improvement over the threshold method when the distance travelled by the neighbouring information is increased using only the central sample, see Figure 6.10.

4. The highest two percentage values of mis-classified patterns within each group are the two corresponding adjacent classes of each class. For example, predicting Symbol "00" as Symbol "01" or Symbol "10", but less likely to be Symbol "11". In addition, the SVM gives a slightly better results over all four classes at the distance of 8,000 km on one-bit errors; while both methods (the SVM and the threshold method) perform the same on two-bit errors.

Chapter 7 investigates the effect of applying the wavelet transforms (WT) prior to using an SVM in an attempt to improve the BER of the second type of optical transmission data at the distances of 8,000 km, 9,000 km and 10,000 km. The major findings are shown as follows:

1. The results confirm that wavelets are more beneficial with the amplitude distorted data than with the frequency and phase distorted data.

2. Using wavelet transforms on simulated optical transmission data do have a small effect on the accuracy of the classification, and in this field small effects can be worth a lot.

Overall, this chapter shows that wavelet transforms can help a little with the noise on phase modulated optical transmission data, however the method does not bring the sort of improvements that the proponents of wavelets led me to believe.

Chapter 8 investigates how effective an SVM classifier and wavelet transforms might be when working on simulated optical transmission data that is based on a meaningful text. The major results are shown as follows:

1. Using neighbouring information importantly provides a large improvement over the threshold method on the meaningful-text data-set.

2. The BER can be further improved slightly by applying Haar wavelet transform.

3. Moreover, the non-linear SVM can extract more information from the training on meaningful data than on random data. Therefore, using a non-linear SVM with neighbouring information can dramatically drop down the value of BER. It decreases the BER less than 0.02 at the long distance of 8,000 km on the meaningful text data set, which is considered to be a significant improvement.

The significant improvement in the BER results is not surprising, since there are connections between the letters in words in the meaningful data that the SVM can work on. Note also that even increasing the number of central samples to twenty-one, while this helped the meaningful data results it did not help the random data results, which is the most realistic data in my thesis. Because we know that the real data consists of text, images, videos and audio.. etc, and it does not consist of only text, then random data is the best representation we can use, and is used in the industry (the second type of optical transmission data).

9.2 Contribution

My major contribution is that through a set of comprehensive investigations my work has confirmed that advanced machine learning techniques can be used for improving the bit error ratio (BER) in fibre optical data transmission. My contributions are:

1. I have thoroughly investigated the use of neighbouring information. My results show that it works well on most of my data, even if the data is generated randomly. This is my most important contribution to knowledge.

2. I have shown that the bit error ratio can be improved by using a trainable machine learning model. The linear classifier gives consistently better BER over all distances than the threshold method on both random optical data and meaningful text data. Especially, I have empirically proved that the linear SVM, which is a hardware realisable algorithm, can be of use in improving the bit error ratio.

3. The non-linear SVM decreases the BER to less than 0.02 at the long distance of 8,000km, which is considered to be a significant improvement. However, this may require more hardware support.

4. Despite the widespread use of wavelets in signal processing, I have found little benefit in using them in the context of my work. Wavelets are more beneficial with the amplitude distorted data then with the frequency and phase distorted data.

9.3 Future work

The future work that may be performed after this research includes:

- Dealing with the nonlinear effect during data transmission in fibre optic systems:

  As described in Chapter 3, the optical transmission data is modelled by a nonlinear Schrodinger equation, which is commonly used in simulating the optical channel with nonlinear effects.

  - Consider other algorithms (for example, the non-linear kernel based SVM) if/when hardware implementations that work fast enough can be produced:

    My results obtained using an SVM with a nonlinear (RBF) kernel show a big improvement can be obtained comparing with using the linear SVM on the meaningful text data. Implementation of non-linear SVM on FPGA is still difficult. However, progress can be seen recently. For example: the study in (Machhout and Tourki, 2017) has shown a non-linear SVM has been implemented, though it is designed for the test (classification) phase only, that is, the proposed work does not work for the training stage.

  - Deep learning:

    Deep neural networks (DNN), which can approximate any nonlinear function (Goodfellow et al., 2016), have been shown to achieve state-of-the-art performance for various applications such as image classification and automatic speech recognition. Recently, the increasing efficiency of DNN may allow DNN to implement online adaptation and learning. In (Aoudia and Hoydis, 2018), authors have implemented an optical fibre communication system as an end-to-end deep neural network, including the complete chain of transmitter, channel model, and receiver. However, authors have tested the performance on some relative short distances, that is, less than 100 km.

- Results of Chapter 8 show using more training examples for different characteristics may be of help in improving the bit error ratio. Therefore, more signals involving more realistic characteristics need to be generated to validate our findings.

- Hardware implementation of wavelets on FPGA:

  Although using wavelet transformation gives little improvement on BER, as I have discussed before, a small improvement may be important in this problem domain. So it is worth investigating the progress on implementing wavelets on FPGA. Authors in (Chuma et al., 2017) have proposed an implementation in FPGA for de-noising using Haar wavelet transform to level 5. Their results show that FPGA can give a fast and reliable platform to make the wavelet transform.

- The use of real non-simulated data, and using more modern phase encoded data (including phase encoded data with more phases (QPSK).

Appendix A

Data Description

A.1 Principles of fiber-optic links simulation

The evolution of the optical field $A(z,t)$ in a fiber-optic link based on lumped amplification scheme with erbium doped fiber amplifiers (EDFA) can be represented by the stochastic general nonlinear Shrödinger equation (GNLSE) (Agrawal, 1997):

$$\frac{\partial A}{\partial z} + \frac{\alpha}{2}A + \frac{i}{2}\beta_2\frac{\partial^2 A}{\partial^2 t} - i\gamma|A|^2 A = A\sum_{k=1}^{k=S} G\delta(z-kL) + \sum_{k=1}^{k=S} N\delta(z-kL) \tag{A.1}$$

where $A(z,t)$ is the complex field envelope, z is the distance along the fiber, t is the time, α is the fiber loss, β_2 is the dispersive term, γ is the nonlinear term, G is the gain coefficient of EDFA, S is the number of spans, L is the length of each span. The term $N(z,t)$ is the one describing amplified spontaneous emission (ASE) noise generation. ASE can be represented by the field that has the statistical properties of additive Gaussian noise. The spectral noise density per polarization at a frequency ν is given by $N_{ASE} = (G-1)n_{sp}\hbar\nu$, where n_{sp} is the coefficient of the spontaneous emission and $\hbar$ is Planck's constant (Essiambre et al., 2010).

The initial condition for $A(z,t)$ usually is given by:

$$A(0,t) = \sum_{m=0}^{M} \exp\left[i\omega_m t\right] A_m(t), \tag{A.2}$$

where M is the number of spectral channels, ω_m is the carrier frequency of mth channel and $A_m(t)$ is the complex amplitude of mth channel. $A_m(t)$ can be written as

$$A_m(t) = \sum_{n=0}^{N} c_n^m f(t - nT_b), \tag{A.3}$$

where N is the number of bits in bit stream, $\{c_n^m\}$ is the modulation alphabet of mth channel, $f(t)$ is the pulse shape and T_b is the bit interval.

For example, $A_m(t)$ for QPSK modulation and Gaussian pulse shaping can be written as

$$A_m(t) = \sum_{n=0}^{N} c_n^m \sqrt{P}_0 \cdot \exp\left[-\frac{(t-nT_b)^2}{2T_0^2}\right], \tag{A.4}$$

where $c_n^m \in \{e^{\frac{\pi}{4}}, e^{\frac{3\pi}{4}}, e^{-\frac{\pi}{4}}, e^{-\frac{3\pi}{4}}\}$.

The GNLSE is solved through the split-step Fourier method (SSFM) (Hardin, 1973). The numerical simulations of ASE noise model are described below.

ASE simulation using SSFM

Without the noise term, equation (A.1) can be numerically solved using the SSFM, which requires the division of the fiber into small steps. For each step Δz, the approximated solution is calculated assuming that the linear and nonlinear effects act independently. Then, in the case of noiseless propagation, we can write

$$\frac{\partial A}{\partial z} = (\hat{D} + \hat{N})A, \tag{A.5}$$

where $\hat{D}$ and $\hat{N}$ are the linear and nonlinear operators, respectively. Linear part of (A.1) is solved in the frequency domain whereas nonlinear term operates in the time domain. The optical field at the position $z + \Delta z$ can be approximated by

$$A(z + \Delta z, t) \approx F^{-1}\left\{\exp\left[\frac{\Delta z}{2}\hat{D}\right] F\left\{\exp\left[\Delta z\hat{N}\right] F^{-1}\left\{\exp\left[\frac{\Delta z}{2}\hat{D}\right] F\{A(z,t)\}\right\}\right\}\right\}, \tag{A.6}$$

where $F\{\}$ and $F^{-1}\{\}$ denotes the forward and backward Fourier-transform operation respectively.

In order to include the noise term, we add the noise discretely to the field. This means that the noise that is created in EDFA is approximated by a certain amount of noise that is added to the field at each point $z_k = kL$ of the SSFM. Using these approaches, the total optical field at the position $z_k + \Delta z$ can be written as

$$A(z_k + \Delta z, t) \approx F^{-1}\left\{\exp\left[\frac{\Delta z}{2}\hat{D}\right] F\left\{\exp\left[\Delta z\hat{N}\right] F^{-1}\left\{\exp\left[\frac{\Delta z}{2}\hat{D}\right] F\{A(z_k,t)\}\right\}\right\}\right\} + n(z_k,t). \tag{A.7}$$

SSFM and generation of noise schematically:

1. applying the linear operator $\hat{D}$ in frequency domain over a distance $\Delta z/2$
2. applying the nonlinear operator $\hat{N}$ in time domain over a distance Δz
3. if $z = z_k$: generating a white Gaussian distribution of points $n(z, t_i)$ with mean equal to zero, $\langle n(z, t_i)\rangle = 0$ and variance given by $\sigma^2 = (G-1)n_{sp}\hbar\nu B$, where B is the optical bandwidth of simulation
4. if $z = z_k$: adding the ASE noise $n(z, t_i)$ to the field $A(z, t_i)$
5. applying the linear operator $\hat{D}$ in frequency domain over a distance $\Delta z/2$

A.2 First Type of Data

This section presents some additional information about the first type of data.

This type of data was generated as follows: in the numerical model, a typical return-to-zero (RZ)-QPSK (Note: QPSK is Quadrature Phase Shift Keying) transmitter with 40 Gbaud symbol rate ($T_b = 25$ ps) using 2^{16} pseudorandom binary sequence (PRBS) was simulated. The input pulses were unchirped Gaussian pulses with a duty cycle of 30% (7.5 ps pulse duration) and peak power 10 mW.

The signal channel at 1550 nm was propagated over the fiber along with 10 200 GHz spaced similar cross-talk channels with decorrelated PRBS sequences. Transmission link consisted of 60 spans of single-mode fibre (SMF) 50 km long. The basic parameters of the fibres are $\beta_2 = -21.7$ ps ^{2} km $^{-1}$, $\gamma = 0.00137$ mW $^{-1}$ km $^{-1}$, $\alpha = 0.2$ dB/km. After each EDFA the signal was noise loaded by the white Gaussian noise calculated using a 4.5 dB amplifier noise figure.

At the receiver a Gaussian optical band pass filter was used and the chromatic dispersion was fully compensated by multiplying the Fourier transformed optical field with the reverse dispersion function. For phase estimation, an algorithm based on the 4th-power Viterbi-Viterbi method has been used.

A.3 Second Type of Data

This section shows some additional information about the second type of data.

This type of data was generated as follows: in the numerical model we simulated a typical non-return-to-zero (NRZ)-DP-QPSK (Note: DP-QPSK is Dual Polarization Quadrature Phase Shift Keying) transmitter with 30 Gbaud symbol rate ($T_b = 33$ ps) using 2^{18} pseudorandom binary sequence was simulated. Input average power was 0.5 mW per channel.

The signal channel at 1550 nm was propagated over the fiber along with 10 50 GHz spaced similar crosstalk channels with decorrelated PRBS sequences. That way we focus on investigation of modern 100 Gb/s channels using DP-QPSK modulation format operating over the 50 GHz wavelength division multiplexing grid (Redyuk et al., 2014). Transmission link consisted of spans of typical single-mode fibre (SMF) 100 km long. The basic parameters of the fibres are $\beta_2 = -21$ ps^{2}km^{-1}, $\gamma = 0.0012$ mW^{-1}km^{-1}, $\alpha = 0.2$ dB/km. In order to model 2-polarization signal propagation over the nonlinear fiber a system of coupled nonlinear Schrödinger equations for two complex amplitudes A_X and A_Y of X and Y polarizations has been used. After each EDFA the signal was noise loaded by the white Gaussian noise calculated using a 6 dB amplifier noise figure.

At the receiver side the signal was filtered by a super Gaussian filter with 30 GHz 3 dB bandwidth. The chromatic dispersion was fully compensated by multiplying the Fourier transformed optical field with the reverse dispersion function. For phase estimation, an algorithm based on the 4th-power Viterbi-Viterbi method has been used.

The simulation process was repeated 10 times with different random realizations of ASE noise and input PRBS. The signal was detected at intervals of 1,000 km to a maximum distance 10,000 km. Each pulse was decoded into one of four symbols according to its phase. Each data point has a corresponding two-bit label for each run. Each run generates one data set. I focus on X-Polarization data and use Y-Polarization data for verification of our results. Each pulse is represented by 64 equally spaced phase samples.

Figure A.1 shows the percentage of each symbol on the top panel and the percentage of each bit ($0'$s and $1'$s) on the bottom panel. The data is obtained from the first dataset of the second type. It shows that both symbols and bits are evenly distributed. Furthermore, Figure A.2 shows the similar information for the test set, which is extracted from the first dataset. In general, the test set keeps the same distribution as shown in Figure A.1.

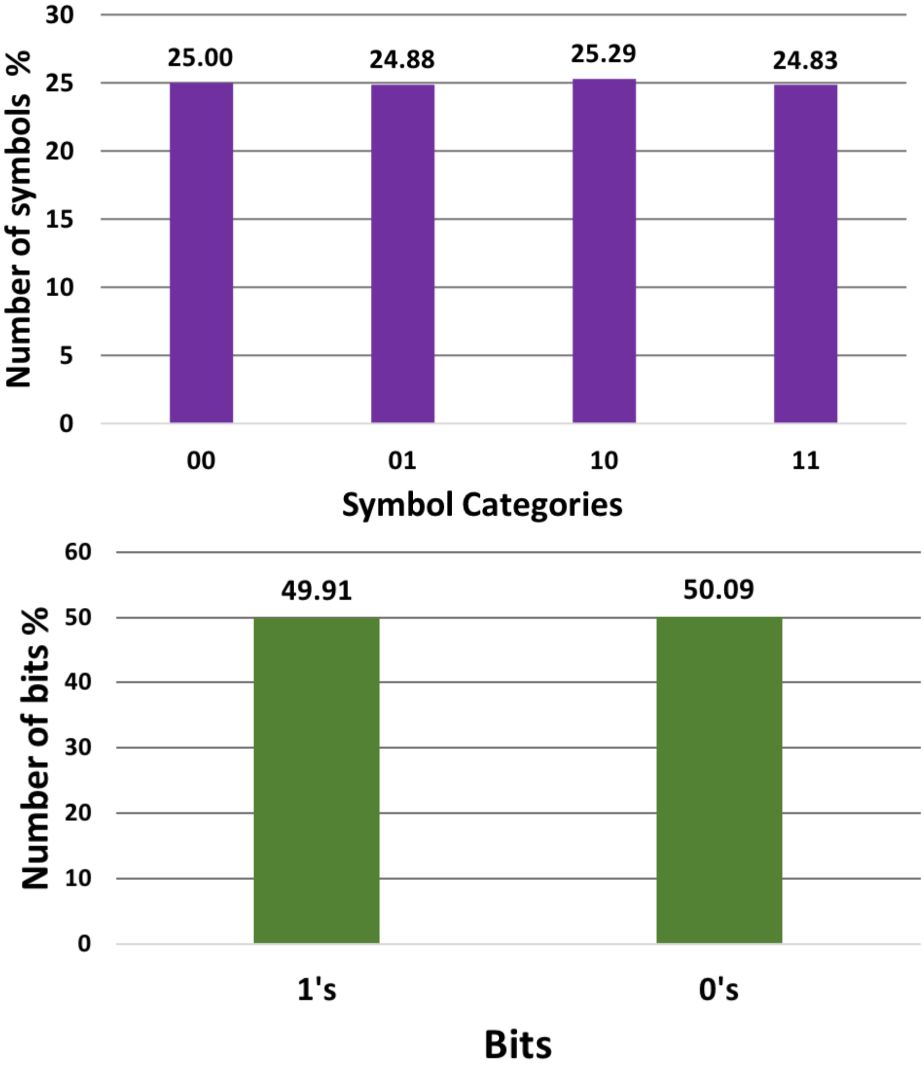


Figure A.1: The top panel shows the percentage of symbols in each class. The percentage of symbols in each class are similar to each other due to the random simulation of the bits. The bottom panel shows the percentage of bits for one or zero, which are similar to each other. The results are obtained from the whole first data-set of the second type of data.

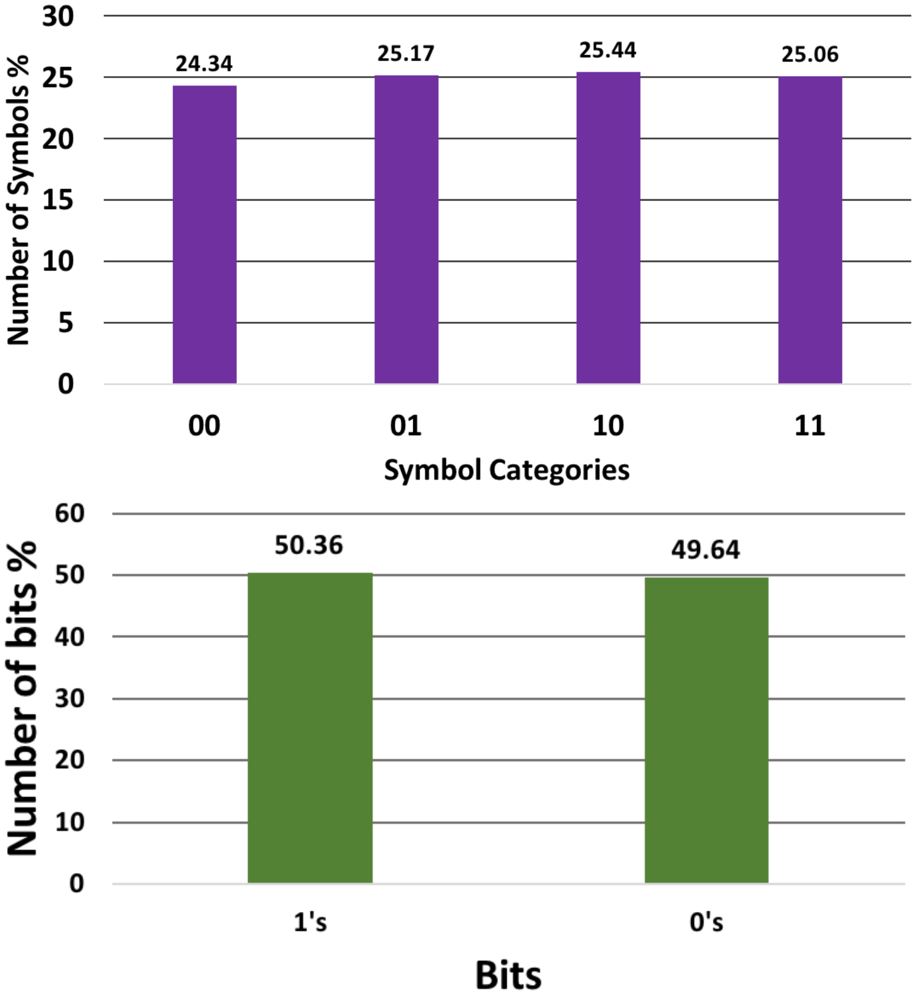


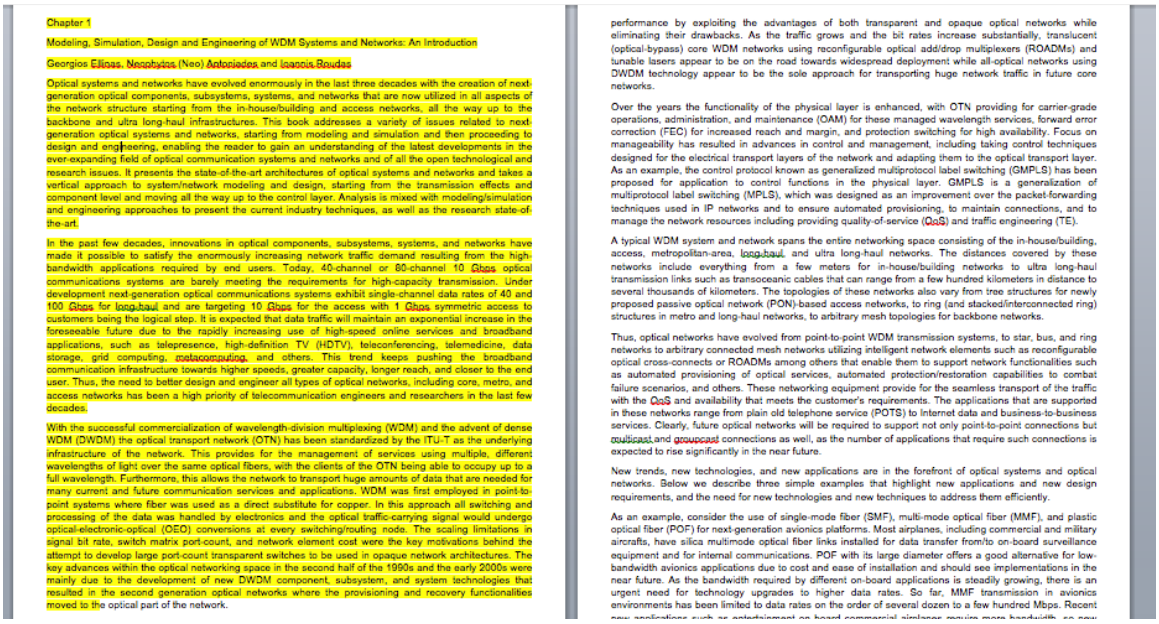
Figure A.2: The top and bottom panels in this figure provide the same information that is showed in Figure A.1, but about only the test set. Also in the test set, it is clear from the top panel that the percentage of all the classes are similar because of the random generation of the bits, which are generated in similar percentage also as it is shown on the bottom panel. The percentages are obtained from the first test set of the second type of data.

A.4 Third Type of Data

This section shows some additional information about the third type of data.

This data has been simulated using the same modulation type and fibre characteristics as used in simulating the second type of data. Figure A.3 presents the modulated text that used in the third type of data. The text is English; and it is a technical text, not a spoken language which is familiar and easy to be predicted.

Table A.1 shows the frequency of each character in the text in the first data set of the third type of data. It can be seen that the frequency of the space (in red colour) and some of the lower case letters (such as letter "e", in blue colour) are much more than the other characters such as numbers or upper case letters. Therefore, I suppose that as an example if the lower case letters appeared



Chapter 1

Modeling, Simulation, Design and Engineering of WDM Systems and Networks: An Introduction

Georgios Ellinas, Neophytos (Neo) Antoniades and Ioannis Roudas

Optical systems and networks have evolved enormously in the last three decades with the creation of next-generation optical components, subsystems, systems, and networks that are now utilized in all aspects of the network structure starting from the in-house/building and access networks, all the way up to the backbone and ultra long-haul infrastructures. This book addresses a variety of issues related to next-generation optical systems and networks, starting from modeling and simulation and then proceeding to design and engineering, enabling the reader to gain an understanding of the latest developments in the ever-expanding field of optical communication systems and networks and of all the open technological and research issues. It presents the state-of-the-art architectures of optical systems and networks and takes a vertical approach to system/network modeling and design, starting from the transmission effects and component level and moving all the way up to the control layer. Analysis is mixed with modeling/simulation and engineering approaches to present the current industry techniques, as well as the research state-of-the-art.

In the past few decades, innovations in optical components, subsystems, systems, and networks have made it possible to satisfy the enormously increasing network traffic demand resulting from the high-bandwidth applications required by end users. Today, 40-channel or 80-channel 10 Gbps optical communications systems are barely meeting the requirements for high-capacity transmission. Under development next-generation optical communications systems exhibit single-channel data rates of 40 and 100 Gbps for long-haul and are targeting 10 Gbps for the access with 1 Gbps symmetric access to customers being the logical step. It is expected that data traffic will maintain an exponential increase in the foreseeable future due to the rapidly increasing use of high-speed online services and broadband applications, such as telepresence, high-definition TV (HDTV), teleconferencing, telemedicine, data storage, grid computing, metacomputing, and others. This trend keeps pushing the broadband communication infrastructure towards higher speeds, greater capacity, longer reach, and closer to the end user. Thus, the need to better design and engineer all types of optical networks, including core, metro, and access networks has been a high priority of telecommunication engineers and researchers in the last few decades.

With the successful commercialization of wavelength-division multiplexing (WDM) and the advent of dense WDM (DWDM) the optical transport network (OTN) has been standardized by the ITU-T as the underlying infrastructure of the network. This provides for the management of services using multiple, different wavelengths of light over the same optical fibers, with the clients of the OTN being able to occupy up to a full wavelength. Furthermore, this allows the network to transport huge amounts of data that are needed for many current and future communication services and applications. WDM was first employed in point-to-point systems where fiber was used as a direct substitute for copper. In this approach all switching and processing of the data was handled by electronics and the optical traffic-carrying signal would undergo optical-electronic-optical (OEO) conversions at every switching/routing node. The scaling limitations in signal bit rate, switch matrix port-count, and network element cost were the key motivations behind the attempt to develop large port-count transparent switches to be used in opaque network architectures. The key advances within the optical networking space in the second half of the 1990s and the early 2000s were mainly due to the development of new DWDM component, subsystem, and system technologies that resulted in the second generation optical networks where the provisioning and recovery functionalities moved to the optical part of the network.

performance by exploiting the advantages of both transparent and opaque optical networks while eliminating their drawbacks. As the traffic grows and the bit rates increase substantially, translucent (optical-bypass) core WDM networks using reconfigurable optical add/drop multiplexers (ROADMs) and tunable lasers appear to be on the road towards widespread deployment while all-optical networks using DWDM technology appear to be the sole approach for transporting huge network traffic in future core networks.

Over the years the functionality of the physical layer is enhanced, with OTN providing for carrier-grade operations, administration, and maintenance (OAM) for these managed wavelength services, forward error correction (FEC) for increased reach and margin, and protection switching for high availability. Focus on manageability has resulted in advances in control and management, including taking control techniques designed for the electrical transport layers of the network and adapting them to the optical transport layer. As an example, the control protocol known as generalized multiprotocol label switching (GMPLS) has been proposed for application to control functions in the physical layer. GMPLS is a generalization of multiprotocol label switching (MPLS), which was designed as an improvement over the packet-forwarding techniques used in IP networks and to ensure automated provisioning, to maintain connections, and to manage the network resources including providing quality-of-service (QoS) and traffic engineering (TE).

A typical WDM system and network spans the entire networking space consisting of the in-house/building, access, metropolitan-area, long-haul, and ultra long-haul networks. The distances covered by these networks include everything from a few meters for in-house/building networks to ultra long-haul transmission links such as transoceanic cables that can range from a few hundred kilometers in distance to several thousands of kilometers. The topologies of these networks also vary from tree structures for newly proposed passive optical network (PON)-based access networks, to ring (and stacked/interconnected ring) structures in metro and long-haul networks, to arbitrary mesh topologies for backbone networks.

Thus, optical networks have evolved from point-to-point WDM transmission systems, to star, bus, and ring networks to arbitrary connected mesh networks utilizing intelligent network elements such as reconfigurable optical cross-connects or ROADMs among others that enable them to support network functionalities such as automated provisioning of optical services, automated protection/restoration capabilities to combat failure scenarios, and others. These networking equipment provide for the seamless transport of the traffic with the QoS and availability that meets the customer's requirements. The applications that are supported in these networks range from plain old telephone service (POTS) to Internet data and business-to-business services. Clearly, future optical networks will be required to support not only point-to-point connections but multicast and groupcast connections as well, as the number of applications that require such connections is expected to rise significantly in the near future.

New trends, new technologies, and new applications are in the forefront of optical systems and optical networks. Below we describe three simple examples that highlight new applications and new design requirements, and the need for new technologies and new techniques to address them efficiently.

As an example, consider the use of single-mode fiber (SMF), multi-mode optical fiber (MMF), and plastic optical fiber (POF) for next-generation avionics platforms. Most airplanes, including commercial and military aircrafts, have silica multimode optical fiber links installed for data transfer from/to on-board surveillance equipment and for internal communications. POF with its large diameter offers a good alternative for low-bandwidth avionics applications due to cost and ease of installation and should see implementations in the near future. As the bandwidth required by different on-board applications is steadily growing, there is an urgent need for technology upgrades to higher data rates. So far, MMF transmission in avionics environments has been limited to data rates on the order of several dozen to a few hundred Mbps. Recent new applications such as entertainment on board commercial airplanes require more bandwidth, so new

Figure A.3: The modulated text that is encoded as the first data set of the third type of data. The text is English; and it is a technical text, not a spoken language which is familiar and easy to be predicted. The following paragraph is a small part of the text ”Optical systems and networks have evolved enormously in the last three decades with the creation of next generation optical components, subsystems, systems and networks that are now utilized in all aspects of the network structure starting from the in-house/building and access networks, all the way up to the backbone and ultra long-haul infrastructures.”

more frequently during the training process, the classifier would be more familiar to them, and might predict them correctly more than the other patterns. For example, in Table A.1, the frequency of the small letter “e” (in red colour) is more than the small letter “z” (in blue colour), and I suppose that letter “e” will appear more frequently during the training process. So, the classifier would be more familiar to it, and might predict it correctly more than letter “z”.

Character	Frequency	Character	Frequency	Character	Frequency
Enter	5	E	3	f	63
New Line	5	F	1	g	81
Space	585	G	5	h	129
(	6	H	1	i	236
)	6	I	7	k	26
,	38	M	7	l	126
–	29	N	5	m	81
.	16	O	5	n	290
/	4	R	1	o	228
0	11	S	2	p	87
1	6	T	13	q	4
2	1	U	2	r	191
4	2	V	2	s	247
8	1	W	7	t	326
9	2	a	252	u	84
:	1	b	36	v	31
A	3	c	135	w	51
C	1	d	130	x	10
D	10	e	409	z	3

Table A.1: The frequency for each character in the modulated text that was used to generate the third type of data. This type of data consists of 4096 characters (58 unique characters). Each character contains 8 bits (4 symbols). It can be seen that the frequency of the space and some of the lower case letters are more than the other characters such as numbers or upper case letters. Therefore, I suppose that as an example if the lower case letters appeared more frequently during the training process, the classifier would be more familiar to them, and might predict them correctly more than the other patterns.

Appendix B

Additional Experiments in the Initial study

B.1 Strategy of Classification using Linear SVM

In these experiments, I have used the data set as both the training set and the test set, to see which software toolbox and the range of parameters can provide a better result initially, and should be adopted in the rest of the study.

Two experiments have been shown in this section:

- **Experiment A:** Using the Toolbox (LIBSVM)for classification which follows the strategy one-against-one (see, Chapter 4), Sub-section B.1.1.
- **Experiment B:** Using the Toolbox (LIBLINEAR) for classification which follows the strategy one-against-all. Both toolboxes was described in the methodology in Chapter 4, Sub-section B.1.2.

In the context of linear classification, one-against-one has shown a better performance compared with one-against-all (Huang, 2010). But it is identified that one-against-one is not suitable for large-scale linear classification because it needs to a large space to save the whole number of classifiers that are generated during the training process (Yuan et al., 2012). However, in the context of sparse weight vectors, one-against-one can deal with large-scale problems for classification. For this reason, I undertook experiments using the toolbox LIBLINEAR to see whether or not it gives better classification results. LIBLINEAR has been considered as an efficient multi-class approach for large scale classification as an open source machine learning library (Fan et al., 2008).

B.1.1 Experiment A

In this experiment, I have used the Toolbox (LIBSVM) for classification which follows the strategy one-against-one (see, Chapter 4). The whole data set is used to train the linear SVM model as a training set, and the same data set is used for testing the model. I have used five different input vectors to the classifier:

1. The complete set of samples of the target symbol (64 × 2 is when the optical transmission symbol is represented as $\sin\theta$ and $\cos\theta$).
2. The central sample of the target symbol (1 × 2 is when the optical transmission symbol is represented as $\sin\theta$ and $\cos\theta$).
3. The three central samples of the target symbol (3 × 2 is when the optical transmission symbol is represented as $\sin\theta$ and $\cos\theta$).
4. The five central samples of the target symbol (5 × 2 is when the optical transmission symbol is represented as $\sin\theta$ and $\cos\theta$).
5. The odd indices of samples of the target symbol (32 × 2 is when the optical transmission symbol is represented as $\sin\theta$ and $\cos\theta$).

Table B.1 shows the results for this section. The best result I have got when using the complete set of samples of each symbol.

Cost Parameter C	**No. of samples of the target symbol**	**No. of features**	**SER %**	**BER %**
1	Central	1 × 2	1.18±0.17	0.59±0.09
512	**Whole**	64 × 2	**0.27±0.07**	**0.14±0.04**
2048	Three central	3 × 2	1.13±0.16	0.57±0.08
2048	Five central	5 × 2	1.11±0.16	0.56±0.08
256	Odd	32 × 2	0.34±0.08	0.17±0.04

Table B.1: Linear SVM results using the toolbox LIBSVM on the first type of data at the distance 3,000 km. All the results are averages over 50 data sets. (×2) denotes each an optical transmission symbol is represented as $\sin\theta$,$\cos\theta$. (×3) denotes each optical transmission symbol is represented as $\sin\theta$, $\cos\theta$ and amplitude. Note that the whole data set is used to train the linear SVM model as a training set, and the same data set is used for testing the model. The text in red denotes the best result.

B.1.2 Experiment B

Linear SVM classification results are obtained using the toolbox (LIBLINEAR (see, Chapter 4)). The whole data set is used to train the linear SVM model as a training set, and the same data set is used for testing the model. I have used two different input vectors:

1. The complete set of samples of the symbols.

2. The central sample of the target symbol.

I have used different cost parameter C values (1, 8, 64, 256 and 2048). Table B.2 shows all the results in this section. Using different cost parameter C values gives similar results. Furthermore, using LIBLINEAR toolbox provides worse results than the best result obtained using LIBSVM toolbox as shown in Table B.2. So, I have decided to use LIBSVM toolbox in the rest of my experiments.

No. of samples of the target symbol	**No. of features**	**SER %**	**BER %**
Central	1×2	1.19±0.17	0.6±0.09
Whole	64×2	**0.69±0.11**	**0.35±0.6**

Table B.2: Linear SVM results using the toolbox LIBLINEAR on the first type of data at the distance 3,000 km. All the results are averages over 50 data sets. The whole data set is used to train the linear SVM model as a training set, and the same data set is used for testing the model. The text in red denotes the best result.

B.2 Experiment C: Increasing the Information that is Used to Train the SVM Classifier

This section investigates whether or not adding the amplitude value to the information that is used to train the SVM classifier can improve classification accuracy. The whole data set is used to train the linear SVM model as a training set, and the same data set is used for testing the model. I have used four different input vectors to the classifier:

1. The complete set of samples of the target symbol (64×3 is when the optical transmission symbol is represented as $\sin\theta$, $\cos\theta$ and amplitude).

2. The central sample of the target symbol (1×3 is when the optical transmission symbol is represented as $\sin\theta$, $\cos\theta$ and amplitude).

3. The three central samples of the target symbol (3×3 is when the optical transmission symbol is represented as $\sin\theta$, $\cos\theta$ and amplitude).

4. The five central samples of the target symbol (5×3 is when the optical transmission symbol is represented as $\sin\theta$, $\cos\theta$ and amplitude).

Table B.3 shows all the results about adding the amplitude to the information used to train the SVM classifier. The best results I have obtained so far is when using the complete set of samples of the target symbol, 192 features as ($\sin\theta$,$\cos\theta$ and *amplitude*), (see the text in red color). The result is a slightly better than using the complete set of samples of the target symbol as (128 as $\sin\theta$,$\cos\theta$), and much better the the result obtained using the threshold method, since the BER is 0.61 %.

Cost Parameter C	**No. of samples from one symbol**	**No. of features**	**SER** %	**BER** %
1028	Central	1×3	1.18±0.17	0.59±0.09
512	**Whole**	64×3	**0.23±0.07**	**0.12±0.04**
2048	Three central	3×3	1.12±0.16	0.56±0.09
2048	Five central	5×3	1.11±0.16	0.56±0.08

Table B.3: Linear SVM results using the toolbox LIBSVM on the first type of data at the distance 3,000 km. All the results are averages over 50 data sets. ($\times 3$) denotes each an optical transmission symbol is represented as ($\sin\theta$,$\cos\theta$ and *amplitude*). The whole data set is used to train the linear SVM model as a training set, and the same data set is used for testing the model. The text in red denotes the best result.

B.3 Experiment D: Extending the gamma (γ) value

I used here just the first and second data sets to implement this experiment. Because of that the searching for the best values of γ and C usually takes long time to be finished. I used two different input vectors to my classifier:

- The central sample (33^{rd}) of the target symbol, 1×2 as ($\sin\theta$, $\cos\theta$).

- The complete set of samples of the target symbol, 64×2 as ($\sin\theta$, $\cos\theta$) and 64×3 as ($\sin\theta$, $\cos\theta$, *amplitude*).

The extending values of the parameters that are used to train the non-linear SVM model on the complete set of samples from one symbol are shown in Chapter 5, Table 5.2. (C) is the cost parameter, and (γ) is Scaling parameter.

The results of extending the γ parameter can be seen in Table B.4. By comparing the obtained results in Table B.4 and the results obtained using the default values of C and γ parameters as shown in Table B.5, extending gamma value would not give us much better result.

B.4 Experiment E: Understanding the SVM's Work

This secton investigates how the SVM works to understand more about the steps implemented during the classification process. I did the binary classification manually using the LIBSVM strategy, and extracted the weight and bias that are usually used to set-up the hyper-plane. This experiment is divided into two folds:

- Doing the binary classification manually.

- Extracting the weight and bais, and drawing the hyper-plan.

Table B.4: Non-linear SVM result using RBF kernel over two data-sets. (×2) denotes each an optical transmission symbol is represented as $\sin\theta$,$\cos\theta$. (×3) denotes each optical transmission symbol is represented as $\sin\theta$, $\cos\theta$ and amplitude. SA and BA denote Symbol and Bit Accuracy, respectively. SE and BE denote the number of Symbol and Bit Errors. This result obtained using the first and second data-set of the first type of data, at the distance of 3,000 km.

No. of samples	No. of features	Data-set No.	Best C values	Best γ values	No. of SE	No. of BE	SA %	BA %
Central	1×2	1	131072	0.000004	93	93	99.14	99.57
		2	512	0.06	85	86	99.21	99.60
Whole	64×2	1	8	0.004	40	40	99.63	99.81
		2	8	0.004	39	39	99.64	99.82
Whole	64×3	**1**	**8**	**0.004**	**33**	**33**	**99.69**	**99.85**
		2	**2**	**0.004**	**37**	**37**	**99.66**	**99.83**

Table B.5: A comparison of the results obtained using linear SVM, non-linear SVM (RBF kernel) and the threshold method. SVM was applied using the complete set of samples of the target symbol, 64×2 as ($\sin\theta$, $\cos\theta$) at the distance of 3,000 km. These results were obtained using the first and second data-sets of the first type of data. MIN SA is the minimum of the symbol accuracy, MAX SA is the maximum of symbol accuracy, Avg SA is the average of symbol accuracy, Avg NSE is the average of number of the symbol errors. MIN BA is the minimum of the bit accuracy, MAX BA is the maximum of bit accuracy, Avg BA is the average of bit accuracy, Avg NBE is the average of number of the bit errors.

Method	MIN SA%	MAX SA%	Avg SA%	Avg NSE	Min BA%	Max BA%	Avg BA%	Avg NBE
Linear SVM	99.12	99.67	99.44±0.14	60.06	99.55	99.84	99.72±0.07	60.50
Non-linear SVM (RBF)	99.11	99.70	99.44±0.15	60.08	99.55	99.85	99.72±0.08	60.58
Threshold	98.12	99.28	98.79±0.26	130.58	99.05	99.64	99.39±0.13	131.16

Doing the binary classification manually:

I faced a problem during doing the previous experiments. It is that the accuracy of prediction is getting worse after a specific increasing in the cost parameter value C. Logically, when the C value is increased, the accuracy of prediction should be better than before. For this reason, this experiment has been done to figure out why this problem happened.

In this experiment, linear SVM classification was repeated on only the first data set with setting the same parameters; excepting the C value which was equal to 512. This specific C value is used because it gives the best SVM models in the previous experiments. Each optical transmission symbol is represented as $\sin\theta$, $\cos\theta$ (128 features).

The classification process is implemented on two classes, not multi-class. In other words, since I have four classes; each two classes are considered as one class. For instance, the first and second classes are considered as the first class; and the third and fourth classes are considered as the second class. Then, the binary classification is employed on these two classes using LIBSVM toolbox. After that each combination of two classes is divided into the original classes to repeat the same experiment again. After that I calculate the accuracy of prediction at each stage. The order of the classes is being changed in each time as it is shown in Figure B.1.

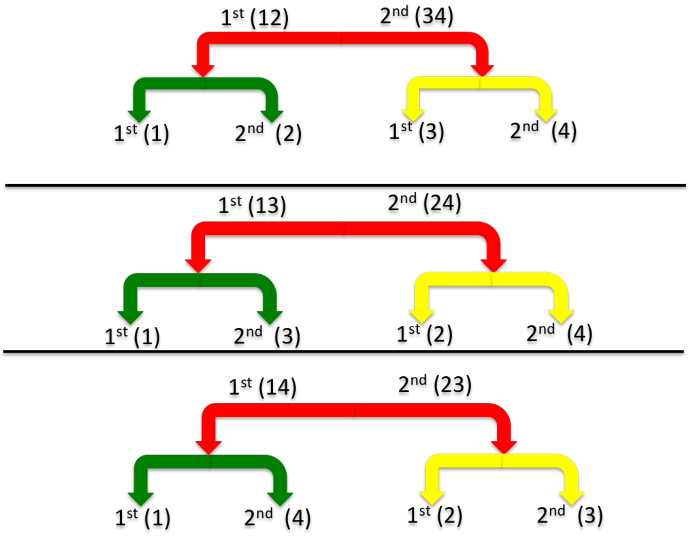


Figure B.1: The order of the classes for each stage during doing the SVM classification manually.

(12_34) _ C = 512

Number of symbols errors (NSE)		Average of accuracy rate (AAR)	
60.44		99.82±0.05	
(12) _ C = 512		**(34) _ C = 512**	
NSE_(12)	**AAR_(12)**	**NSE_(34)**	**AAR_(34)**
87.52	99.73±0.08	87.5	99.73±0.08

(13_24) _ C = 512

Number of symbols errors (NSE)		Average of accuracy rate (AAR)	
61.68		99.81±0.04	
(13) _ C = 512		**(24) _ C = 512**	
NSE_(13)	**AAR_(13)**	**NSE_(24)**	**AAR_(24)**
87.26	99.73±0.76	87.5	99.73±0.08

(14_23) _ C = 512

Number of symbols errors (NSE)		Average of accuracy rate (AAR)	
Experiment is terminated		Experiment is terminated	
(14) _ C = 512		**(23) _ C = 512**	
NSE_(14)	**AAR_(14)**	**NSE_(23)**	**AAR_(23)**
87.16	99.73±0.08	87.34	99.73±0.08

Table B.6: The result of doing the binary classification manually. There is no way to draw a line that can separate the two classes (14) and (23), see Figure B.2.

Table B.6 shows the results of doing the binary classification manually. As we can see the experiment has not been completed in the last stage, and I had to terminate the classification process. The reason is that using linear SVM to classify these combination of classes (14) and (23) is impossible, see Figure B.2, because there is no way to draw the line that can separate these two classes.

Extracting the weight and bais, and drawing the hyper-plan:

This experiment investigates the difference between the hyper-plane that SVM produces, and the boundaries that can be drawn using the threshold method. Herein, the first and second class are separated as the first class, and the third and fourth class as the second class; and that is implemented on the first data set of the first type of data.

To draw the Hyper-plane manually as SVM, the weight and the bias should be calculated. These two values can be extracted from the information inside the model file, which resulted from the training process. Because of the binary class is considered here, the weight can be given by:

$$\mathbf{w} = SVs \times Coef \tag{B.1}$$

where SVs is the number of support vectors, and $Coef$ is the support vector coefficient. Also, the bias can be calculated by:

$$b = -rho \tag{B.2}$$

where rho is the bias term in the decision function.

All of SVs, $Coef$ and rho can be found in the model file. Equation B.1 gives two values for w that are $w_1 = -5.18$, and $w_2 = -4.99$. By these two points $(-5.18\,,\,-4.99)$, and y-intercept $(0\,,\,b)$, which is $(0\,,\,0.1)$, the vector line is drawn. Then, drawing the hyper-plane that is usually a perpendicular line on the vector line and passes through the y-intercept point as well.

The result reveals that the hyper-plane that is drawn by SVM is parallel to the hyper-plane that resulted from the threshold method. But the difference is only in the intercept value where is 0.1, whereas the hyper-plane that is drawn by the threshold method passed through the center $(0, 0)$, so that its intercept value is 0.

Figure B.2 shows the two hyper-planes, where the black line is the hyper-plane that resulted from the SVM method, while the red lines are the hyper-planes resulted from the threshold method. Moreover, the hyper-plane resulted from SVM separated the two classes (12) and (34) better than the one that is drawn by the threshold method.

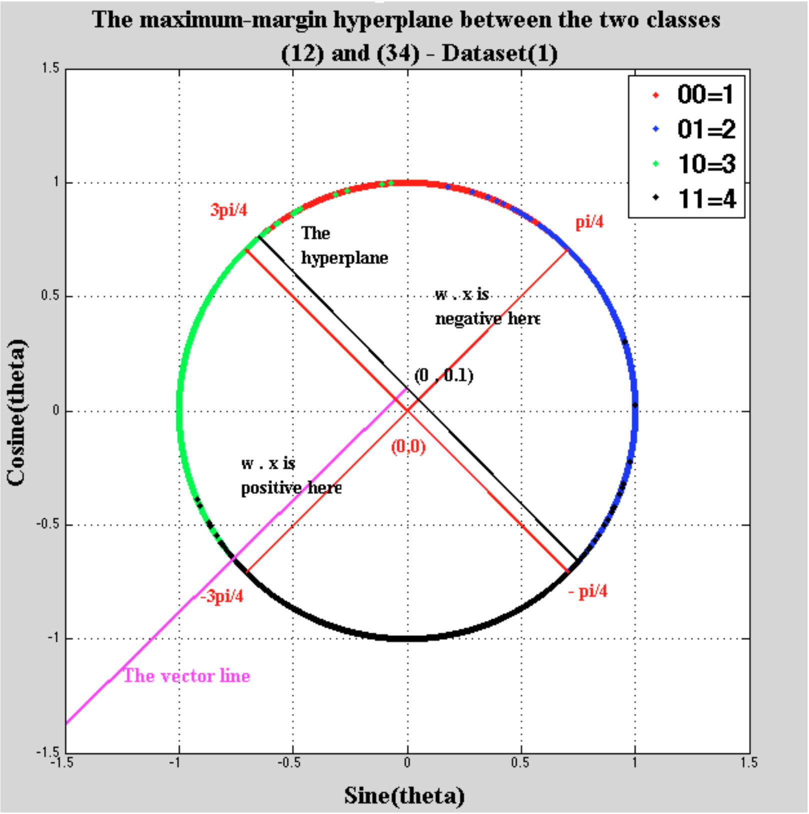


Figure B.2: Drawing the Hyper-plane to separate two classes (12) and (34) on the first data set of the first type of data. There is no way to draw a line that can separate the two classes (14) and (23).

Appendix C

Additional Results for Chapters 7, 8 and 9

This appendix presents all the additional results (tables).

C.1 Results for the Second Set of Modulated Data (Chapter 6)

C.1.1 Using Different Samples from One Symbol (Section 6.2)

Table C.1: Using Different Samples from One Symbol

(a) Distance 2,000 km

Method	**No. of samples**	**No. of features**	**SER** ($\times 10^{-4}$)	**BER** ($\times 10^{-4}$)	**P-Value**
Threshold	Central	-	2.14±1.75	1.07±0.88	
SVM	Whole	64 × 2	2.97±1.22	1.49±0.6	0.12
	Central	1 × 2	2.79±1.64	1.39±0.82	0.07
	3 Mid	3 × 2	2.69±1.61	1.35±0.8	0.17
	5 Mid	5 × 2	2.6±1.5	1.30±0.75	0.21
	32 Mid	32 × 2	2.88±1.19	1.44±0.6	0.18

(b) Distance 3,000 km

Method	**No. of samples**	**No. of features**	**SER** ($\times 10^{-4}$)	**BER** ($\times 10^{-4}$)	**P-Value**
Threshold	Central	-	15.51±2.87	7.75±1.44	
SVM	Whole	64 × 2	15.42±3.87	7.71±1.94	0.91
	Central	1 × 2	15.14±3.61	7.57±1.81	0.42
	3 Mid	3 × 2	15.42±3.4	7.71±1.7	0.85
	5 Mid	5 × 2	15.32±2.98	7.66±1.49	0.64
	32 Mid	32 × 2	15.51±3.36	7.75±1.68	1

(c) Distance 4,000 km

Method	**No. of samples**	**No. of features**	**SER** ($\times 10^{-4}$)	**BER** ($\times 10^{-4}$)	**P-Value**
Threshold	Central	-	47.27±5.93	23.63±2.96	
SVM	Whole	64 × 2	43.55 ± 6.14	21.82±3.1	0.004
	Central	1 × 2	47.08±6	23.54±3	0.66
	3 Mid	3 × 2	46.43±6.73	23.22±3.36	0.23
	5 Mid	5 × 2	46.71±6.3	23.36±3.15	0.42
	32 Mid	32 × 2	45.32±6.45	22.71±3.21	0.003

(d) Distance 5,000 km

Method	**No. of samples**	**No. of features**	**SER** ($\times 10^{-4}$)	**BER** ($\times 10^{-4}$)	**P-Value**
Threshold	Central	-	95.93±8.23	48.15±4.19	
SVM	Whole	64 × 2	91.75±10.85	46.06±5.47	0.07
	Central	1 × 2	95.56±8.03	47.97±4.06	0.58
	3 Mid	3 × 2	94.82±9.73	47.64±4.97	0.39
	5 Mid	5 × 2	95.84±10.28	48.06±5.2	0.88
	32 Mid	32 × 2	94.54±9.62	47.46±4.84	0.48

Table C.1: Using Different Samples from One Symbol (continued)

(e) Distance 6,000 km

Method	No. of samples	No. of features	SER $(\times 10^{-4})$	BER $(\times 10^{-4})$	P-Value
Threshold	Central	-	170.97±10.21	86.13±4.96	
SVM	Whole	64 × 2	157.69±10.57	79.49±5.22	0.00001
	Central	1 × 2	170.41±10.81	85.86±5.24	0.28
	3 Mid	3 × 2	164.93±13.45	83.07±6.69	0.02
	5 Mid	5 × 2	163.73±12.41	82.56±6.17	0.01
	32 Mid	32 × 2	164.65±11.55	83.02±5.69	0.0001

(f) Distance 7,000 km

Method	No. of samples	No. of features	SER $(\times 10^{-4})$	BER $(\times 10^{-4})$	P-Value
Threshold	Central	-	261.61±7.13	132.06±3.91	
SVM	Whole	64 × 2	250.74±8.38	126.67±4.22	0.003
	central	1 × 2	261.89±7.89	132.2±4.33	0.71
	3 Mid	3 × 2	254.83±6.01	128.76±3.17	0.0004
	5 Mid	5 × 2	254.18±6.65	128.53±3.49	0.003
	32 Mid	32 × 2	257.71±7.01	130.2±3.6	0.15

(g) Distance 8,000 km

Method	No. of samples	No. of features	SER $(\times 10^{-4})$	BER $(\times 10^{-4})$	P-Value
Threshold	Central	-	370.17±14.76	187.27±7.69	
SVM	Whole	64 × 2	356.24±13.74	179.84±7.36	0.001
	central	1 × 2	370.54±16.23	187.41±8.36	0.85
	3 Mid	3 × 2	359.03±15.05	181.65±7.98	0.003
	5 Mid	5 × 2	359.68±16.22	181.93±8.59	0.01
	32 Mid	32 × 2	359.12±16.96	181.28±8.94	0.003

(h) Distance 9,000 km

Method	No. of samples	No. of features	SER $(\times 10^{-4})$	BER $(\times 10^{-4})$	P-Value
Threshold	Central	-	488.48±16.13	247.72±8.67	
SVM	Whole	64 × 2	474.46±16.94	240.53±8.68	0.007
	Central	1 × 2	489.13±14.45	248.05±7.87	0.49
	3 Mid	3 × 2	478.36±17.01	242.48±9.33	0.001
	5 Mid	5 × 2	478.08±18.15	242.38±9.77	0.001
	32 Mid	32 × 2	480.03±21.04	243.31±11.1	0.02

Table C.1: Using Different Samples from One Symbol (continued)

(i) Distance 10,000 km

Method	**No. of samples**	**No. of features**	**SER** ($\times10^{-4}$)	**BER** ($\times10^{-4}$)	**P-Value**
Threshold	Central	-	607.08±16.68	309.53±9.98	
SVM	Whole	64 × 2	588.5±23.71	300.06±13.05	0.01
	Central	1 × 2	610.33±19.01	311.2±10.95	0.06
	3 Mid	3 × 2	591.75±18.22	301.91±10.55	0.02
	5 Mid	5 × 2	589.34±17.75	300.66±10.02	0.004
	32 Mid	32 × 2	593.61±20.57	302.66±11.39	0.02

Table C.1: The linear SVM results using optical signals (using different samples from one symbol) at distances from 2,000 km to the maximum 10,000 km, compared with the threshold method result. SER denotes the average of symbol error ratio. BER denotes the average of bit error ratio. ($\times2$) denotes that each an optical signal (X-POL) input is represented by ($\sin\theta$,$\cos\theta$). The text in blue refers to the results obtained using the threshold method. The text in red refers to the best SVM result.

C.1.2 Using Samples from Neighboring Symbols (Section 6.3)

Table C.2: Using Samples from Neighboring Symbols: the letters P, T and S denote that preceding, target and succeeding symbol respectively

(a) Distance 2,000 km

Method	**No. of symbols and samples**	**No. of features**	**SER** $(\times 10^{-4})$	**BER** $(\times 10^{-4})$	**P-Value**
Threshold	1(Central/T)	-	2.14±1.75	1.07±0.88	
SVM	1(Whole/T)	64 × 2	2.97±1.22	1.49±0.6	0.12
SVM	3(Central/P,T,S)	3 × 2	1.86±1.38	0.93±0.69	0.04
	3($\frac{1}{2}P$,Whole/T,$\frac{1}{2}S$)	128 × 2	2.04±0.96	1.02±0.48	0.8
	3(Whole/P,T,S)	192 × 2	2.23±1.25	1.11±0.63	0.78
	2(Central/P,T)	2 × 2	1.58±1.39	0.79±0.69	0.02
	2(Central/T,S)	2 × 2	2.14±0.98	1.07±0.49	1
	3(Central/2P,T)	3 × 2	1.67±1.44	0.84±0.72	0.1
	2(Whole/P,T)	128 × 2	2.88±1.19	1.44±0.6	0.15

(b) Distance 3,000 km

Method	**No. of symbols and samples**	**No. of features**	**SER** $(\times 10^{-4})$	**BER** $(\times 10^{-4})$	**P-Value**
Threshold	1(Central/T)	-	15.51±2.87	7.75±1.44	
SVM	1(Whole/T)	64 × 2	15.42±3.87	7.71±1.94	0.91
SVM	3(Central/P,T,S)	3 × 2	9.85±1.87	4.92±0.93	0.00001
	3($\frac{1}{2}P$,Whole/T,$\frac{1}{2}S$)	128 × 2	13.65±3.64	6.83±1.82	0.11
	3(Whole/P,T,S)	192 × 2	12.54±3.31	6.27±1.66	0.01
	2(Central/P,T)	2 × 2	12.72±2.7	6.36±1.35	0.01
	2(Central/T,S)	2 × 2	12.91±2.95	6.45±1.48	0.001
	3(Central/2P,T)	3 × 2	11.98±2.82	5.99±1.41	0.0002
	2(Whole/P,T)	128 × 2	15.04±2.58	7.52±1.29	0.67

(c) Distance 4,000 km

Method	**No. of symbols and samples**	**No. of features**	**SER** $(\times 10^{-4})$	**BER** $(\times 10^{-4})$	**P-Value**
Threshold	1(Central/T)	-	47.27±5.93	23.63±2.96	
SVM	1(Whole/T)	64 × 2	43.55 ± 6.14	21.82±3.1	0.004
SVM	3(Central/P,T,S)	3 × 2	34.92±4.07	17.46±2.03	0.0001
	2($\frac{1}{2}P$,Whole/T,$\frac{1}{2}S$)	128 × 2	40.59±5.54	20.29±2.77	0.001
	3(Whole/P,T,S)	192 × 2	41.05±6.65	20.53±3.33	0.01
	2(Central/P,T)	2 × 2	38.82±5.14	19.41±2.57	0.0001
	2(Central/T,S)	2 × 2	43.83±5.25	21.92±2.62	0.02
	3(Central/2P,T)	3 × 2	37.43±4.15	18.71±2.08	0.00003
	2(Whole/P,T)	128 × 2	41.51±5.52	20.76±2.76	0.001

Table C.2: Using Samples from Neighboring Symbols: the letters P, T and S denote that preceding, target and succeeding symbol respectively (continued)

(d) Distance 5,000 km

Method	**No. of symbols and samples**	**No. of features**	**SER** ($\times 10^{-4}$)	**BER** ($\times 10^{-4}$)	**P-Value**
Threshold	1(Central/T)	-	95.93±8.23	48.15±4.19	
SVM	1(Whole/T)	64 × 2	91.75±10.85	46.06±5.47	0.07
SVM	3(Central/P,T,S)	3 × 2	83.13±8.83	41.75±4.41	0.00003
	3($\frac{1}{2}P$,Whole/T,$\frac{1}{2}S$)	128 × 2	87.12±9.36	43.7±4.65	0.001
	3(Whole/P,T,S)	192 × 2	85.45±10.58	42.91±5.34	0.002
	2(Central/P,T)	2 × 2	87.48±11.17	43.93±5.68	0.0004
	2(Central/T,S)	2 × 2	94.35±8.16	47.36±4.04	0.21
	3(Central/2P,T)	3 × 2	85.25±11.3	42.81±5.69	0.0002
	2(Whole/P,T)	128 × 2	87.76±7.21	44.07±3.68	0.0001

(e) Distance 6,000 km

Method	**No. of symbols and samples**	**No. of features**	**SER** ($\times 10^{-4}$)	**BER** ($\times 10^{-4}$)	**P-Value**
Threshold	1(Central/T)	-	170.97±10.21	86.13±4.96	
SVM	Whole	64 × 2	157.69±10.57	79.49±5.22	0.00001
SVM	3(Central/P,T,S)	3 × 2	152.24±12.34	76.77±6.07	0.00004
	3($\frac{1}{2}P$,Whole/T,$\frac{1}{2}S$)	128 × 2	150.65±9.78	75.97±4.83	0.000001
	3(Whole/P,T,S)	192 × 2	151.11±12.73	76.25±6.3	0.00001
	2(Central/P,T)	2 × 2	153.79±11.28	77.54±5.49	0.000004
	2(Central/T,S)	2 × 2	163.63±8.45	82.47±4	0.002
	3(Central/2P,T)	3 × 2	152.12±9.71	76.71±4.69	0.000002
	2(Whole/P,T)	128 × 2	156.39±9.86	78.89±5	0.00001

(f) Distance 7,000 km

Method	**No. of symbols and samples**	**No. of features**	**SER** ($\times 10^{-4}$)	**BER** ($\times 10^{-4}$)	**P-Value**
Threshold	1(Central/T)	-	261.61±7.13	132.06±3.91	
SVM	1(Whole/T)	64 × 2	250.74±8.38	126.67±4.22	0.003
SVM	3(Central/P,T,S)	3 × 2	239.55±7.89	120.98±3.99	0.000002
	3($\frac{1}{2}P$,Whole/T,$\frac{1}{2}S$)	128 × 2	242.04±10.93	122.36±5.63	0.0002
	3(Whole/P,T,S)	192 × 2	242.22±6.17	122.32±3.48	1.86E-05
	2(Central/P,T)	2 × 2	243.5±8.94	123±4.41	0.00003
	2(Central/T,S)	2 × 2	258.36±8.09	130.43±4.26	0.04
	3(Central/2P,T)	3 × 2	241.08±7.37	121.8±3.74	0.000003
	2(Whole/P,T)	128 × 2	250.19±8.47	126.39±4.34	0.001

Table C.2: Using Samples from Neighboring Symbols (continued)

(g) Distance 8,000 km

Method	**No. of symbols and samples**	**No. of features**	**SER** ($\times 10^{-4}$)	**BER** ($\times 10^{-4}$)	**P-Value**
Threshold	1(Central/T)	-	370.17±14.76	187.27±7.69	
SVM	1(Whole/T)	64×2	356.24±13.74	179.84±7.36	0.001
SVM	3(Central/P,T,S)	3×2	342±13.65	173±7.43	0.0001
	3($\frac{1}{2}P$,Whole/T,$\frac{1}{2}S$)	128×2	343.46±8.8	173.49±4.79	0.00003
	3(Whole/P,T,S)	192×2	344.01±8.57	173.82±4.87	2.26E-05
	2(Central/P,T)	2×2	351.32±15.27	177.7±8.09	0.0002
	2(Central/T,S)	2×2	368.78±10.61	186.62±5.71	0.57
	3(Central/2P,T)	3×2	347.14±13.17	175.61±7.2	0.0001
	2(Whole/P,T)	128×2	349.18±12.16	176.45±6.57	0.00003

(h) Distance 9,000 km

Method	**No. of symbols and samples**	**No. of features**	**SER** ($\times 10^{-4}$)	**BER** ($\times 10^{-4}$)	**P-Value**
Threshold	1(Central/T)	-	488.48±16.13	247.72±8.67	
SVM	1(Whole/T)	64×2	474.46±16.94	240.53±8.68	0.01
SVM	3(Central/P,T,S)	3×2	468.6±19.25	237.74±10.15	0.000005
	3($\frac{1}{2}P$,Whole/T,$\frac{1}{2}S$)	128×2	461.6±15.92	234.19±7.88	0.0004
	3(Whole/P,T,S)	192×2	459±19.52	233.03±10.17	0.0001
	2(Central/P,T)	2×2	468.33±18.88	237.65±9.85	0.00002
	2(Central/T,S)	2×2	483.75±16.48	245.31±8.81	0.07
	3(Central/2P,T)	3×2	466.66±18.35	236.77±9.72	0.00003
	2(Whole/P,T)	128×2	466.01±17.2	236.3±8.92	0.0001

(i) Distance 10,000 km

Method	**No. of symbols and samples**	**No. of features**	**SER** ($\times 10^{-4}$)	**BER** ($\times 10^{-4}$)	**P-Value**
Threshold	1(Central/T)	-	607.08±16.68	309.53±9.98	
SVM	1(Whole/T)	64×2	588.5±23.71	300.06±13.05	0.01
SVM	3(Central/P,T,S)	3×2	583.32±19.93	297.14±11.51	0.0004
	3($\frac{1}{2}P$,Whole/T,$\frac{1}{2}S$)	128×2	576.39±17.95	294.00±10.15	0.0001
	3(Whole/P,T,S)	192×2	576.58±21.04	293.95±11.50	0.0004
	2(Central/P,T)	2×2	588.97±23.48	300.33±13.41	0.005
	2(Central/T,S)	2×2	606.8±20.44	309.39±11.73	0.88
	3(Central/2P,T)	3×2	577.64±18.38	294.39±10.91	0.00003
	2(Whole/P,T)	128×2	584.6±28.43	297.97±15.4	0.004

Table C.2: The linear SVM results using neighboring information on the optical waves/symbols at distances from 2,000 km to the maximum 10,000 km, compared with the threshold method result. SER denotes the average of Symbol Error Ratio. BER denotes the average of Bit Error Ratio. ($\times 2$) denotes that each an optical signal input is represented by ($\sin\theta$,$\cos\theta$). The letters P, T and S denote that preceding, target and succeeding symbol respectively. The text in blue refers to the results obtained using the threshold method. The text in red refers to the best SVM result.

C.1.3 Using Central Samples from Neighboring Information (Section 6.4)

Table C.3: Using Central Samples from Neighboring Information, the letters P, T and S denote that preceding, target and succeeding symbol respectively.

(a) Distance 2,000 km

Method	No. of symbols and samples	No. of features	SER $(\times 10^{-4})$	BER $(\times 10^{-4})$	P-Value
Threshold	1(Central/T)	-	2.14±1.75	1.07±0.88	
SVM	1(Whole/T)	64 × 2	2.97±1.22	1.49±0.6	0.12
	3(Central/P,T,S)	3 × 2	1.86±1.38	0.93±0.69	0.04
SVM	5(Central/2P,T,2S)	5 × 2	1.58±1.52	0.79±0.76	0.58
	7(Central/3P,T,3S)	7 × 2	1.3±1.09	0.65±0.55	0.33

(b) Distance 3,000 km

Method	No. of symbols and samples	No. of features	SER $(\times 10^{-4})$	BER $(\times 10^{-4})$	P-Value
Threshold	1(Central/T)	-	15.51±2.87	7.75±1.44	
SVM	1(Whole/T)	64 × 2	15.42±3.87	7.71±1.94	0.91
	3(Central/P,T,S)	3 × 2	9.85±1.87	4.92±0.93	0.00001
SVM	5(Central/2P,T,2S)	5 × 2	10.4±2.96	5.2±1.48	0.000002
	7(Central/3P,T,3S)	7 × 2	10.87±2.81	5.43±1.4	0.0002

(c) Distance 4,000 km

Method	No. of symbols and samples	No. of features	SER $(\times 10^{-4})$	BER $(\times 10^{-4})$	P-Value
Threshold	1(Central/T)	-	47.27±5.93	23.63±2.96	
SVM	1(Whole/T)	64 × 2	43.55 ± 6.14	21.82±3.1	0.004
	3(Central/P,T,S)	3 × 2	34.92±4.07	17.46±2.03	0.0001
SVM	5(Central/2P,T,2S)	5 × 2	36.41±4.4	18.21±2.2	0.0002
	7(Central/3P,T,3S)	7 × 2	36.04±3.81	18.02±1.91	0.0001

(d) Distance 5,000 km

Method	No. of symbols and samples	No. of features	SER $(\times 10^{-4})$	BER $(\times 10^{-4})$	P-Value
Threshold	1(Central/T)	-	95.93±8.23	48.15±4.19	
SVM	1(Whole/T)	64 × 2	91.75±10.85	46.06±5.47	0.07
	3(Central/P,T,S)	3 × 2	83.13±8.83	41.75±4.41	0.00003
SVM	5(Central/2P,T,2S)	5 × 2	82.2±9.09	41.29±4.59	0.00002
	7(Central/3P,T,3S)	7 × 2	82.58±8.85	41.48±4.4	0.0001

Table C.3: Using Central Samples from Neighboring Information, the letters P, T and S denote that preceding, target and succeeding symbol respectively. (continued)

(e) Distance 6,000 km

Method	No. of symbols	No. of	SER (×10^{-4})	BER	P-Value
	and samples	features		(×10^{-4})	
Threshold	1(Central/T)	-	170.97±10.21	86.13±4.96	
SVM	1(Whole/T)	64 × 2	157.69±10.57	79.49±5.22	0.00001
	3(Central/P,T,S)	3 × 2	152.24±12.34	76.77±6.07	0.00004
SVM	5(Central/2P,T,2S)	5 × 2	150.29±10.45	75.84±4.87	0.00003
	7(Central/3P,T,3S)	7 × 2	148.07±9.87	74.69±4.72	0.0000004

(f) Distance 7,000 km

Method	No. of symbols	No. of	SER	BER	P-Value
	and samples	features	(×10^{-4})	(×10^{-4})	
Threshold	1(Central/T)	-	261.61±7.13	132.06±3.91	
SVM	1(Whole/T)	64 × 2	250.74±8.38	126.67±4.22	0.003
	3(Central/P,T,S)	3 × 2	239.55±7.89	120.98±3.99	0.000002
SVM	5(Central/2P,T,2S)	5 × 2	240.94±7.91	121.73±4.07	0.0000005
	7(Central/3P,T,3S)	7 × 2	237.34±9.8	119.88±5.01	0.000001

(g) Distance 8,000 km

Method	No. of symbols	No. of	SER	BER	P-Value
	and samples	features	(×10^{-4})	(×10^{-4})	
Threshold	1(Central/T)	-	370.17±14.76	187.27±7.69	
SVM	1(Whole/T)	64 × 2	356.24±13.74	179.84±7.36	0.001
	3(Central/P,T,S)	3 × 2	342±13.65	173±7.43	0.0001
SVM	5(Central/2P,T,2S)	5 × 2	347.02±17.05	175.51±8.8	0.002
	7(Central/3P,T,3S)	7 × 2	339.9±9.8	171.95±5.41	0.000004

(h) Distance 9,000 km

Method	No. of symbols	No. of	SER	BER	P-Value
	and samples	features	(×10^{-4})	(×10^{-4})	
Threshold	1(Central/T)	-	488.48±16.13	247.72±8.67	
SVM	1(Whole/T)	64 × 2	474.46±16.94	240.53±8.68	0.01
	3(Central/P,T,S)	3 × 2	468.6±19.25	237.74±10.15	0.000005
SVM	5(Central/2P,T,2S)	5 × 2	466.28±19.06	236.53±10.23	0.002
	7(Central/3P,T,3S)	7 × 2	456.11±15.83	231.35±8.64	0.0000001

Table C.3: Using Central Samples from Neighboring Information

(i) Distance 10,000 km

Method	No. of symbols and samples	No. of features	SER (×10^{-4})	BER (×10^{-4})	P-Value
Threshold	1(Central/T)	-	607.08±16.68	309.53±9.98	
SVM	1(Whole/T)	64×2	588.5±23.71	300.06±13.05	0.01
	3(Central/P,T,S)	3×2	583.32±19.93	297.14±11.51	0.0004
SVM	5(Central/2P,T,2S)	5×2	576.07±17.1	293.7±10.08	0.00001
	7(Central/3P,T,3S)	7×2	574.27±15.81	292.85±9.561	0.000004

Table C.3: The linear SVM results using different number of central samples from different number of adjacent symbols on the optical signals at the distances from 2,000 km to the maximum 10,000 km, compared with the threshold method result. SER denotes the average of symbol error ratio. BER denotes the average of bit error ratio. (×2) denotes that each an optical signal input is represented by ($\sin\theta$,$\cos\theta$). The letters P, T and S denote that preceding, target and succeeding symbol respectively. The text in blue refers to the results obtained using the threshold method. The text in red refers to the best SVM result.

C.1.4 Some Results about the Second Type of Data (Y-Polarization)

The experiments include:

1. The threshold method. See Table C.4.
2. Linear SVM and linear Search to find the best cost parameter C value:
   - Investigate different number of samples of the target symbol:
     - Experiment 1: Using the complete set of samples of the target symbol, See Table C.5, Figure C.1.
     - Experiment 2: Using the 33^{rd} central sample of the target symbol, See Table C.6, Figure C.2.
   - Investigate the effect of the symbols either side:
     - Experiment 1: Using the 33^{rd} central sample of the target symbol and symbol either side, See Table C.7, Figure C.3.
     - Experiment 2: Using the 33^{rd} central sample from the target symbol and two symbols either side, See Table C.8, Figure C.4.
     - Experiment 3: Using the 33^{rd} central sample of the target symbol and three symbols either side, See Table C.9, Figure C.5.
     - Experiment 4: Using the 33^{rd} central sample of the target symbol and four symbols either side, See Table C.10, Figure C.6.

- Experiment 5: Using the 33^{rd} central sample of the target symbol and two preceding symbols, See Table C.11, Figure C.7.
- Experiment 6: Using the complete set of samples of the target and the preceding symbol, See Table C.12, Figure C.8.

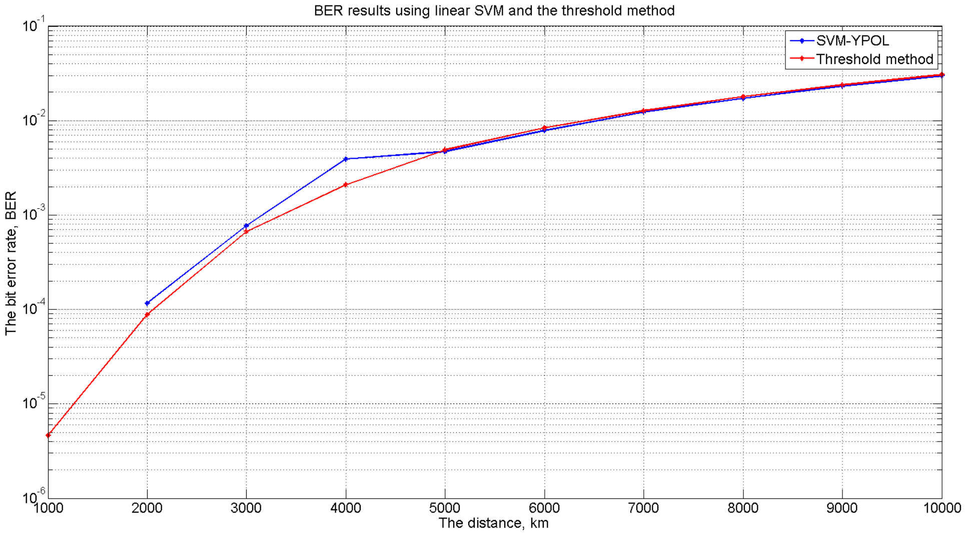


Figure C.1: A comparison between the BER (log-scaled) obtained using linear SVM and the threshold method. The input vector to our classifier is the complete set of samples of the target symbol, at the distances from 1,000 km to 10,000 km, Y-Polarization.

Table C.4: The threshold method results using the 33^{rd}central sample at the distances from 1,000 km to 10,000 km, Y-Polarization. MIN SA is the minimum of the symbol accuracy, MAX SA is the maximum of symbol accuracy, Avg SA is the average of symbol accuracy, Avg SER is the average of symbol error ratio, Avg NSE is the average of number of the symbol errors. MIN BA is the minimum of the bit accuracy, MAX BA is the maximum of bit accuracy, Avg BA is the average of bit accuracy, Avg BER is the average of bit error ratio, Avg NBE is the average of number of the bit errors.

Distance	MIN SA%	MAX SA%	Avg SA%	Avg SER	Avg NSE	Min BA%	Max BA%	Avg BA%	Avg BER	Avg NBE
1,000 km	99.99	100.00	100 ± 0	0 ± 0	0.1	100.00	100.00	100 ± 0	0.000005 ± 0.00001	0.1
2,000 km	99.96	99.99	99.98 ± 0.01	0.02 ± 0.01	1.9	99.98	100.00	99.99 ± 0.01	0.01 ± 0.01	1.9
3,000 km	99.81	99.93	99.87 ± 0.04	0.13 ± 0.04	14.3	99.91	99.97	99.93 ± 0.02	0.07 ± 0.02	14.4
4,000 km	99.50	99.64	99.58 ± 0.04	0.42 ± 0.04	45.1	99.75	99.82	99.79 ± 0.02	0.21 ± 0.02	45.2
5,000 km	98.88	99.15	99.02 ± 0.08	0.98 ± 0.08	105.2	99.44	99.57	99.51 ± 0.04	0.49 ± 0.04	105.5
6,000 km	98.14	98.54	98.34 ± 0.13	1.66 ± 0.13	178.8	99.07	99.27	99.16 ± 0.06	0.84 ± 0.06	180
7,000 km	97.17	97.68	97.49 ± 0.15	2.51 ± 0.15	270.4	98.57	98.83	98.73 ± 0.08	1.27 ± 0.08	274.1
8,000 km	96.04	96.81	96.46 ± 0.27	3.54 ± 0.27	380.8	97.99	98.39	98.21 ± 0.14	1.79 ± 0.14	385.7
9,000 km	94.99	95.62	95.31 ± 0.22	4.69 ± 0.22	505.1	97.46	97.78	97.62 ± 0.11	2.38 ± 0.11	512.7
10,000 km	93.62	94.51	93.96 ± 0.25	6.04 ± 0.25	650.3	96.75	97.22	96.93 ± 0.14	3.7 ± 0.14	660.9

Table C.5: Linear SVM results using the complete set of samples the target symbol, at the distances from 1,000 km to 10,000 km, Y-Polarization.

Distance	MIN SA%	MAX SA%	Avg SA%	Avg SER	Avg NSE	Min BA%	Max BA%	Avg BA%	Avg BER	Avg NBE
1,000 km	100	100	100 ± 0	0 ± 0	0	100	100	100 ± 0	0 ± 0	0
2,000 km	99.95	99.99	99.98 ± 0.01	0.02 ± 0.01	2.5	99.98	100.00	99.99 ± 0.01	0.01 ± 0.01	2.5
3,000 km	99.79	99.92	99.85 ± 0.04	0.15 ± 0.04	16.5	99.89	99.96	99.92 ± 0.02	0.08 ± 0.02	16.6
4,000 km	99.54	99.67	99.61 ± 0.04	0.39 ± 0.04	42.1	99.77	99.83	99.8 ± 0.02	0.2 ± 0.02	42.2
5,000 km	98.94	99.17	99.06 ± 0.08	0.94 ± 0.08	100.9	99.47	99.59	99.53 ± 0.04	0.46 ± 0.06	101.3
6,000 km	98.25	98.66	98.44 ± 0.12	1.56 ± 0.12	168.4	99.13	99.33	99.21 ± 0.06	0.79 ± 0.06	169.4
7,000 km	97.22	97.74	97.57 ± 0.17	2.43 ± 0.17	262	98.60	98.86	98.77 ± 0.08	1.23 ± 0.08	265.2
8,000 km	96.26	96.96	96.61 ± 0.23	3.39 ± 0.23	365.3	98.09	98.48	98.28 ± 0.12	1.72 ± 0.12	369.9
9,000 km	95.12	95.73	95.42 ± 0.18	4.58 ± 0.18	492.9	97.53	97.86	97.68 ± 0.1	2.32 ± 0.1	500.1
10,000 km	93.95	94.44	94.15 ± 0.16	0.06 ± 0.002	629.4	96.94	97.19	97.03 ± 0.08	0.03 ± 0.001	639

Table C.6: Linear SVM results using the 33^{rd} sample of the target symbol, Y-Polarization.

Distance	MIN SA%	MAX SA%	Avg SA%	Avg SER	Avg NSE	Min BA%	Max BA%	Avg BA%	Avg BER	Avg NBE
1,000 km	99.99	100	˜100 ± 0.003	0.00001 ± 0.00003	0.1	˜100	100	˜100 ± 0.001	0.000005 ± 0.00001	0.1
2,000 km	99.96	99.99	99.98 ± 0.01	0.0002 ± 0.0001	1.9	99.98	˜100	99.99 ± 0.005	0.0001 ± 0.00005	1.9
3,000 km	99.81	99.93	99.87 ± 0.04	0.001 ± 0.0004	14.5	99.91	99.96	99.93 ± 0.02	0.001 ± 0.0002	14.6
4,000 km	99.54	99.66	99.59 ± 0.04	0.004 ± 0.0004	44.4	99.77	99.83	99.79 ± 0.02	0.002 ± 0.0002	44.5
5,000 km	98.88	99.12	99.02 ± 0.08	0.01 ± 0.001	105	99.44	99.56	99.51 ± 0.04	0.005 ± 0.0004	105.3
6,000 km	98.14	98.55	98.34 ± 0.13	0.02 ± 0.001	178.7	99.07	99.27	99.16 ± 0.06	0.01 ± 0.001	179.9
7,000 km	97.18	97.67	97.48 ± 0.15	0.03 ± 0.002	271.5	98.57	98.83	98.72 ± 0.08	0.01 ± 0.001	275.2
8,000 km	96.04	96.78	96.44 ± 0.25	0.04 ± 0.003	382.9	97.99	98.36	98.2 ± 0.13	0.02 ± 0.001	387.9
9,000 km	94.99	95.63	95.3 ± 0.22	0.05 ± 0.002	506.4	97.46	97.79	97.61 ± 0.11	0.02 ± 0.001	514
10,000 km	93.56	94.28	93.93 ± 0.21	0.06 ± 0.002	653.4	96.72	97.1	96.92 ± 0.11	0.03 ± 0.001	664

Table C.7: Linear SVM results using the central sample (33^{rd}) of the target and symbol either side at the distances from 1,000 km to 10,000 km, Y-Polarization.

Distance	MIN SA%	MAX SA%	Avg SA%	Avg SER	Avg NSE	Min BA%	Max BA%	Avg BA%	Avg BER	Avg NBE
1,000 km	99.99	100	˜100 ± 0.003	0.00001 ± 0.00003	0.1	˜100	100	˜100 ± 0.001	0.000005 ± 0.00001	0.1
2,000 km	99.97	99.99	99.98 ± 0.01	0.0002 ± 0.0001	1.7	99.99	˜100	99.99 ± 0.003	0.0001 ± 0.00003	1.7
3,000 km	99.83	99.95	99.9 ± 0.04	0.001 ± 0.0004	10.8	99.92	99.98	99.95 ± 0.02	0.001 ± 0.0002	10.9
4,000 km	99.61	99.74	99.68 ± 0.04	0.003 ± 0.0004	34.4	99.8	99.87	99.84 ± 0.02	0.002 ± 0.0002	34.5
5,000 km	98.99	99.28	99.14 ± 0.1	0.01 ± 0.001	92.7	99.49	99.64	99.57 ± 0.05	0.004 ± 0.0005	93
6,000 km	98.32	98.72	98.52 ± 0.12	0.01 ± 0.001	159.3	99.16	99.35	99.25 ± 0.06	0.01 ± 0.001	160.5
7,000 km	97.38	97.89	97.7 ± 0.15	0.02 ± 0.002	247.9	98.68	98.93	98.83 ± 0.08	0.01 ± 0.001	251.4
8,000 km	96.45	96.92	96.69 ± 0.18	0.03 ± 0.002	356.5	98.19	98.44	98.32 ± 0.09	0.02 ± 0.001	361.4
9,000 km	95.26	95.86	95.56 ± 0.21	0.04 ± 0.002	478.2	97.59	97.89	97.74 ± 0.11	0.02 ± 0.001	485.7
10,000 km	93.97	94.58	94.2 ± 0.19	0.06 ± 0.002	624.7	96.93	97.25	97.05 ± 0.1	0.03 ± 0.001	635.1

Table C.8: Linear SVM results using the central sample (33^{rd}) of the target and two symbols either side at the distances from 1,000 km to 10,000 km, Y-Polarization

Distance	MIN SA%	MAX SA%	Avg SA%	Avg SER	Avg NSE	Min BA%	Max BA%	Avg BA%	Avg BER	Avg NBE
1,000 km	99.99	100	˜100 ± 0.003	0.00001 ± 0.00002	0.1	˜100	100	˜100 ± 0.001	0.000005 ± 0.00001	0.1
2,000 km	99.98	99.99	99.99 ± 0.004	0.0001 ± 0.00005	1.4	99.99	˜100	99.99 ± 0.002	0.0001 ± 0.00002	1.4
3,000 km	99.85	99.97	99.9 ± 0.04	0.001 ± 0.0003	10.4	99.93	99.99	99.95 ± 0.02	0.0005 ± 0.0002	10.5
4,000 km	99.6	99.72	99.67 ± 0.04	0.003 ± 0.0004	35.4	99.8	99.86	99.84 ± 0.02	0.002 ± 0.0002	35.5
5,000 km	˜99	99.28	99.15 ± 0.09	0.01 ± 0.001	91.3	99.5	99.64	99.57 ± 0.05	0.004 ± 0.0005	91.6
6,000 km	98.39	98.72	98.52 ± 0.1	0.01 ± 0.001	159.5	99.2	99.36	99.25 ± 0.05	0.01 ± 0.001	160.6
7,000 km	97.45	97.88	97.7 ± 0.13	0.02 ± 0.001	247.9	98.71	98.92	98.83 ± 0.06	0.01 ± 0.001	251.5
8,000 km	96.43	96.94	96.72 ± 0.2	0.03 ± 0.002	353.4	98.18	98.45	98.34 ± 0.1	0.02 ± 0.001	358.3
9,000 km	95.25	95.85	95.59 ± 0.2	0.04 ± 0.002	474.7	97.59	97.88	97.76 ± 0.1	0.02 ± 0.001	482.2
10,000 km	93.95	94.64	94.22 ± 0.2	0.06 ± 0.002	622.3	96.92	97.28	97.06 ± 0.11	0.03 ± 0.001	632.9

Table C.9: Linear SVM results using the central sample (33^{rd}) of the target and three symbols either side at the distances from 1,000 km to 10,000 km, Y-Polarization

Distance	MIN SA%	MAX SA%	Avg SA%	Avg SER	Avg NSE	Min BA%	Max BA%	Avg BA%	Avg BER	Avg NBE
1,000 km	99.99	100	˜100 ± 0.003	0.00001 ± 0.00003	0.1	˜100	100	˜100 ± 0.001	0.000005 ± 0.00001	0.1
2,000 km	99.97	99.99	99.98 ± 0.01	0.0002 ± 0.0001	1.8	99.99	˜100	99.99 ± 0.004	0.0001 ± 0.00004	1.8
3,000 km	99.81	99.98	99.89 ± 0.04	0.001 ± 0.0004	11.8	99.91	99.99	99.94 ± 0.02	0.001 ± 0.0002	11.9
4,000 km	99.61	99.76	99.68 ± 0.05	0.003 ± 0.001	34.8	99.8	99.88	99.84 ± 0.02	0.002 ± 0.0002	34.9
5,000 km	99.02	99.35	99.16 ± 0.09	0.01 ± 0.001	90.6	99.51	99.67	99.58 ± 0.05	0.004 ± 0.0005	90.9
6,000 km	98.38	98.72	98.55 ± 0.1	0.01 ± 0.001	155.6	99.19	99.36	99.27 ± 0.05	0.01 ± 0.001	156.6
7,000 km	97.41	97.94	97.72 ± 0.15	0.02 ± 0.002	245	98.69	98.95	98.85 ± 0.08	0.01 ± 0.001	248.6
8,000 km	96.44	97.01	96.74 ± 0.22	0.03 ± 0.002	350.8	98.19	98.48	98.35 ± 0.11	0.02 ± 0.001	355.6
9,000 km	95.32	95.88	95.62 ± 0.2	0.04 ± 0.002	471.2	97.63	97.9	97.78 ± 0.1	0.02 ± 0.001	478.6
10,000 km	94.05	94.58	94.29 ± 0.18	0.06 ± 0.002	614.3	96.97	97.25	97.1 ± 0.09	0.03 ± 0.001	625

Table C.10: Linear SVM results using the central sample (33^{rd}) of the target and four symbols either side at the distances from 1,000 km to 10,000 km, Y-Polarization

Distance	MIN SA%	MAX SA%	Avg SA%	Avg SER	Avg NSE	Min BA%	Max BA%	Avg BA%	Avg BER	Avg NBE
1,000 km	99.99	100	˜100 ± 0.003	0.00001 ± 0.00003	0.1	˜100	100	˜100 ± 0.001	0.000005 ± 0.00001	0.1
2,000 km	99.97	99.99	99.98 ± 0.01	0.0002 ± 0.0001	1.8	99.99	˜100	99.99 ± 0.003	0.0001 ± 0.00003	1.8
3,000 km	99.8	99.98	99.89 ± 0.05	0.001 ± 0.001	11.5	99.9	99.99	99.95 ± 0.03	0.001 ± 0.0003	11.6
4,000km	99.64	99.74	99.69 ± 0.04	0.003 ± 0.0004	33.7	99.82	99.87	99.84 ± 0.02	0.002 ± 0.0002	33.8
5,000 km	99.03	99.36	99.16 ± 0.1	0.01 ± 0.001	89.9	99.52	99.68	99.58 ± 0.05	0.004 ± 0.001	90.2
6,000 km	98.39	98.76	98.56 ± 0.13	0.01 ± 0.001	155.3	99.19	99.38	99.27 ± 0.06	0.01 ± 0.001	156.4
7,000 km	97.4	97.9	97.72 ± 0.16	0.02 ± 0.002	245.6	98.69	98.93	98.84 ± 0.08	0.01 ± 0.001	249.1
8,000 km	96.45	97.03	96.76 ± 0.2	0.03 ± 0.002	348.7	98.2	98.48	98.36 ± 0.1	0.02 ± 0.001	353.4
9,000 km	95.31	95.88	95.62 ± 0.19	0.04 ± 0.002	471.6	97.62	97.89	97.78 ± 0.1	0.02 ± 0.001	479.2
10,000 km	94.04	94.66	94.33 ± 0.21	0.06 ± 0.002	610.2	96.97	97.3	97.12 ± 0.11	0.03 ± 0.001	620.5

Table C.11: Linear SVM results using the central sample (33^{rd}) of the target and two preceding symbols at the distances from 1,000 km to 10,000 km, Y-Polarization

Distance	MIN SA%	MAX SA%	Avg SA%	Avg SER	Avg NSE	Min BA%	Max BA%	Avg BA%	Avg BER	Avg NBE
1,000 km	99.99	100	˜100 ± 0.003	0.00001 ± 0.00003	0.1	˜100	100	˜100 ± 0.001	0.000005 ± 0.00001	0.1
2,000 km	99.96	99.99	99.98 ± 0.01	0.0002 ± 0.0001	2	99.98	˜100	99.99 ± 0.005	0.0001 ± 0.00005	2
3,000 km	99.82	99.96	99.88 ± 0.04	0.001 ± 0.0004	13.4	99.91	99.98	99.94 ± 0.02	0.001 ± 0.0002	13.5
4,000 km	99.60	99.7	99.64 ± 0.04	0.004 ± 0.0004	38.4	99.8	99.85	99.82 ± 0.02	0.002 ± 0.0002	38.5
5,000 km	98.96	99.21	99.12 ± 0.08	0.01 ± 0.001	94.9	99.48	99.61	99.56 ± 0.04	0.004 ± 0.0004	95.2
6,000 km	98.28	98.68	98.5 ± 0.12	0.02 ± 0.001	161.6	99.14	99.34	99.24 ± 0.06	0.01 ± 0.001	162.8
7,000 km	97.37	97.9	97.65 ± 0.16	0.02 ± 0.002	253.1	98.67	98.93	98.81 ± 0.08	0.01 ± 0.001	256.5
8,000 km	96.41	96.99	96.7 ± 0.21	0.03 ± 0.002	355.5	98.17	98.46	98.33 ± 0.11	0.02 ± 0.001	360.4
9,000 km	95.22	95.8	95.55 ± 0.21	0.04 ± 0.002	479.2	97.57	97.89	97.74 ± 0.11	0.02 ± 0.001	486.8
10,000 km	94.04	94.54	94.25 ± 0.18	0.06 ± 0.002	618.7	96.96	97.24	97.08 ± 0.1	0.03 ± 0.001	629.3

Table C.12: Linear SVM results using the complete set of samples of the target symbol and the preceding symbol at the distances from 1,000 km to 10,000 km, Y-Polarization

Distance	MIN SA%	MAX SA%	Avg SA%	Avg SER	Avg NSE	Min BA%	Max BA%	Avg BA%	Avg BER	Avg NBE
1,000 km	99.99	100	˜100 ± 0.003	0.00001 ± 0.00003	0.1	˜100	100	˜100 ± 0.001	0.000005 ± 0.00001	0.1
2,000 km	99.97	99.99	99.98 ± 0.01	0.0002 ± 0.0001	1.7	99.99	˜100	99.99 ± 0.004	0.0001 ± 0.00004	1.7
3,000 km	99.80	99.92	99.86 ± 0.03	0.001 ± 0.0003	14.9	99.9	99.96	99.93 ± 0.02	0.001 ± 0.0002	15
4,000 km	99.59	99.68	99.64 ± 0.03	0.004 ± 0.0003	39.2	99.80	99.84	99.82 ± 0.02	0.002 ± 0.0002	39.3
5,000 km	98.89	99.2	99.09 ± 0.1	0.01 ± 0.001	98.3	99.45	99.6	99.54 ± 0.05	0.005 ± 0.0005	98.6
6,000 km	98.27	98.64	98.44 ± 0.14	0.02 ± 0.001	167.8	99.14	99.32	99.22 ± 0.07	0.01 ± 0.001	168.8
7,000 km	97.31	97.84	97.61 ± 0.17	0.02 ± 0.002	257.1	98.64	98.9	98.79 ± 0.09	0.01 ± 0.001	260.3
8,000 km	96.37	96.99	96.67 ± 0.21	0.03 ± 0.002	359.1	98.15	98.49	98.31 ± 0.11	0.02 ± 0.001	363.6
9,000 km	95.1	95.83	95.48 ± 0.21	0.05 ± 0.002	486.6	97.51	97.91	97.71 ± 0.12	0.02 ± 0.001	493.5
10,000 km	94	94.42	94.2 ± 0.14	0.06 ± 0.001	624.5	96.96	97.17	97.05 ± 0.07	0.03 ± 0.001	634.4

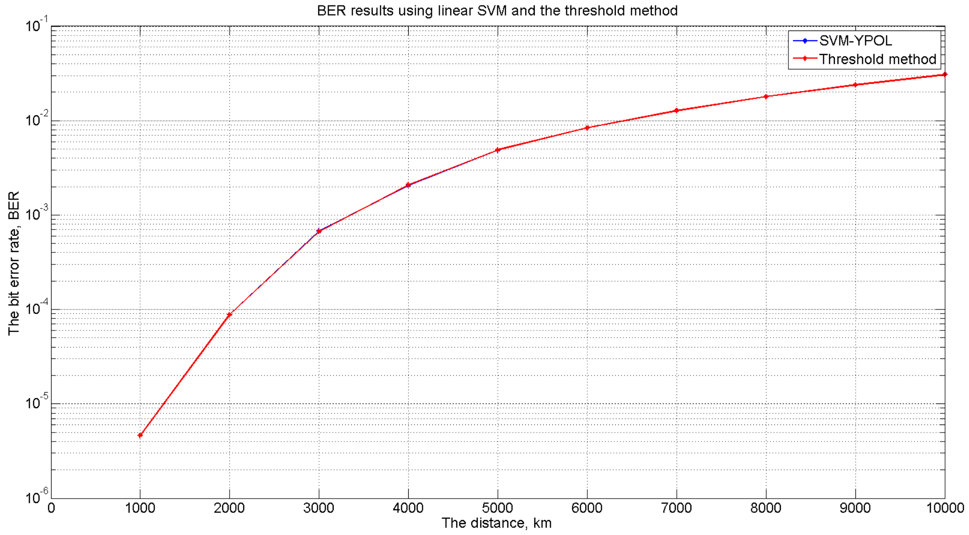


Figure C.2: A comparison between the BER (log-scaled) obtained using linear SVM and the threshold method. The input vector to our classifier is the central sample of the target symbol, at the distances from 1,000 km to 10,000 km, Y-Polarization.

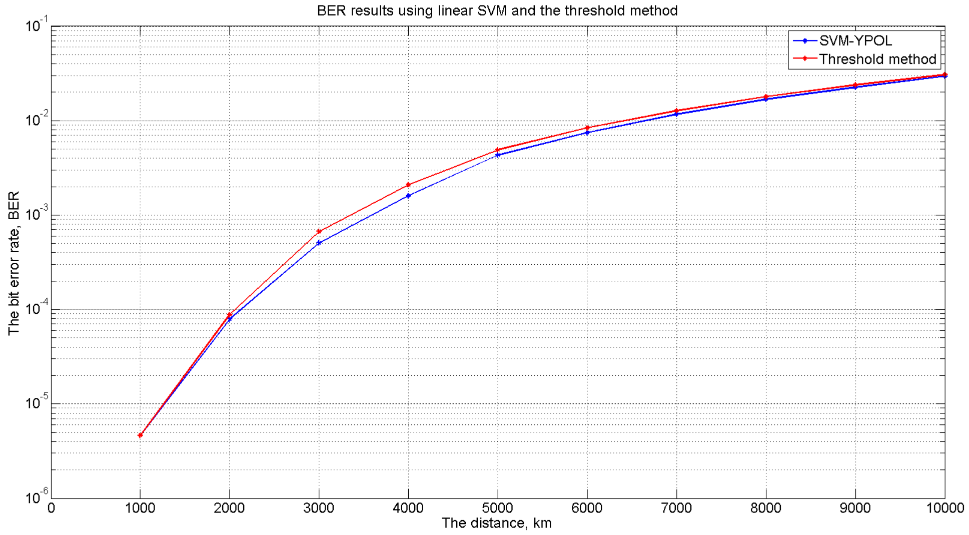


Figure C.3: A comparison between the BER (log-scaled) obtained using linear SVM and the threshold method. The input vector to our classifier is the central sample of the target symbol and symbol either side, at the distances from 1,000 km to 10,000 km, Y-Polarization.

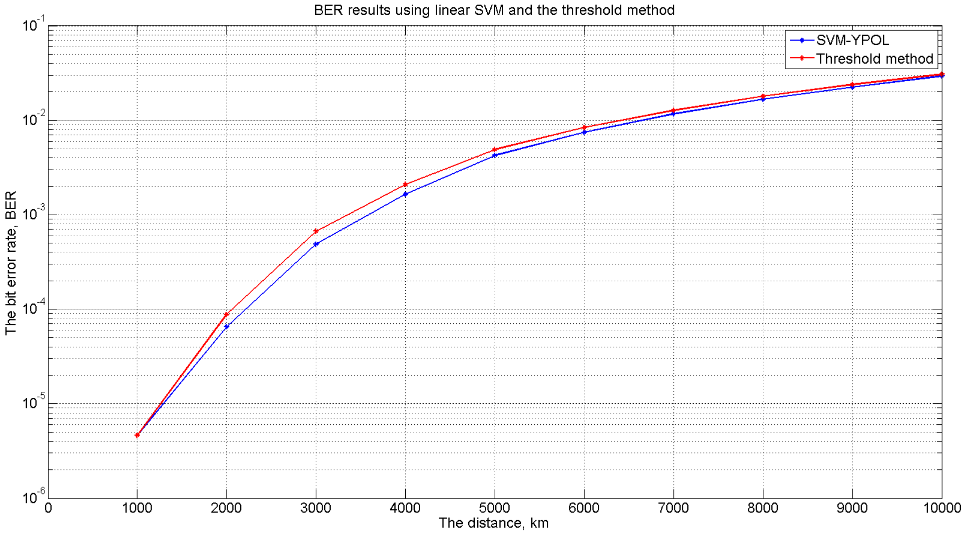


Figure C.4: A comparison between the BER (log-scaled) obtained using linear SVM and the threshold method. The input vector to our classifier is the central sample of the target symbol and two symbols either side, at the distances from 1,000 km to 10,000 km, Y-Polarization.

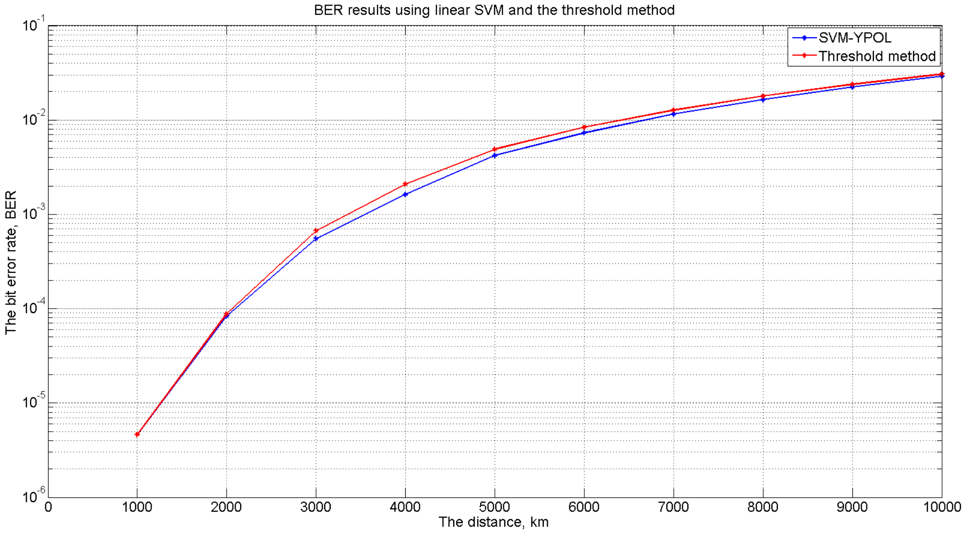


Figure C.5: A comparison between the BER (log-scaled) obtained using linear SVM and the threshold method. The input vector to our classifier is the central sample of the target symbol and three symbols either side, at the distances from 1,000 km to 10,000 km, Y-Polarization.

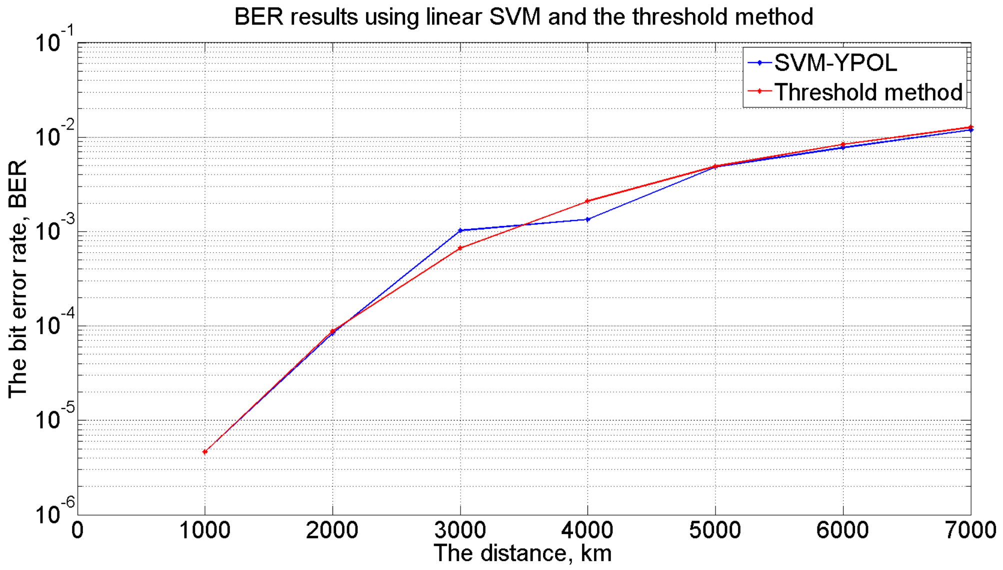


Figure C.6: A comparison between the BER (log-scaled) obtained using linear SVM and the threshold method. The input vector to our classifier is the central sample of the target symbol and four symbols either side, at the distances from 1,000 km to 10,000 km, Y-Polarization.

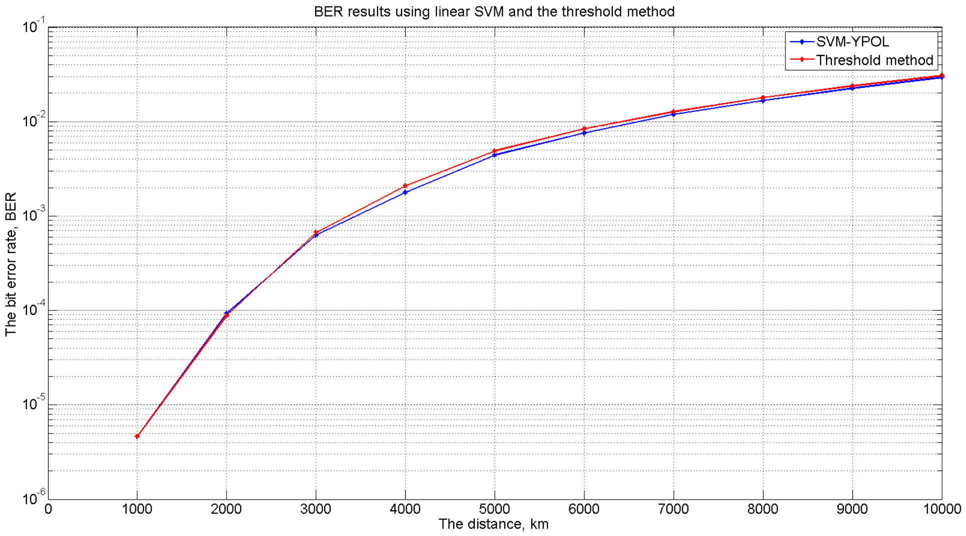


Figure C.7: A comparison between the BER (log-scaled) obtained using linear SVM and the threshold method. The input vector to our classifier is the central sample of the target symbol and two preceding symbols, at the distances from 1,000 km to 10,000 km, Y-Polarization.

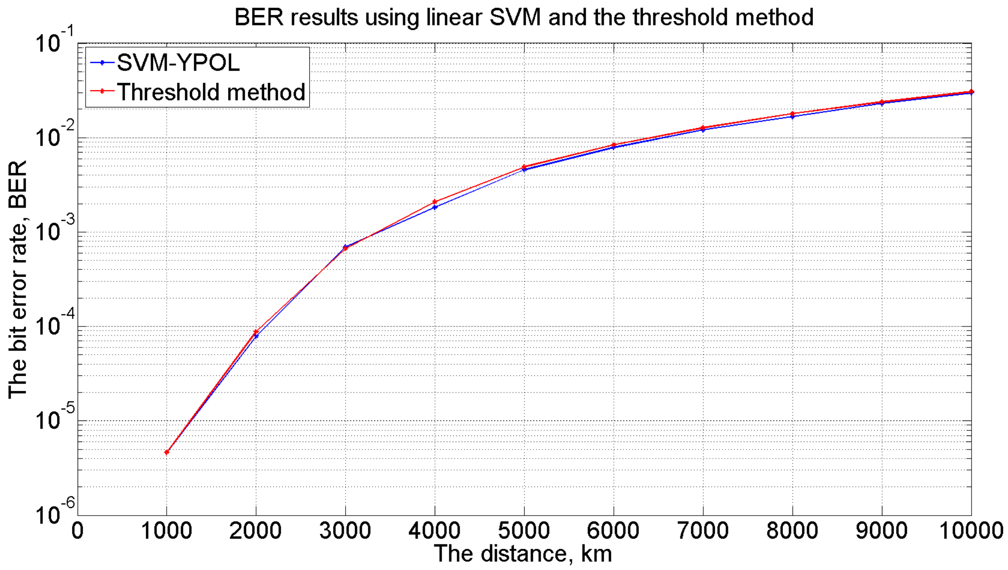


Figure C.8: A comparison between the BER (log-scaled) obtained using linear SVM and the threshold method. The input vector to our classifier is the complete set of samples of the target and preceding symbol, at the distances from 1,000 km to 10,000 km, Y-Polarization.

C.2 Experiments and Results using an SVM with Wavelets (Chapter 7)

C.2.1 Results using Optical Data (Complex Data) (Section 7.2)

Table C.13: Results of linear SVM based on wavelets, optical data, The letters P, T and S denote that preceding, target and succeeding symbol respectively.

(a) Distance 8,000 km

Method	**No. of symbols and sample**	**No. of features**	**Type and level of (WT)**	**SER** ($\times 10^{-4}$)	**BER** ($\times 10^{-4}$)	**P-Value**
Threshold	1(Central/T)	-	No/0	370.17±14.76	187.27±7.69	
SVM	1(Central/T)	1 × 2	No/0	370.54±16.23	187.41±8.36	0.85
			Haar/1	363.67±16.13	183.97±8.48	0.01
			Haar/2	358.56±15.93	181.51±8.55	0.004
			db4/2	605.40±46.58	307.35±23.55	5.42E-08
	1(Whole/T)	64 × 2	No/0	356.24±13.74	179.84±7.36	0.001
		32 × 2	Haar/1	356.24±13.21	179.93±7.15	0.003
		16 × 2	Haar/2	354.94±12.87	179.33±7.02	0.0005
	7(Central/3P,T,3S)	7 × 2	No/0	339.9±9.8	171.95±5.41	0.000004
			Haar/2	332.65±11.34	168.32±6.14	2.08E-07
			db4/2	425.08±21.79	216.12±11.84	7.23E-06

(b) Distance 9,000 km

Method	**No. of symbols and sample**	**No. of features**	**Type and level of (WT)**	**SER** ($\times 10^{-4}$)	**BER** ($\times 10^{-4}$)	**P-Value**
Threshold	1(Central/T)	-	No/0	488.48±16.13	247.72±8.67	**P-value**
SVM	1(Central/T)	1 × 2	No/0	489.13±14.45	248.05±7.87	0.49
			Haar/1	484.21±17.06	245.64±8.99	0.06
			Haar/2	479.10±19.23	242.80±10.42	0.0003
			db4/2	722.98±23.52	368.92±12.44	2.51E-12
	1(Whole/T)	64 × 2	No/0	474.46±16.94	240.53±8.68	0.01
		32 × 2	Haar/1	473.07±17.25	239.92±8.88	0.003
		16 × 2	Haar/2	474.37±16.65	240.57±8.52	0.002
	7(Central/3P,T,3S)	7 × 2	No/0	456.11±15.83	231.35±8.64	0.0000001
			Haar/2	451.28±15.06	228.75±8.37	5.64E-07
			db4/2	558.2±29.34	285.14±15.54	1.41E-06

Table C.13: Results of linear SVM based on wavelets, optical data (continued)

(c) Distance 10,000 km

Method	**No. of symbols and sample**	**No. of features**	**Type and level of (WT)**	**SER** $(\times 10^{-4})$	**BER** $(\times 10^{-4})$	**P-Value**
Threshold	1(Central/T)	-	No/0	607.08±16.68	309.53±9.98	**P-value**
SVM	1(Central/T)	1 × 2	No/0	610.33±19.01	311.2±10.95	0.06
			Haar/1	604.01±18.35	307.67±10.67	0.1
			Haar/2	593.8±18.47	302.75±10.63	0.01
			db4/2	871.19±25.19	447.39±13.22	2.36E-11
	1(Whole/T)	64 × 2	No/0	588.5±23.71	300.06±13.05	0.01
		32 × 2	Haar/1	586.27±24.62	298.89±13.62	0.005
		16 × 2	Haar/2	584.88±23.95	298.2±13.14	0.002
	7(Central/3P,T,3S)	7 × 2	No/0	574.27±15.81	292.85±9.561	0.000004
			Haar/2	566.84±14.84	289.13±9.05	2.47E-06
			db4/2	694.38±29.46	356.85±15.57	3.43E-08

Table C.13: The linear SVM results using different input vectors of optical waves before and after using wavelet transforms at the distances from 8,000 km to the maximum 10,000 km, compared with the threshold method result. SER denotes the average of symbol error ratio. BER denotes the average of bit error ratio. (×2) denotes that each an optical signal input is represented by ($\sin\theta$,$\cos\theta$). The letters P, T and S denote that preceding, target and succeeding symbol, respectively. In $M(X/NP,T,NS)$, M is the number of symbols, X is the number and position of the samples from each using symbols, and N is the number of symbols that are used from preceding (P) and succeeding (S) of the target symbol (T). The text in blue color denotes corresponding results are obtained without applying wavelets. The text in red color refer to the best wavelet result.

C.2.2 Applying Wavelet Transform (WT) on Optical Transmission Data

The experiment investigates whether applying WT on the optical transmission data using two different ways can provide the same results or not. These two methods are:

1. **Method 1:** Applying the WT on the complex number, then calculate the $\sin\theta$ and $\cos\theta$ values from the extracted features.

2. **Method 2:** Calculate the phase value of the complex number, then compute $\sin\theta$ and $\cos\theta$ values. Then applying the WT on the resulted values.

In this experiment, I applies the Haar wavelet (level 2), using two input vectors to the linear SVM classifier:

1. The whole set of samples of the target symbol. After using the WT, this point means that using the complete set of values (coefficients) of the approximation part, which is extracted from the target symbol, as an input to the classifier.

2. The central sample of the target. After using the WT, this point means that using the central coefficient value of the approximation part, which is extracted from the target symbol, as an input to the classifier.

The optical transmitted signal, which is used herein, is received at the distance of 10,000 km. Table C.14 shows the linear SVM results obtained using the previous two methods. There is a slightly difference in the prediction accuracy between both methods when the input vector is the whole set of samples of the target symbol. But when the input vector is only the central sample of the target symbol, the results are the same using both methods. I adopted the first method in my experiments.

Method	**Input vector**	**NSE**	**SAR%**	**NBE**	**BAR%**
Method 1	ALL	626	94.19	637	97.04
	Mid	652	93.95	662	96.93
Method 2	ALL	628	94.17	639	97.03
	Mid	652	93.95	662	96.93

Table C.14: Comparing of using two ways to prepare the input vector for the SVM classifier. The results obtained from the first data set of the second type of data at the distance of 10,000 km. ALL denotes the whole set of samples of the target symbol, and Mid denotes the central sample of the target symbol. NSE and NBE denote the number of symbol and bit errors, respectively. SAR and BAR denote the symbol and bit accuracy rate, respectively. As it has shown when the input vector is the whole set of samples of the target symbol, there is a slightly difference between both methods. But when the input vector is only the central sample, the results are the same using both methods.

C.3 Results using Data based on Real Text (Chapter 8)

C.3.1 Results using a Linear SVM (Section 8.1)

Table C.15: Results using a Linear SVM

(a) Distance 1,000 km

Method	No. of symbols and samples	No. of features	SER ($\times 10^{-4}$)	BER ($\times 10^{-4}$)	P-Value
Threshold	1(Central/T)	-	35.81±14.28	17.9±7.14	
SVM	1(Whole/T)	64 × 2	28.69±9.9	14.34±4.95	0.02
	1(Central/T)	1 × 2	34.59±13.38	17.29±6.69	0.41
	3(Central/P,T,S)	3 × 2	22.18±9.04	11.09±4.52	0.001
	5(Central/2P,T,2S)	5 × 2	18.32±7.09	9.16±3.55	0.001
	7(Central/3P,T,3S)	7 × 2	12.83±5.79	6.41±2.9	0.0002
	9(Central/4P,T,4S)	9 × 2	8.76±5.99	4.38±3	5.77E-05
	11(Central/5P,T,5S)	11 × 2	9.78±6.01	4.89±3	6.26E-05
	13(Central/6P,T,6S)	13 × 2	6.93±4.47	3.46±2.23	7.32E-05
	21(Central/10P,T,10S)	21 × 2	3.67±3.55	1.83±1.78	4.34E-05

(b) Distance 2,000 km

Method	No. of symbols and samples	No. of features	SER ($\times 10^{-4}$)	BER ($\times 10^{-4}$)	P-Value
Threshold	1(Central/T)	-	158.5±34.84	79.35±17.42	
SVM	1(Whole/T)	64 × 2	121.06±22.88	60.94±11.72	6.79E-05
	1(Central/T)	1 × 2	158.09±31.41	79.15±15.69	0.84
	3(Central/P,T,S)	3 × 2	97.27±18.28	48.74±9.2	4.79E-05
	5(Central/2P,T,2S)	5 × 2	82.64±14.32	41.42±7.14	2.15E-05
	7(Central/3P,T,3S)	7 × 2	61.68±11.33	30.94±5.84	6.15E-06
	9(Central/4P,T,4S)	9 × 2	42.35±9.36	21.28±4.64	1.95E-06
	11(Central/5P,T,5S)	11 × 2	48.47±10.96	24.34±5.48	3.46E-06
	13(Central/6P,T,6S)	13 × 2	37.89±10.85	19.04±5.36	3.13E-06
	21(Central/10P,T,10S)	21 × 2	25.07±9.66	12.64±4.81	8.17E-07

(c) Distance 3,000 km

Method	No. of symbols and samples	No. of features	SER ($\times 10^{-4}$)	BER ($\times 10^{-4}$)	P-Value
Threshold	1(Central/T)	-	350.77±39.35	176±19.75	
SVM	1(Whole/T)	64 × 2	256.36±28.76	129.81±14.81	9.96E-06
	1(Central/T)	1 × 2	355.04±37.17	178.13±18.65	0.02
	3(Central/P,T,S)	3 × 2	225.27±18.72	113.35±9.29	4.53E-07
	5(Central/2P,T,2S)	5 × 2	203.33±20.32	102.28±10.22	5.13E-07
	7(Central/3P,T,3S)	7 × 2	156.55±21.82	78.99±11.09	1.15E-08
	9(Central/4P,T,4S)	9 × 2	127.46±22.95	64.34±11.9	8.66E-08
	11(Central/5P,T,5S)	11 × 2	133.8±16.19	67.41±7.9	7.05E-08
	13(Central/6P,T,6S)	13 × 2	104.7±16.43	52.86±8.12	3.1E-08
	21(Central/10P,T,10S)	21 × 2	73.38±20.67	37.51±10.39	9.32E-09

Table C.15: Results using a Linear SVM (continued)

(d) Distance 4,000 km

Method	**No. of symbols and samples**	**No. of features**	**SER** ($\times 10^{-4}$)	**BER** ($\times 10^{-4}$)	**P-Value**
Threshold	1(Central/T)	-	547.52±44.22	275.9±22.13	
SVM	1(Whole/T)	64 × 2	406.93±33.5	209.67±17.96	1.43E-08
	1(Central/T)	1 × 2	551.38±39.94	277.73±20.15	0.46
	3(Central/P,T,S)	3 × 2	377.29±36.96	191.29±19.36	2.56E-07
	5(Central/2P,T,2S)	5 × 2	348.66±36.57	177.28±19.26	2.95E-08
	7(Central/3P,T,3S)	7 × 2	287.85±23.74	146.78±12.88	3.54E-09
	9(Central/4P,T,4S)	9 × 2	230.29±25.21	118.1±13.09	6.08E-10
	11(Central/5P,T,5S)	11 × 2	243.56±25.57	124.84±13.84	1.54E-09
	13(Central/6P,T,6S)	13 × 2	204.91±25.73	105.31±14.04	8.23E-10
	21(Central/10P,T,10S)	21 × 2	144.52±26.74	74.71±14.56	7.4E-11

(e) Distance 5,000 km

Method	**No. of symbols and samples**	**No. of features**	**SER** ($\times 10^{-4}$)	**BER** ($\times 10^{-4}$)	**P-Value**
Threshold	1(Central/T)	-	762.38±39.66	386.99±20.88	
SVM	1(Whole/T)	64 × 2	568.68±30.15	297.46±18.01	5.52E-08
	1(Central/T)	1 × 2	767.06±35.1	389.22±18.55	0.25
	3(Central/P,T,S)	3 × 2	555.35±43.19	283.37±23.68	5.64E-08
	5(Central/2P,T,2S)	5 × 2	514.95±47.38	264.09±25.64	3.22E-08
	7(Central/3P,T,3S)	7 × 2	434.43±39.57	224.24±22.42	1.33E-09
	9(Central/4P,T,4S)	9 × 2	357.95±37.06	185.29±20.57	9.47E-11
	11(Central/5P,T,5S)	11 × 2	364.33±40.24	189.6±22.23	3.25E-11
	13(Central/6P,T,6S)	13 × 2	316.94±26.7	165.5±14.97	1.27E-10
	21(Central/10P,T,10S)	21 × 2	244.2±32.37	127.3±18.36	2.59E-11

(f) Distance 6,000 km

Method	**No. of symbols and samples**	**No. of features**	**SER** ($\times 10^{-4}$)	**BER** ($\times 10^{-4}$)	**P-Value**
Threshold	1(Central/T)	-	973.37±41.21	497.26±20.94	
SVM	1(Whole/T)	64 × 2	756.88±45.56	401.13±24.39	1.54E-08
	1(Central/T)	1 × 2	977.44±41.55	499.4±21.25	0.21
	3(Central/P,T,S)	3 × 2	752.95±36.42	387.26±18.97	1.1E-09
	5(Central/2P,T,2S)	5 × 2	703.43±39.88	363.31±20.76	3.45E-09
	7(Central/3P,T,3S)	7 × 2	613.17±41.45	318.09±22.56	5.35E-10
	9(Central/4P,T,4S)	9 × 2	514.12±29.76	268.16±17.28	1.22E-10
	11(Central/5P,T,5S)	11 × 2	521.34±33.88	272.58±17.91	6.23E-11
	13(Central/6P,T,6S)	13 × 2	465.63±39.75	243.51±22.23	2.51E-10
	21(Central/10P,T,10S)	21 × 2	366.91±38.11	193.54±20.71	8.26E-12

Table C.15: Results using a Linear SVM (continued)

(g) Distance 7,000 km

Method	No. of symbols and samples	No. of features	SER ($\times 10^{-4}$)	BER ($\times 10^{-4}$)	P-Value
Threshold	1(Central/T)	-	1187±49.72	611.3±24.33	
SVM	1(Whole/T)	64 × 2	937.76±45.17	501.03±25.71	1.39E-08
	1(Central/T)	1 × 2	1178.86±55.07	607.74±27.79	0.24
	3(Central/P,T,S)	3 × 2	936.71±48.19	487.08±26.79	3.29E-08
	5(Central/2P,T,2S)	5 × 2	894.95±46.09	466.92±25.45	1.65E-08
	7(Central/3P,T,3S)	7 × 2	775.62±60.72	408.57±32.64	2.22E-08
	9(Central/4P,T,4S)	9 × 2	682.1±39.85	360.19±22.69	3.96E-10
	11(Central/5P,T,5S)	11 × 2	674.49±51.52	357.20±27.6	4.18E-10
	13(Central/6P,T,6S)	13 × 2	613.71±45.09	326.2±24.32	8.19E-11
	21(Central/10P,T,10S)	21 × 2	495.73±36.64	263.56±20.81	9.36E-11

(h) Distance 8,000 km

Method	No. of symbols and samples	No. of features	SER ($\times 10^{-4}$)	BER ($\times 10^{-4}$)	P-Value
Threshold	1(Central/T)	-	1367.47±55.89	706.12±28.67	
SVM	1(Whole/T)	64 × 2	1112.54±51.17	599.81±29.68	9.77E-08
	1(Central/T)	1 × 2	1370.12±61.46	707.44±31.93	0.66
	3(Central/P,T,S)	3 × 2	1130.65±55.58	590.05±30.99	2.81E-08
	5(Central/2P,T,2S)	5 × 2	1075.7±57.17	563.9±30.41	1.39E-08
	7(Central/3P,T,3S)	7 × 2	961.08±60.37	507.1±32.26	2.22E-10
	9(Central/4P,T,4S)	9 × 2	843.56±51.75	446.11±28.05	1.09E-10
	11(Central/5P,T,5S)	11 × 2	843.52±52.33	445.89±27.3	2.29E-11
	13(Central/6P,T,6S)	13 × 2	765.05±54.64	406.46±29.23	7.17E-11
	21(Central/10P,T,10S)	21 × 2	634.54±53.54	338.37±29.88	2.17E-10

Table C.15: The linear SVM results at the distances from 1,000 km to the maximum 8,000 km, compared with the threshold method result. SER denotes the average of symbol error ratio. BER denotes the average of bit error ratio. (×2) denotes that each an optical wave/symbol input is represented by ($\sin\theta$,$\cos\theta$). The letters P, T and S denote that preceding, target and succeeding symbol respectively. In $M(X/NP,T,NS)$, M is the number of symbols, X is the total number and position of the samples from each used symbols, and N is the number of symbols that are used from preceding and succeeding of the target symbol T. The text in blue color denotes corresponding results are obtained using the threshold method. The text in green color denotes corresponding results are obtained using the SVM with only the central sample of the target wave/symbol. It can be seen that there is not any BER improvement over the threshold method when using SVM with only the central sample of the target wave. The text in red color refer to the best result. Using more than one sample from the target symbol (e.g the whole set of samples) improves the BER. Increasing the number of the neighbouring central samples helps to improve the BER.

Appendix D

Publications on this Thesis

I have published three conference papers during my studying as I mentioned in the introduction in Chapter 1:

- **Reducing Bit Error Rate of Optical Data Transmission with Neighboring Symbol Information Using a Linear Support Vector Machine** by Weam M. Binjumah, Alexey Redyuk, Neil Davey, Rod Adams and Yi Sun - presented in European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, from 7 to 11 September, 2015 in Porto, Portugal.
- **Error Correction over Optical Transmission** by Weam M. Binjumah, Alexey Redyuk, Rod Adams, Neil Davey and Yi Sun - presented in ICPRAM 2017; 6th International Conference on Pattern Recognition Applications and Methods, from 24 to 26 February, 2017 in Porto, Portugal.
- **Investigating Optical Transmission Error Correction using Wavelet Transforms** by Weam M. Binjumah, Alexey Redyuk, Rod Adams, Neil Davey and Yi Sun - presented in ESANN 2017; Investigating Optical Transmission Error Correction using Wavelet Transforms, from 26 to 28 April, 2017 in Bruges, Belgium.

The papers are presented on the following pages as published:

Reducing Bit Error Rate of Optical Data Transmission with Neighboring Symbol Information Using a Linear Support Vector Machine

Weam M. Binjumah[1], Alexey Redyuk[2], Neil Davey[1], Rod Adams[1], Yi Sun[1]

[1]The School of Computer Science, University of Hertfordshire, Hatfield, AL10 9AB, UK, weam.m.j@gmail.com, [n.davey, R.G.Adams, y.2.sun]@herts.ac.uk.
[2]Novosibirsk State University, Novosibirsk, 2 Pirogova Street, 630090, Russia, alexey.redyuk@gmail.com.

Abstract. Improvement of bit error rate in optical transmission systems is a crucial and challenging problem. Whenever the distance travelled by the pulses increases, the difficulty of it being classified correctly also increases. We apply a linear support vector machine for classifying Wavelength-Division-Multiplexing Non-Return-to-Zero Dual-Polarization Quadrature-Phase-Shift-Keying (WDM NRZ DP QPSK) signals with neighboring information. We demonstrate a bit error rate (BER) improvement in comparison with the traditional threshold method.

Key words: Support Vector Machine (SVM), Classification, Machine Learning, Bit Error Rate (BER), Signal Processing, Optical Data Transmission.

1 Introduction

In optical communication systems, there are many different causes of loss in the quality of the signal [1]. Increasing the distance travelled by the pulses leads to an increase in the number of bit errors. As is normally the case the phase is measured at the mid point of the pulse because that represents the highest power level. Both linear and nonlinear distortions can be present in the signal. Pulse linear distortion can be modeled, and therefore factor it out. The same is not true for non-linear distortion. And so, we are using machine learning technique to detect and correct such distortions. Metaxas et al. demonstrates that linear Support Vector Machines (SVM) outperformed other trainable classifiers for error correction in optical data transmission, such as using neural networks [2].

In this paper, we investigate the most significant samples that can be used for training the linear SVM classifier to reduce the number of bit errors during the classification process. In particular, we take into account the neighboring information from each symbol.

2 Motivation

There is an on going need to increase global bandwidth. Due to its capacity and speed at long distances optical links are currently used and probably will also be used in the foreseeable future. The greater distance signals travel the more likely noise will corrupt the signal, giving rise to an ever increasing bit error rate (BER). Errors can be dealt with by adding check bits to the signal, but this uses bandwidth. In our work, we use an alternative approach in which we train a machine learning system to automatically detect and correct errors.

3 Background

3.1 Related Work

One technique that can be used to reduce the effect of signal distortion is using machine learning systems. In earlier works, we demonstrated the possibility of using simple artificial neural networks to help error correction and detection accurately at a high speed [3]. Since the system is trainable, it could cope with the change over the time of a channel's characteristics. The decoder used a perceptron which can classify at high speed. In fact, it could be built in hardware and give real time error correction even at bit rates of over 100 GHz. One problem of using a perceptron is to regularize the decision boundary to avoid over/under fitting. [2] demonstrated the efficiency of a trainable classifier at improving the bit error rate. It is known that a support vector machine (SVM) is a better classifier than perceptron. In the work reported here, we show how a linear SVM can be used to perform error detection and correction.

3.2 Computational Model

Fig. 1 shows the link configuration under investigation. In the numerical model we simulated a typical non-return-to zero (NRZ)-DP-QPSK transmitter which consisted of a laser, modulated at 30 Gbaud using a 2^{15} pseudorandom binary sequence (PRBS) and filtered by a 2^{nd} order super Gaussian filter with 30 GHz 3 dB bandwidth. The signal channel at 1550 nm was propagated over the fiber along with 10 50 GHz spaced similar crosstalk channels, with decorrelated PRBS sequences. In order to model signal propagation over the nonlinear fiber a system of coupled nonlinear Schrödinger equations (CNLSE) was used. CNLSE has been solved using the split-step Fourier method [4]. After each erbium doped fiber amplifier (EDFA), the signal was noise loaded with the white Gaussian noise, calculated using a 6 dB amplifier noise figure. At the receiver side the signal was filtered by a 2^{nd} order super Gaussian filter with 30 GHz 3 dB bandwidth. The chromatic dispersion was fully compensated by multiplying the Fourier transformed optical field with the reverse dispersion function. For phase estimation, an algorithm based on the 4^{th}-power Viterbi–Viterbi method has been used. The effects of signal digitization, polarization rotation and PMD have not been considered in the simulations.

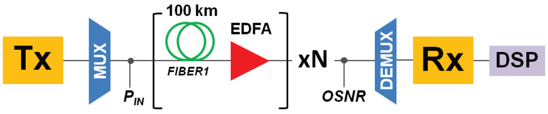
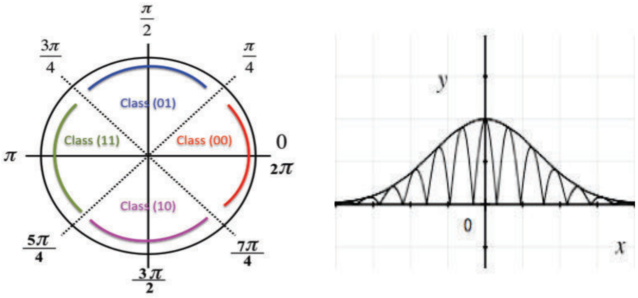


Fig. 1: The fiber link configuration.



(a) Four different variant of the phase for PI/4 QPSK modulation with Gray coding.

(b) Gaussian carrier wave superimposed on an sinusoidal light wave

Fig. 2: PI/4 QPSK modulation and Gaussian carrier wave.

The data we are analyzing consists of 32,768 symbols. Our data was generated by a dual-polarization optical communication system, X and Y polarization. The simulation process was repeated 10 times with different random realizations of the amplified spontaneous emission (ASE) noise and input PRBS, each run generates 32,768 symbols. The signal was detected at intervals of 1,000 km to a maximum distance 10,000 km.

Each pulse was decoded into one of four symbols; see Fig. 2(a), according to its phase. Each data point has a corresponding two-bit label for each run. Each run generates one data set. Fig. 2(b) shows a Gaussian carrier wave superimposed on an sinusoidal light wave. In this paper we focus on X-Polarization data and use Y-Polarization data for verification of our results. Each pulse is represented by 64 equally spaced phase samples. Fig. 3 shows the phase of central sample of one of the data sets at 10,000 km. As we can see from Fig. 3, the phase of some pulses is different from their actual encodings (provided by labels). This means these signals were distorted after traveling 10,000 km.

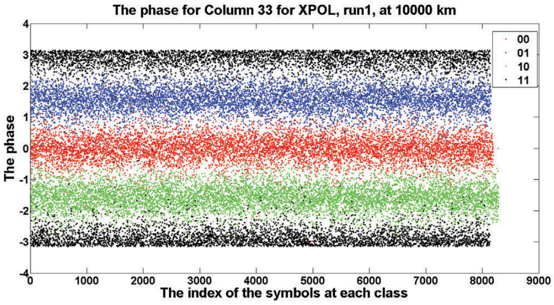


Fig. 3: The phase of the sample 33, XPOL, run1, at 10,000 km.

4 Machine Learning Decoder

In this work, we used a trainable classifier to help decode the received pulses. Note that the data used in this work is simulated data. Since this decoding must take place in real time, the classifier must be capable of being hardware based. To this end, we used a linear support vector machine (SVM) [2]. SVM is a soft maximum margin classifier, see Fig. 4. It has only one non-learnable parameter, which is the regularizing cost parameter C [2]. This parameter allows the cost of misclassification to be specified. The Linear SVM is arguably the most successful method in machine learning.

In order to have a continuously varying phase value, both the Sine and Cosine of the phase were used as input to our decoding (The phase angle has a discontinuity at 0 / 2π).

We had used a variety of inputs to our decoder as can be seen in Table 1. From Fig. 5, we see the reason behind using symbols on either side of the symbol being analyzed. Fig. 5 shows three consecutive symbols at 0 km and at 10,000 km. At 10,000 km the middle symbol was incorrectly decoded (the dotted line) when using the threshold method. As we can see from Fig.5, the first symbol has a phase of (π) whereas the phase of the middle symbol is (0) or (2π). However, at 10,000 km the central symbol has been degraded. At the distance of 10,000 km, the first symbol tries to pull the second symbol from (2π) to (π), which led to the prediction of the middle symbol at the middle point as (3π/2). From the above observation, our hypothesis is that the neighboring symbols can affect the target symbol, for which we want to predict the label. Therefore, in this work we investigate the effect of using the symbol either side of the target in an attempt to reduce the bit error rate. Table 1 shows a description of some experiments that had done on our data in terms of the samples used and features considered.

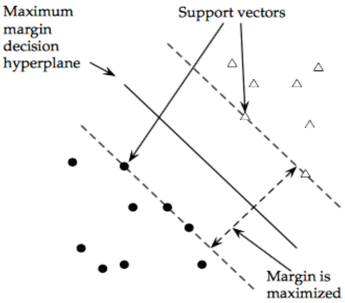


Fig. 4: The margin and support vectors[5].

Table 1: A description of the experiment that were implemented on our data. In the experiment D we used the central sample from the target symbol and symbol either side. In the experiment E we used the central sample from the target symbol and two symbols either side. In the experiment F we used the central sample from the target symbol and two preceding symbols.

Exp.	Method	Sample	Feature's type	Symbol
A	Threshold	Central	Phase value (ϑ)	One symbol
B	Linear SVM	Central	sin(ϑ),cos(ϑ)	One symbol
C	Linear SVM	64 samples	sin(ϑ),cos(ϑ)	One symbol
D	Linear SVM	3 samples	sin(ϑ),cos(ϑ)	Three symbols
E	Linear SVM	5 samples	sin(ϑ),cos(ϑ)	Five symbols
F	Linear SVM	3 samples	sin(ϑ),cos(ϑ)	Three symbols

5 Experiments Setup and Results

In each experiment, we divided each data set into 2/3 training and 1/3 testing. LIBSVM [6] program was used to classify our data sets linearly. We divided the data in the training set into 5 parts and used 5-fold cross validation to find a perfect value for C.

Table 2 shows a comparison between the number of bit errors that obtained from using the threshold method (column A), and the linear SVM using different numbers of samples and symbols (columns B, C, D, E and F, (see Table 1 for details)). Each number in Table 2 from column A to F is the average of the number of bit errors over 10 data sets; and the best result in each distance has been shown in bold font. As we can see from Table 2, the best results obtained so far is from the linear SVM when using 3 samples, the central sample from the target symbol and symbols either side. And also when using 5 samples, the central sample from the target symbol and two symbols either side. Compared with the result obtained from using the traditional threshold method, the linear SVM has showed a useful improvement when using more than one sample. Especially, when those samples were taken from more than one symbol. For example, the experiment D correctly classified the middle symbol shown in Fig. 5, whereas experiment A, B and C, which did not involve neighboring information misclassified the symbol.

Fig. 6 shows the percentage of the improvement over the threshold method against the distance for the experiments D in Fig. 6 (a) and E in Fig. 6 (b). As can be seen using the neighboring information, an improvement can be obtained from the distance of 2,000 km; with the best improvement was obtained at the distance 3,000 km.

Table 2: How the number of bit errors varies with distance (Note that each number in columns A to F is the average of the number of bit errors over 10 data sets).

Distance	A	B	C	D	E	F
0 km	0	0	0	0	0	0
1,000 km	0	0	0	0	0	0
2,000 km	2.3	3	3.2	2	**1.7**	1.8
3,000 km	16.7	16.3	16.5	**10.6**	11.2	12.9
4,000 km	50.9	50.7	46.9	**37.6**	39.2	40.3
5,000 km	103.7	103.3	98.3	89.9	**88.9**	92.2
6,000 km	185.5	184.9	172.3	165.3	**163.3**	165.2
7,000 km	284.4	284.7	273.1	**260.5**	262.1	262.3
8,000 km	403.3	403.6	386.6	**372.5**	377.9	378.2
9,000 km	533.5	534.2	517.4	511.9	**509.3**	509.9
10,000 km	666.6	670.2	642.1	639.8	**632.4**	634

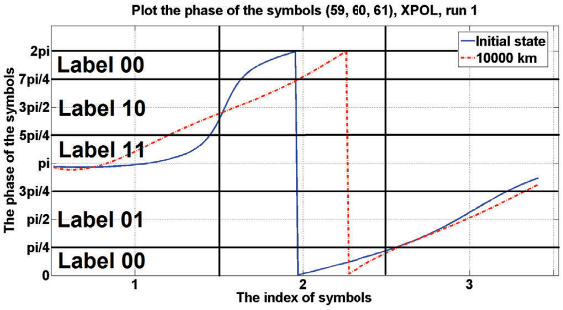
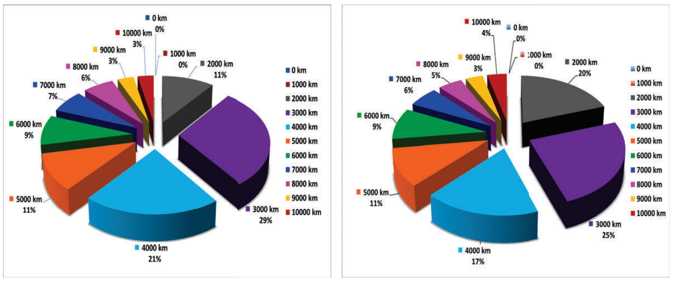


Fig. 5: Three contiguous symbols to show the effect of the symbols either side.



(a) Experiment D.

(b) Experiment E.

Fig. 6: The improvement over the threshold method (%). (a) Experiment D: applying linear SVM using 3 samples (the 33^{rd}central sample from the target symbol and symbol either side). (b) Experiment E: applying linear SVM using linear SVM with 5 samples (the 33^{rd}central sample from the target symbol and two symbols either side).

6 Summary and Conclusions

In this work we demonstrated the bit error rate can be reduced using machine-learning techniques. So far, the best results has been obtained for all distances using a linear SVM trained on data from the target symbol with either the symbol either side, or two symbols either side.

We are investigating that how many neighboring symbols should be used as inputs of our machine learning decoder. At the current stage, the target symbol with three, four, five symbols either side respectively are being investigated. Furthermore, since essentially, the sequence of pulses is time series, we shall apply embedding dimension [7] as a guide to find out a suitable number of neighbors.

We expect that our investigations with nonlinear SVM allow us to obtain further BER improvement along all distances. In addition, features extracted from the signal wave will be investigated in the future work as well. Moreover, we will investigate our methods on different kinds of modulation.

References

1. Bernstein, G., Rajagopalan, B., Saha, D.: Optical network control: architecture, protocols, and standards. Addison-Wesley Longman Publishing Co., Inc. (2003)
2. Metaxas, A., Redyuk, A., Sun, Y., Shafarenko, A., Davey, N., Adams, R.: Linear support vector machines for error correction in optical data transmission. In: Adaptive and Natural Computing Algorithms. Springer (2013) 438–445
3. Hunt, S., Sun, Y., Shafarenko, A., Adams, R., Davey, N., Slater, B., Bhamber, R., Boscolo, S., Turitsyn, S.K.: Adaptive electrical signal post-processing with varying representations in optical communication systems. In: Engineering Applications of Neural Networks. Springer (2009) 235–245
4. Agrawal, G.: Applications of nonlinear fiber optics. Academic press (2001)
5. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to information retrieval. Volume 1.
6. Chang, C.C., Lin, C.J.: Libsvm: A library for support vector machines. acm transactions on intelligent systems and technology, 2: 27: 1–27: 27, 2011. Software available at http://www. csie. ntu. edu. tw/cjlin/libsvm (2011)
7. Cao, L.: Practical method for determining the minimum embedding dimension of a scalar time series. Physica D: Nonlinear Phenomena **110**(1) (1997) 43–50

Error Correction over Optical Transmission

Weam M. Binjumah[1,2], Alexey Redyuk[3], Rod Adams[1], Neil Davey[1] and Yi Sun[1]
[1]*The School of Computer Science, University of Hertfordshire, Hatfield, U.K.*
[2]*The Community College, Taibah University, Madinah, Kingdom of Saudi Arabia*
[3]*Institute of Computational Technologies SB RAS, Novosibirsk, Russia*
{*weam.m.j , alexey.redyuk*}*@gmail.com*, {*R.G.Adams, n.davey, y.2.sun*}*@herts.ac.uk*

Keywords: Support Vector Machine (SVM), Machine Learning, Optical Signals, Coherent Optical Communications, Error Correction, Wavelet Transform.

Abstract: Reducing bit error rate and improving performance of modern coherent optical communication system is a significant issue. As the distance travelled by the information signal increases, bit error rate will degrade. Support Vector Machines are the most up to date machine learning method for error correction in optical transmission systems. Wavelet transform has been a popular method to signals processing. In this study, the properties of most used Haar and Daubechies wavelets are implemented for signals correction. Our results show that the bit error rate can be improved by using classification based on wavelet transforms (WT) and support vector machine (SVM).

1 INTRODUCTION

Improving the bit error rate (the number of bit errors divided by the total number of transmitted bits) in optical transmission systems is a crucial and challenging problem. There are many different causes of the transmitted signal degradation in optical communication systems, for instance optical losses, fiber nonlinearity, dispersive properties of the medium etc (Bernstein et al., 2003). Increasing the distance travelled by the optical pulses along long-haul fiber links also leads to an increase in the number of error bits. In optical telecommunications an information signal may be encoded by amplitude or the phase of the optical pulses. In this work, we consider phase encoding signals. Metaxas et al. demonstrates that linear Support Vector Machines (SVM) outperformed other trainable classifiers, such as using neural networks, for error correction in optical data transmission; besides that it is easier to build the hardware for an SVM in real time (Metaxas et al., 2013).

The purpose of signal decomposition is to extract the relevant information from the signal and reduce the level of interfering noise. The wavelet transform has become widespread in analyzing and processing signals. Wavelet signal processing can be applied to extract the underlying information of the signal (Rioul and Vetterli, 1991). For various kinds of signals, different kinds of wavelets can be selected. In this paper, we investigate whether wavelets can be used on the distorted optical signals to extract the reliable information of the original signals or not. Especially, we look into whether wavelets can deal with noise in phase and/or frequency of optical signals.

2 PROBLEM DOMAIN

Typical optical communication systems consist of three main components, see Figure 1: an optical transmitter (Tx in Figure 1) that converts the electrical signal into an optical signal, an optical fiber as the propagation medium of the optical signal and an optical receiver (Rx in Figure 1) that converts the received optical signal into an electrical signal again. During the transmission, the optical signals are exposed to many kinds of impairments such as attenuation, dispersion broadening and nonlinear distortion (Kanprachar, 1999).

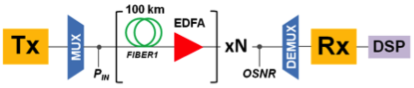


Figure 1: The optical fiber link configuration (Binjumah et al., 2015).

These impairments generate some error informa-

Binjumah, W., Redyuk, A., Adams, R., Davey, N. and Sun, Y.
Error Correction over Optical Transmission.
DOI: 10.5220/0006211402390248
In *Proceedings of the 6th International Conference on Pattern Recognition Applications and Methods (ICPRAM 2017)*, pages 239-248
ISBN: 978-989-758-222-6

tion bits at the receiver of the fiber link. Increasing the distance travelled by the signal leads to a loss in the quality of the signal and further bit error rate (BER) degradation (Binjumah et al., 2015). With the increase in speed currently achievable, the complexity of reduction in bit error rates increases. The high-speed and long distance data transmission in optical systems needs to be accompanied with as low bit error rate as possible (Metaxas et al., 2013). Therefore, the reduction of bit error rate in optical data transmission is a significant issue and is difficult to be achieved.

In earlier work, we investigated how a linear SVM classifier can be trained to automatically detect and correct bit errors. We took into consideration the most important neighbouring information, which can be used for training the linear SVM classifier, from each signal (Binjumah et al., 2015).

In this paper, a linear SVM classifier was used to classify the bits accurately, which reduces the error rate while transmitting the data across a specific distance. In addition, we investigate using wavelet transforms to remove noise from the signals prior to classification in order to improve the system performance.

3 METHODS

3.1 Representation of Signals using Wavelet Transforms

Wavelet transform is a mathematical tool that can be used for the extraction of information from a variety of data forms, such as images and audio signals (Lee and Lim, 2012). The theory of wavelet stands out amongst the present day scientific techniques in producing effective methods for the extraction of optimal data. It was mostly created by French scientists, according to (Plonka et al., 2013). This theory is currently utilized as an essential technique in specialized research in electronics, mechanical, computers, communications, medicine, biology, astronomy etc. In the field of image and signal handling, the fundamental uses of wavelet is to compress and de-noise them (Liu et al., 2013). In this work, we started with the simplest wavelet transforms: Haar wavelet transform. We have also used Daubechies wavelet transforms, which have been successfully applied in many engineering related works (Williams and Amaratunga, 1994).

3.1.1 Haar Wavelet Transforms

Haar wavelets have been used extensively as examples in teaching due to its simplicity. In fact, it is the simplest wavelet and has been a prototype for all other types of wavelet transforms. The Haar transform can be used for signal decomposition. It can be carried out at several levels. At the top level that is 1-level, a signal is transformed to two sub-signals, which are the approximation part and the details part. The approximation part is obtained by calculating the inner (scalar) product between the signal and the Haar scaling signals, while the details part is obtained by calculating the inner product between the signal and the Haar wavelets. Both Haar scaling signals and Haar wavelets are defined as basis functions, which can be seen in most of textbooks for wavelets (for example (Walker, 2008)). Once 1-level Haar transform is carried out, we can continue with the same process to work on the next level, where the signal is always the approximation part obtained from the previous/preceding level.

3.1.2 Daubechies Wavelet Transforms

The only difference between Daubechies and Haar transform is the definition of their scaling signals and the wavelets (Walker, 2008). All Daubechies wavelet transforms are similar to each other. The simplest type, that is Daub4 wavelet transform, is used in this work.

3.2 Linear SVM

Support Vector Machines (SVM) defines a class of machine learning algorithms and method used for classification, recognition and regression analysis. It is arguably the most successful method in machine learning. SVMs can be both linear and non-linear models. The SVM is a soft maximum margin classifier. Linear SVM has only one non-learnable parameter, which is the regularising cost parameter C (Smola and Schölkopf, 1998). This parameter allows the cost of mis-classification to be specified. The Linear SVM model is trained on a set of training data; the training data are linearly separable by a margin (supervised learning) and categorized into groups. Each input data sample is tested against the margin while the model tries to maximize the margin as much as possible (Wang et al., 2012).

3.3 The Threshold Method

The threshold method is based on using the middle point of the signal, where the signal (pulse) reaches the peak at the initial state. In this method, each signal is classified to one of four classes by measuring the phase value of its central sample. More details can be seen in section 4.3.

4 DESCRIPTION OF DATA

Optical signal data once it has been transmitted is subjected to a distortion in its amplitude, frequency and phase. As far as we can tell wavelet transformations have not been applied to data of this type, in particular when the data is to be subsequently analysed using an SVM. In order to fully quantify how effective wavelets might be with this distorted data we started by analysing how effective wavelet transformation would be on very simple data that had simulated noise added to its amplitude, frequency or to its phase. After that we then applied the wavelet transformation to our optical data. So the data we are analyzing is divided into two types, which have been transformed using wavelets and then used as input to the linear SVM classifier. The first type is sinusoidal waves/signals (simple data), and the second type is simulating optical signals (complex data).

4.1 Simple Data with Frequency and Amplitude Noise

Four classes A, B, C and D of Sinusoidal signals were generated with simulated noise added to its frequency via a Gaussian distribution based on a different mean frequency. Each class has a different mean value of frequency that is 10 (A), 15 (B), 20 (C) and 12 (D) respectively. All of them have the same standard deviation for the added 'noise', which is 2. Each class of data consists of 500 data points (each data point being a wave form/signal), and each wave consists of a vector of 640 y coordinates (samples). Each vector (wave) has a corresponding label. We then added Gaussian amplitude 'noise' with a mean value of 0 and standard deviation of 0.5 to the signal, this was added at each y coordinate of each generating signal.

4.2 Simple Data with Phase and Amplitude Noise

This time phase 'noise' was simulated, but no frequency 'noise'. Two classes of Sinusoidal signals were generated. Each class was initialized with different mean value of phase that is 0 radians (first class), and $\frac{\pi}{2}$ radians (second class). The Gaussian 'noise' has the same standard deviation in each class, which is 0.5. Again each class of data consists of 500 data points (wave forms/signals). Each wave has a corresponding label, and is represented as a 640 vector (samples). We then added Gaussian amplitude noisewith a mean value of 0 and standard deviation of 1, this was again added at each y coordinate of each generating signal. The signal was generated according to the following equation:

$$s = \sin(t+a) + AN \qquad (1)$$

where s is the signal, t is the index for the total number of time series, a is the phase value and AN is the amplitude noise of the signal.

4.3 Optical Signals (Simulated Data)

This part of data was generated using a simulating optical fibre link. It consists of 32,768 symbols per one WDM channel encoded by the quadrature phase shift keying (QPSK) modulation scheme. We consider a dual-polarization optical communication system (X and Y polarization). The simulation process was repeated 10 times with different random realizations of Amplified Spontaneous Emission (ASE) noise and input pseudorandom binary sequence (PRBS), each run generates 32,768 symbols.

The signal was detected at intervals of 1,000 km to a maximum distance 10,000 km. Each pulse was decoded into one of four symbols according to its phase. Signals that their phase values were bigger than $-\frac{\pi}{4}$ and smaller than $\frac{\pi}{4}$ will belong to the class 00. Signals that their phase values were bigger than $\frac{\pi}{4}$ and smaller than $\frac{3\pi}{4}$ will belong to the class 01. The class 11 have all signals that their phase values were bigger than $\frac{3\pi}{4}$ and smaller than π, or were smaller than $-\frac{3\pi}{4}$ and bigger than $-\pi$. And the last class 10 has all pulses that their phase values were bigger than $-\frac{3\pi}{4}$ and smaller than $-\frac{\pi}{4}$ (Binjumah et al., 2015). Each data point has a corresponding two-bit label for each run. Each run generates one data set. Each pulse is represented by 64 equally spaced phase samples. In this paper we focus on X-Polarization data at the distance 8,000 km. Furthermore, neighbouring information was used as input to the linear SVM classifier as well. The neighbouring information is using different numbers of samples from the symbol (signals) that will being decoded and different symbols either side.

5 EXPERIMENTAL SET-UP AND RESULTS

The aim of these experiments is to observe whether using wavelets can extract the original information from the distorted signals, and remove the noise that corrupts them. A linear SVM classifier was used to help decode the received signals with or without using wavelets. Linear SVM results that obtained using the noisy signals were compared to the results obtained using the extracted signals after using the wavelet transforms.

5.1 Experiments and Results using Simple Data

5.1.1 Simple Data with Frequency and Amplitude Noise

The aim of these experiments is to investigate whether using wavelet transforms can enable the SVM to better distinguish between the two sets of noisy data than without using the transforms. The data sets that were used in these experiments consist of a combination of two classes of data; they are AC, AB, AD and BD. For example, AC is a combination of the two classes of data A and C, and so on. Each pair of classes have different distances between their means and so represent a different level of difficulty when attempting to classify the noisy data. The 1,000 data points (500 from each class) was randomly selected to give 700 data points (signals) that were used to train the model, and the rest of the data (300 data points) were used as a test set.

Six tests were made: the signals with no added amplitude 'noise', without and with two types of wavelet transforms; the signals with added amplitude 'noise', without and with two types of wavelet transforms. The two wavelet transforms were: Haar and DB4 wavelet transforms, both at level 2. Then, the results were compared with each other to see if using wavelet transforms can help in improving the classification process or not.

Table 1 shows the linear SVM results for four different data sets with and without using wavelet transforms. As we see from the final column, the difference between the mean values of the frequency for class A and C is quite high (a difference of 10) and consequently the data could be partitioned with 98.67% accuracy. As a result, using the wavelet transforms on the test set AC did not give any improvement, with or without amplitude 'noise'. Essentially 1.33% of the waves were ambiguous even with no amplitude noise added. However, on the classes with closer means the data were more overlapping and the accuracy rates were further reduced. Significantly the use of wavelets did not have any effect on the data with just frequency noise in any of the tests. However, once the Amplitude noise was added the use of wavelets did improve the accuracy back towards the values obtained with the Frequency noise only version. For instance with classes A and B the wavelet transformed waves nearly brought the fully noisy wave performance up to that of the Frequency only noisy wave (from 84.33% to 90.67%, which is very close to the 91% Frequency only-noisy version), this being the best result obtained.

5.1.2 Simple Data with Phase and Amplitude Noise

The aim of these kind of experiments is to investigate whether using wavelet transforms can improve the signals that have phase noise or not. The data set used herein consists of 1,000 data points/signals, and 640 samples for each data point. Half of the data set has the phase value of zero, and the other half has phase value of 90 degrees. In this experiment, a linear SVM was applied on the data set for classification of the received signals. 600 data points (signals) were used to train the model as a training set, and the rest of the data (400 data points) were used as a test set. Tests that were made are three types: the signals with no amplitude noise, noisy signals, and signals after using wavelet transforms (extracted signals).

Here we also tried to normalize the extracted signals to see if that would help in improving the classification process or not. The average of difference between the original and extracted signal got bigger after increasing the level of wavelet transforms. In the normalization process, the range of the extracted signals is re-scaled to be between -1 and 1 as the original signals. Figure 2 shows two original signals without any noise from two classes using solid lines (Red for phase of 0 and blue for phase of 90 degrees), and ten signals of each class after adding random phase and amplitude noise. Figure 3, shows ten extracted signals, using Haar wavelet transform at level 2 without normalisation, and Figure 4 shows the same with normalization. As we can see from the Figures, the signal samples become between the range [-1,1] after the normalization.

In this section, Haar wavelet transform at different levels from 1 to 5, and db4 wavelet transform at level 2 were implemented. Then, the linear SVM classifier was applied using the extracted signals. The classification process was done using two types of input. The first type using the whole samples (i.e the vector of all 640 points), and the second type using the central sample (the middle point of the wave) of the extracted signals. Results were obtained without normalisation and with normalisation.

1) Linear SVM Results using Extracted Signals without Normalization

Table 1 presents the accuracy rate of prediction using linear SVM classifier on the non-normalized extracted signals. Table 1 (A), shows the linear SVM re-

Table 1: Linear SVM results on 4 different data sets.

Group of Data	Type of data	Type of (WT)	Level of (WT)	Accuracy rate%
AC (10)	F-noise only	-	-	98.67%
		Haar wavelet	2	98.67%
		db4 wavelet	2	98.67%
	F + A noise	-	-	98.67%
		Haar wavelet	2	98.67%
		db4 wavelet	2	98.67%
AB **(5)**	F-noise only	**-**	**-**	91%
		Haar wavelet	2	91%
		db4 wavelet	2	91%
	F + A noise	**-**	**-**	**84.33**%
		Haar wavelet	**2**	**90**%
		db4 wavelet	**2**	**90.67**%
AD (2)	F-noise only	-	-	69%
		Haar wavelet	2	69%
		db4 wavelet	2	69%
	F + A noise	-	-	65.33%
		Haar wavelet	2	67.67%
		db4 wavelet	2	67%
BD (3)	F-noise only	-	-	79%
		Haar wavelet	2	79%
		db4 wavelet	2	79%
	F + A noise	-	-	73%
		Haar wavelet	2	77.33%
		db4 wavelet	2	76.67%

Note: The number in the brackets underneath the data is the difference between the means of the frequency. F-noise means Frequency noise only. F + A noise means both Frequency and Amplitude noise were used. (-) denotes corresponding results are obtained without applying wavelets.

sults using the whole samples of the non-normalized extracted signals. Unfortunately, the results in Table 1 (A) did not show a noticeable improvement, where the accuracy rate before using wavelet transform (using noisy signals) is 92.5%, and after using wavelet transform is improved to 92.75% using Haar wavelets at levels 1, 2 and 5. Table 1 (B) demonstrated the linear SVM results using just the central sample of the non-normalized extracted signals. With less input information these values are lower than those in Table 1 (A). Interestingly, the best result obtained was using db4 wavelet transform at level 2, which is 93% from the noisy signal level of 91.75%. Whereas Haar wavelet transform did not show any improvement in the result.

2) Linear SVM Results using Normalized Extracted Signals

Table 2 presents the accuracy rate of prediction using linear SVM classifier on the normalized extracted signals. Table 2 (A), shows the linear SVM results using the whole samples of the extracted normalized signals. Again, unfortunately, the linear SVM results using the whole samples did not show any improvement, where the accuracy rate was only improved from 92.5% to 92.75% after using wavelet transform. Table 2 (B) show the linear SVM results using the central sample of the extracted normalized signals. Here the wavelet transformations did have an effect, perhaps representing that they had more work to do when only using the central sample. The best result was obtained using DB4 wavelet transform at level 2, where the accuracy rate is 93.75% (from 91.75%). Comparing Tables 1 and 2 we see that generally, using the normalization improved the results, especially when using the central sample of the signals as inputs. However the overall results show the difficulty that wavelets have with phase distorted data.

5.2 Experiments and Results using Optical Signals (Complex Data)

Finally we experiment on the full optical data. The purpose of this experiment is to figure out whether using wavelet transforms can process the distorted optical signals or not. In this experiment, a linear SVM was implemented using lots of different input vectors: just the central sample, the whole set of samples from

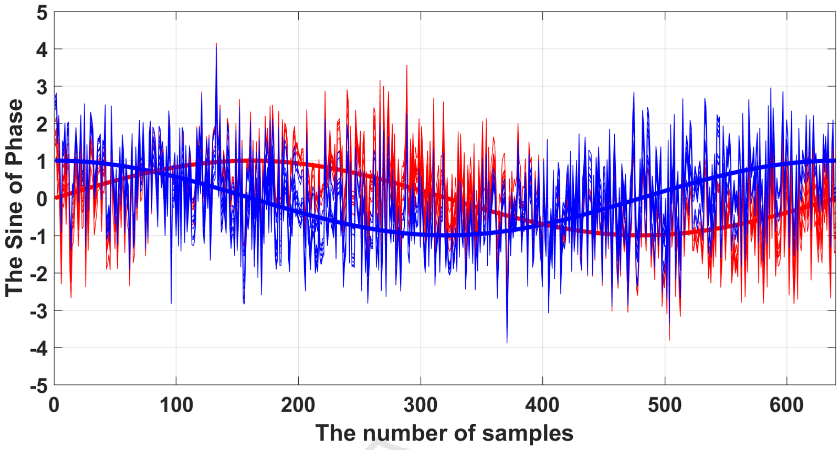


Figure 2: Ten Sinusoid signals with phase and amplitude noise compared with non-noisy signals (solid lines). Blue signals has phase of 90 and red signals has phase of 0 degrees.

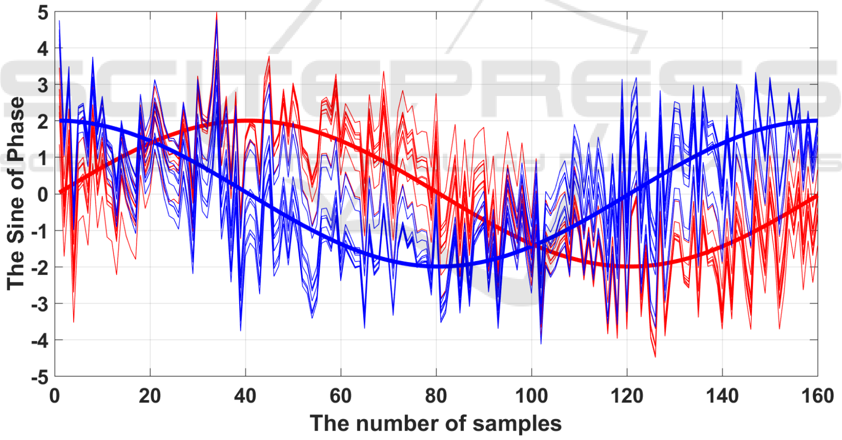


Figure 3: Ten extracted signals using Haar wavelet transform, level 2 (Approximation part). Blue signals has phase of 90 and red signals has phase of 0 degrees.

the wave (all 64 values) and neighbouring information from waves before and after the wave being classified. A selection of different transformations were tried, from none at all (original signal) to Haar level 1 and 2 and db4 level 2 wavelets. Regarding using the neighbouring information, we focused on using 7 central samples from 7 adjacent symbols (from the target symbol and three symbols either sides). We have found that using 7 central samples from 7 neighbouring symbols gave the best linear SVM results when we have used neighbouring information previously. In this experiment, $\frac{2}{3}$ of the symbols/signals were used to train the linear SVM model, and the rest of the data (a third of the symbols) was used as a test set.

Table 3 shows the linear SVM results using the optical signals at the distance 8,000 km, with and without using wavelet transform. These results were

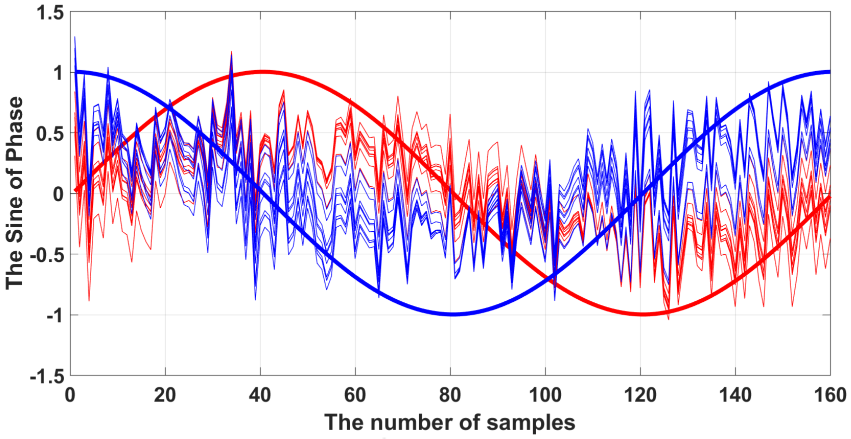


Figure 4: Ten extracted normalized signals using Haar wavelet transform, level 2 (Approximation part). Blue signals has phase of 90 and red signals has phase of 0 degrees.

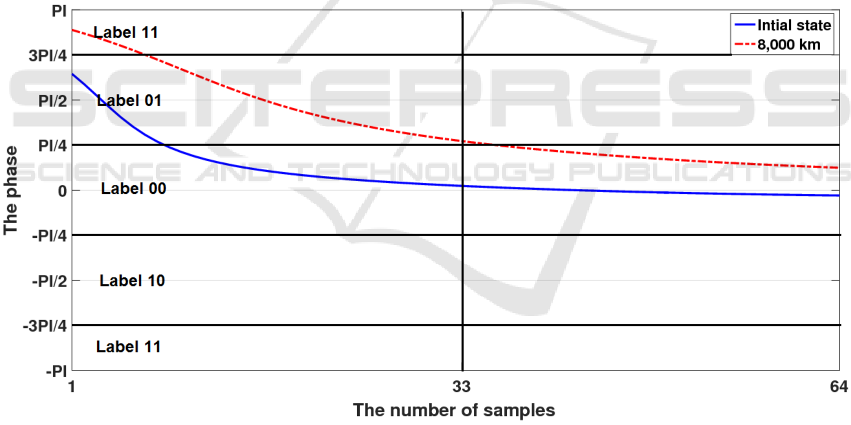


Figure 5: An optical signal has been classified incorrectly using both linear SVM using central samples from 7 symbols, Haar transforms at level 2, and the threshold method (the first data set).

compared with the results obtained by measuring the phase of the mid-point of the signal (threshold method) which is the current hardware implemented method. The Table presents the symbol accuracy rate (SAR%), number of bit errors (NBE) and bit error rate (BER%), which are an average over ten data sets. As we can see from Table 3, the results using samples from 7 consecutive symbols (using 3 either side of the target symbol) were best, even though they only used the central value of each of the 7 waves. This is the result we have obtained before. Using a linear SVM using the extracted signals obtained from DB4 wavelet transform did not improve the classification process. The best result we have got so far is the linear SVM result using 7 central samples from 7 neighbouring extracted signals, obtained from Haar wavelet transform, level 2 which is a 1.68 BER.

Figures 5, 6 and 7 show some examples of optical signals at the initial state (blue solid line), and after 8,000 km (red dotted line). The mid-point of the sig-

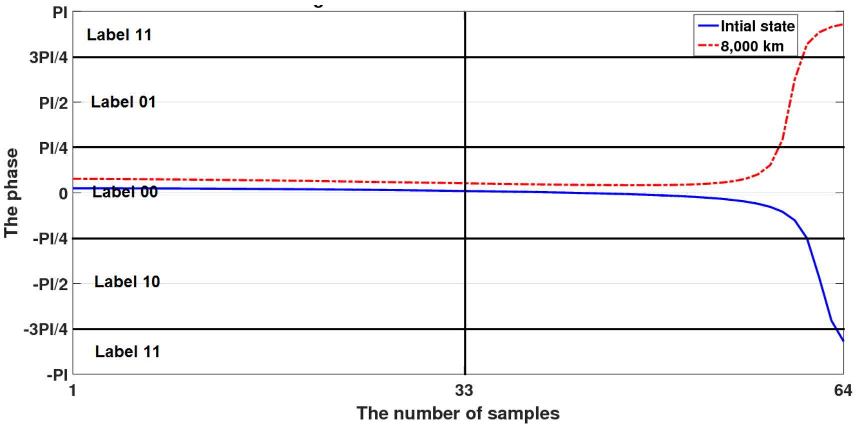


Figure 6: An optical signal has been classified correctly using both linear SVMusing central samples from 7 symbols, Haar transforms at level 2, and the threshold method (the first data set).

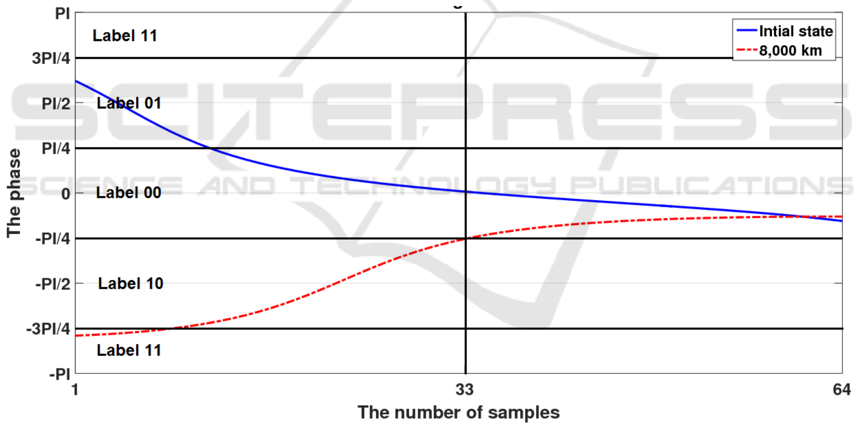


Figure 7: An optical signal has been classified correctly using linear SVM using central samples from 7 symbols, Haar transforms at level 2, and misclassified using the threshold method (the first data set).

nal is the 33rd sample, where the phase is measured, because that represents the highest power level. These figures were selected from the best linear result using the Haar wavelet at level 2, from the target signal and three signals either side. From Figure 5, we can see an optical signal that has been mis-classified as class 01, using both linear SVM and the threshold method, where it belongs to class 00. Figure 6 presents an optical signal that has been detected correctly as class 00, using both linear SVM and the threshold method. Figure 7 shows an optical signal that has been detected correctly using linear SVM, but incorrectly using the threshold method. As we can see, the signal should belong to the class 00, but was mis-classified as class 10 by the threshold method, at the distance 8,000 km. From our observation, we can say that using linear SVMs based on wavelets transformations can ensure some types of distorted signals be classified correctly (for example, Figure 7).

Table 4: The linear SVM results using optical signals before and after using wavelet transforms at the distance 8,000 km, compared with the threshold method result.

Method	Number of samples	Signal types	SAR %	NBE	BER %
Threshold	Central sample	Original signal	96.3± 0.15	403.3± 16.55	1.87 ± 0.08
Linear SVM	Central sample	Original signal	96.29 ± 0.16	403.6 ± 18.001	1.87 ± 0.08
Linear SVM	Central sample	Haar level 1	96.36 ± 0.16	396.2 ± 17.37	1.84 ± 0.08
Linear SVM	Central sample	Haar level 2	96.41 ± 0.16	390.9 ± 17.15	1.82 ± 0.09
Linear SVM	Central sample	db4 level 2	93.95 ± 0.47	661.9 ± 50.71	3.07 ± 0.24
Linear SVM	Whole samples	Original signal	96.44 ± 0.14	387.3 ± 15.85	1.8 ± 0.07
Linear SVM	Whole samples	Haar level 1	96.44 ± 0.13	387.5 ± 14.22	1.8 ± 0.07
Linear SVM	Whole samples	Haar level 2	96.45 ± 0.13	386.2 ± 13.86	1.79 ± 0.07
Linear SVM	Central samples from 7 symbols	Original signal	96.6 ± 0.1	370.2 ± 11.65	1.72 ± 0.05
Linear SVM	**Central samples from 7 symbols**	**Haar level 2**	**96.67 ± 0.11**	**362.4 ± 13.21**	**1.68 ± 0.06**
Linear SVM	Central samples from 7 symbols	db4 level 2	95.75 ± 0.22	465.3 ± 23.45	2.16 ± 0.12

Table 2: A comparison between linear SVM results using noisy signals and the extracted signals.

A) The whole samples of the extracted signal were used as input to the linear SVM classifier.

Data set	Type of (WT)	(WT) level	Accuracy rate %
P-noise	-	-	92.5 %
	Haar	2	91.5 %
P + A noise	-	-	92.5 %
	Haar	**1**	**92.75** %
	Haar	2	92.75 %
	Haar	3	92.5 %
	Haar	4	92.5 %
	Haar	**5**	**92.75** %
	Db4	2	92.25 %

B) The central sample of the extracted signals was used as input to the linear SVM classifier.

Data set	Type of (WT)	(WT) level	Accuracy rate %
P-noise	-	-	91.75 %
	Db4	2	91.75 %
P + A noise	-	-	91.75%
	Haar	1	91.75%
	Haar	2	91%
	Haar	3	91%
	Haar	4	90%
	Haar	5	87.25%
	Db4	**2**	**93**%

Note: P-noise means Phase noise only. P + A noise means both Phase and Amplitude noise were used. (-) denotes corresponding results are obtained without applying wavelets.

Table 3: A comparison between linear SVM results using noisy signals and the extracted normalized signals.

A) The whole samples of the extracted normalized signal were used as input to the linear SVM classifier.

Data set	Type of (WT)	(WT)	Accuracy rate %
P-noise	-	-	92.5 %
	Haar	3	92.5 %
P + A noise	-	-	92.5%
	Haar	1	92.5%
	Haar	2	92.5%
	Haar	**3**	**92.75**%
	Haar	4	92.5%
	Haar	5	92.5%
	Db4	**2**	**92.75**%

B) The central sample of the extracted normalized signal were used as input to the linear SVM classifier.

Data set	Type of (WT)	(WT)	Accuracy rate %
P-noise	-	-	91.75 %
	Db4	2	91.75 %
P + A noise	-	-	91.75%
	Haar	1	93.5%
	Haar	2	93.5%
	Haar	3	90.5%
	Haar	4	93%
	Haar	5	88%
	Db4	**2**	**93.75**%

Note: P-noise means Phase noise only. P + A noise means both Phase and Amplitude noise were used. (-) denotes corresponding results are obtained without applying wavelets.

6 DISCUSSION AND CONCLUSION

In this work, we have demonstrated that the bit error rate can be improved by using classification based on wavelet transforms (WT) and support vector machine (SVM). From the results obtained using the simple data with frequency noise in Table 1, we can see that the best linear SVM result was when we used the data set (AB) after using wavelet transform level 2. Regarding the results obtained using the simple data with phase noise in Table 2, the best linear SVM result was when we used the central sample of the extracted normalized signals resulted from DB4 wavelet transform at level 2, which was 93.75%. Wavelets were more beneficial with the frequency distorted data than with the phase distorted data. However, overall the use of wavelet transforms was disappointing.

The second part of the results were obtained using wavelets on the optical signals at a distance of 8,000 km. The best result was when using a linear SVM trained on the extracted data (using Haar wavelet level 2) from the target symbol and three symbols either side. So wavelet transforms did have a small effect on the accuracy, and in this work small effects can be worth a lot. In particular using the combination of neighbourhood information and wavelets gave much better results than using the threshold method, see Table 3. This is crucial since Bit Error Rates less than 2 are required for optical data and the further we can drive this rate down the better. Furthermore, this work shows that wavelet transforms can help a little with the noise on both frequency and phase since optical data has both.

In this paper, our initial work on wavelets has been presented; different types of the wavelets will be investigated in the future.

REFERENCES

Bernstein, G., Rajagopalan, B., and Saha, D. (2003). *Optical network control: architecture, protocols, and standards*. Addison-Wesley Longman Publishing Co., Inc.

Binjumah, W., Redyuk, A., Davey, N., Adams, R., and Sun, Y. (2015). Reducing bit error rate of optical data transmission with neighboring symbol information using a linear support vector machine. In *Proceedings of the ECMLPKDD 2015 Doctoral Consortium(2015)*, pages 67–74. Aalto University.

Kanprachar, S. (1999). Modeling and analysis of the effects of impairments in fiber optic links.

Lee, S.-H. and Lim, J. S. (2012). Parkinson's disease classification using gait characteristics and wavelet-based feature extraction. *Expert Systems with Applications*, 39(8):7338–7344.

Liu, Z., Cao, H., Chen, X., He, Z., and Shen, Z. (2013). Multi-fault classification based on wavelet svm with pso algorithm to analyze vibration signals from rolling element bearings. *Neurocomputing*, 99:399–410.

Metaxas, A., Redyuk, A., Sun, Y., Shafarenko, A., Davey, N., and Adams, R. (2013). Linear support vector machines for error correction in optical data transmission. In *International Conference on Adaptive and Natural Computing Algorithms*, pages 438–445. Springer.

Plonka, G., Iske, A., and Tenorth, S. (2013). Optimal representation of piecewise hölder smooth bivariate functions by the easy path wavelet transform. *Journal of Approximation Theory*, 176:42–67.

Rioul, O. and Vetterli, M. (1991). Wavelets and signal processing. *IEEE signal processing magazine*, 8(LCAV-ARTICLE-1991-005):14–38.

Smola, A. J. and Schölkopf, B. (1998). *Learning with kernels*. Citeseer.

Walker, J. S. (2008). *A primer on wavelets and their scientific applications*. CRC press.

Wang, H., Guo, J., Wang, T., Zhang, Q., and Shao, J. (2012). Physical layer design for free space optical communication. In *Control Engineering and Communication Technology (ICCECT), 2012 International Conference on*, pages 978–981. IEEE.

Williams, J. R. and Amaratunga, K. (1994). Introduction to wavelets in engineering. *International Journal for Numerical Methods in Engineering*, 37(14):2365–2388.

ESANN 2017 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence
and Machine Learning. Bruges (Belgium), 26-28 April 2017, i6doc.com publ., ISBN 978-287587039-1.
Available from http://www.i6doc.com/en/.

Investigating Optical Transmission Error Correction using Wavelet Transforms

Weam M. Binjumah[1,2] , Alexey Redyuk[3] , Rod Adams[1], Neil Davey[1], and Yi Sun[1]

1- The School of Computer Science
University of Hertfordshire, Hatfield, AL10 9AB - UK,
[R.G.Adams, n.davey, y.2.sun]@herts.ac.uk.

2- The Community College - Department of Computer Science
Taibah University, Madinah - KSA,
weam.m.j@gmail.com.

3- Institute of Computational Technologies SB RAS
Novosibirsk, 6 Acad. Lavrentiev avenue, 630090 - Russia,
alexey.redyuk@gmail.com.

Abstract. Reducing bit error rate and improving performance of modern coherent optical communication system is a significant issue. As the distance travelled by the information signal increases, bit error rate will degrade. Support Vector Machines are the most up to date machine learning method for error correction in optical transmission systems. Wavelet transform has been a popular method to signals processing. In this study, results show that the bit error rate can be improved by using classification based on wavelet transforms (WT) and support vector machine (SVM).

1 Introduction

Improving the bit error rate in optical transmission systems is a crucial and challenging problem. There are many different causes of the transmitted signal degradation in optical communication systems [1]. Linear Support Vector Machines (SVM) outperformed other trainable classifiers for error correction in optical data transmission; besides that it is easier to be build in the hardware in real time [2]. The wavelet transform has become widespread in analyzing and processing signals. It can be used for signal decomposition to help extract the relevant information from the signal and reduce the level of interfering noise [3]. In this paper, we investigate whether wavelets can be used on the distorted optical signals to extract the reliable information of the original signals or not. Especially, we look into whether wavelets can deal with noise in phase and/or frequency of optical signals.

2 Problem Domain

During the transmission, the optical signals are exposed to many kinds of impairments such as attenuation, dispersion broadening and nonlinear distortion [4]. These impairments generate some error information bits at the receiver of the fiber link. Increasing the distance travelled by the signal leads to a loss in the quality of the signal and further bit error rate (BER) degradation [5]. With the increase in speed currently achievable, the complexity of reduction in bit error rates increases. The high-speed and long distance data transmission in optical systems needs to be accompanied with as low bit error rate as possible [2]. Therefore, the reduction of bit error rate in optical data transmission is a significant issue and is difficult to be achieved. In earlier work, we investigated how a linear SVM classifier can be trained to automatically detect and correct bit errors. We took into consideration the most important neighbouring information, which can be used for training the linear SVM classifier, from each signal [5]. In this paper, we investigate using wavelet transforms to remove noise from the signals prior to classification.

3 Method

In this work, the classifying and processing of the signals was based on two methods: linear SVM and wavelet transforms (WT). Linear SVM is a soft maximum margin classifier, which has only one non-learnable/cost parameter C [6]. Wavelet transforms are a mathematical tool that can be used for the extraction of information from a variety of data forms [7]. In the field of image and signal handling, wavelet is used to compress and de-noise them [8]. In this work, we started with the simplest wavelet transform: Haar wavelet transform, which can be used for signal decomposition [9]. We have also investigated other types of wavelet; for example, Daubechies. However, the results we have obtained are very similar.

4 Description of data

Optical signal data once it has been transmitted is subjected to a distortion in its amplitude, frequency and phase. As far as we can tell wavelet transformations have not been applied to data of this type, in particular when the data is to be subsequently analysed using an SVM. In order to fully quantify how effective wavelets might be with this distorted data we started by analysing how effective wavelet transformation would be on very simple data that had simulated noise added to its amplitude, frequency or to its phase. After that we then applied the wavelet transformation to our optical data.

4.1 Simple data with frequency noise

Four classes A, B, C and D of Sinusoidal signals were generated with simulated noise added to its frequency. This frequency noise was added via a Gaussian distribution based on a different mean frequency, which is 10 (A), 15 (B), 20 (C) and 12 (D) respectively. Each class of data consists of 500 signals, and each signal consists of a vector of 640 y coordinates (samples). Each vector (wave) has a corresponding class label. Gaussian amplitude 'noise' was added at each y coordinate of each generating signal.

4.2 Simple data with phase noise

Two classes of Sinusoidal signals were initialized with simulated phase noise via a Gaussian distribution based on a different mean values of phase; that is 0 radians (first class), and $\frac{\pi}{2}$ radians (second class). Again each class of data consists of 500 signals. Each signals has a corresponding class label, and is represented as a 640 vector (samples). Gaussian amplitude noise was again added at each y coordinate of each generating signal.

4.3 Optical signals (simulated data)

The data was generated using a simulating optical fiber link. It consists of 32,768 signals per one WDM channel encoded by the quadrature phase shift keying (QPSK) modulation scheme. The signal was detected at distance 8,000 km. Each pulse was decoded into one of four symbols according to its phase [5]. Each signal has a corresponding two-bit label. Each pulse is represented by 64 equally spaced phase samples. Furthermore, neighbouring information was used as input to the linear SVM classifier as well. The neighbouring information is using different numbers of samples from the symbol (signal) that will being decoded and different symbols either side.

5 Experimental set-up and results

The aim of these experiments is to observe whether using wavelets can extract the original information from the distorted signals, and remove the noise that corrupts them. A linear SVM classifier was used to help decode the received signals with or without using wavelets.

5.1 Experiments and results using simple data

5.1.1 Simple data with frequency noise

In these experiments, we investigate whether using wavelet transforms can enable the SVM to better distinguish between the two sets of noisy data than without using the transforms. The linear SVM was implemented using the whole samples of the signal as input vector. The data sets that were used in these experiments consist of a combination of two classes of data; they are AC, AB and AD. Each pair of classes have different distances between their means and so represent a different level of difficulty when attempting to classify the noisy data. The 1,000 data points (500 from each class) was randomly selected to give 700 data points (signals) that were used to train the model, and the rest of the data (300 data points) were used as a test set. The wavelet transforms used were Haar at level 2, although others have been tried giving similar results. Table 1 shows the linear SVM results for three different data sets with and without using wavelet transforms. As expected we can see from the final column that as the difference between the mean values of the frequency decreases so does the accuracy rate. However, the use of wavelets did improve the accuracy for the classes with a closer mean. For instance with classes A and B the wavelet transformed waves improved accuracy from 84.33% to 90%. This is the best result obtained.

Data set	Type of WT	Level of WT	Accuracy rate %
AC (10)	-	-	98.67 %
	Haar wavelet	2	98.67 %
AB (5)	**-**	**-**	**84.33** %
	Haar wavelet	**2**	**90** %
AD (2)	-	-	65.33 %
	Haar wavelet	2	67.67%

Note: The number in the brackets is the difference between the means of the frequency. The data (noisy signals) has both Frequency and Amplitude noise. (-) denotes corresponding results are obtained without applying wavelets.

Table 1: Linear SVM results on 3 different data sets of noisy signals.

5.1.2 Simple data with phase noise

In these experiments, we investigate whether using wavelet transforms can improve the signals that have phase noise or not. The data set used herein consists of 1,000 data points/signals. Half of the data set has the phase value of zero, and the other half has phase value of 90 degrees. 600 data points (signals) were used to train the model as a training set, and the rest of the data (400 data points) were used as a test set. Here we also normalized the extracted signals to values between [-1,1] in order to help in improving the classification process. In this section, Haar wavelet transform at level 2 is reported although others were used with similar results. The classification process was done using two types of input: the whole samples (i.e the vector of all 640 points), and the central sample (the middle point of the wave) of the extracted signals. Table 2 presents the accuracy rate of prediction using linear SVM classifier on the noisy and the normalized extracted signals. As we can see, the linear SVM results using the whole samples did not show any improvement. But, the wavelet transformations using the central sample did have an effect. interestingly the best result obtained was 93.5% (from 91.75%) which was better than that obtained using the whole samples.

Samples No.	Type of (WT)	Level of (WT)	Accuracy rate %
Whole samples	-	-	92.5 %
	Haar wavelet	2	92.5 %
Central sample	**-**	**-**	**91.75** %
	Haar wavelet	**2**	**93.5** %

Note: The data (noisy signals) has both Phase and Amplitude noise. (-) denotes corresponding results are obtained without applying wavelets.

Table 2: A comparison between linear SVM results using noisy signals and the extracted normalized signals.

5.2 Experiments and results using optical signals (complex data)

The moderate success found using wavelets on the simple data meant it was worth experimenting on the full optical data. In this experiment, a linear SVM was implemented using lots of different input vectors. A selection of different transformations were tried, from none at all (original signal) to Haar levels (1 and 2), and other wavelets. Again for space reasons we only give the results for Haar 2, although the other results are very similar or worse. Regarding using the neighbouring information, we focused on using 7 central samples from 7 adjacent symbols (from the target symbol and three symbols either sides). We have found that using 7 central samples from 7 neighbouring symbols gave the best linear SVM results when we have used neighbouring information previously. In this experiment, $\frac{2}{3}$ of the symbols/signals were used to train the linear SVM model, and the rest of the data (a third of the symbols) was used as a test set.

Method	No. of samples	Signal types	SAR %	BER %
Threshold	Central sample	Original signal	96.3±0.15	1.87±0.08
Linear SVM	Central sample	Original signal	96.29±0.16	1.87±0.08
		Haar (level 2)	96.41±0.16	1.82±0.09
	Whole samples	Original signal	96.44±0.14	1.8±0.07
		Haar (level 2)	96.45±0.13	1.79±0.07
	Neighbouring Info.	Original signal	96.6±0.1	1.72±0.05
		Haar (level 2)	**96.67±0.11**	**1.68±0.06**

Note: Neighbouring information means that using 7 central samples from 7 adjacent symbols.

Table 3: The linear SVM results using optical signals before and after using wavelet transforms at the distance 8,000 km, compared with the threshold method result.

Table 3 shows the linear SVM results using the optical signals at a distance of 8,000 km, with and without using wavelet transform. These results were compared with the results obtained by measuring the phase of the mid-point of the signal (threshold method) which is the current hardware implemented method. The Table presents the symbol accuracy rate (SAR%) and bit error rate (BER%), which are an average over ten data sets. As we can see from Table 3, the results using samples from 7 consecutive symbols (using 3 either side of the target symbol) were best, even though they only used the central value of each of the 7 waves. The best result we have got so far is the linear SVM result using 7 central samples from 7 neighbouring extracted signals, obtained from Haar wavelet transform, level 2, which is a 1.68 BER.

6 Discussion and conclusion

In this work, we have demonstrated that the bit error rate can be improved by using classification based on wavelet transforms (WT) and support vector machine (SVM). From the results obtained using the simple data with frequency

ESANN 2017 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence
and Machine Learning. Bruges (Belgium), 26-28 April 2017, i6doc.com publ., ISBN 978-287587039-1.
Available from http://www.i6doc.com/en/.

noise in Table 1, we can see that the best result was when we used the data set (AB) after using Haar wavelet transform at level 2. Regarding the results obtained using the simple data with phase noise in Table 2, the best result was when we used the central sample of the extracted normalized signals, which was 93.5%. Wavelets were more beneficial with the frequency distorted data than with the phase distorted data. However, overall the use of wavelet transforms was a bit disappointing. The second part of the results were obtained using wavelets on the optical signals at a distance of 8,000 km. The best result was using a linear SVM trained on the extracted data (using Haar wavelet level 2) from the target symbol and three symbols either side. So wavelet transforms did have a small effect on the accuracy, and in this work small effects can be worth a lot. In particular using the combination of neighbourhood information and wavelets gave much better results than using the threshold method, see Table 3. This is crucial since Bit Error Rates less than 2 are required for optical data and the further we can drive this rate down the better. Furthermore, this work shows that wavelet transforms can help a little with the noise on both frequency and phase since optical data has both.

References

[1] Greg Bernstein, Bala Rajagopalan, and Debanjan Saha. *Optical network control: architecture, protocols, and standards*. Addison-Wesley Longman Publishing Co., Inc., 2003.

[2] Alex Metaxas, Alexei Redyuk, Yi Sun, Alex Shafarenko, Neil Davey, and Rod Adams. Linear support vector machines for error correction in optical data transmission. In *International Conference on Adaptive and Natural Computing Algorithms*, pages 438–445. Springer, 2013.

[3] Olivier Rioul and Martin Vetterli. Wavelets and signal processing. *IEEE signal processing magazine*, 8(LCAV-ARTICLE-1991-005):14–38, 1991.

[4] Surachet Kanprachar. Modeling and analysis of the effects of impairments in fiber optic links. 1999.

[5] Weam Binjumah, Alexey Redyuk, Neil Davey, Rod Adams, and Yi Sun. Reducing bit error rate of optical data transmission with neighboring symbol information using a linear support vector machine. In *Proceedings of the ECMLPKDD 2015 Doctoral Consortium(2015)*, pages 67–74. Aalto University, 2015.

[6] Alex J Smola and Bernhard Schölkopf. *Learning with kernels*. Citeseer, 1998.

[7] Sang-Hong Lee and Joon S Lim. Parkinson's disease classification using gait characteristics and wavelet-based feature extraction. *Expert Systems with Applications*, 39(8):7338–7344, 2012.

[8] Zhiwen Liu, Hongrui Cao, Xuefeng Chen, Zhengjia He, and Zhongjie Shen. Multi-fault classification based on wavelet svm with pso algorithm to analyze vibration signals from rolling element bearings. *Neurocomputing*, 99:399–410, 2013.

[9] James S Walker. *A primer on wavelets and their scientific applications*. CRC press, 2008.

Bibliography

AAC (n.d.). Understanding i/q signals and quadrature modulation. `https://www.allaboutcircuits.com/textbook/radio-frequency-analysis-design/radio-frequency-demodulation/understanding-i-q-signals-and-quadrature-modulation/`. Accessed: 2018-08-24.

Aaruni, V., Harsha, A., and Joseph, L. A. (2015). Classification of eeg signals using fractional calculus and wavelet support vector machine. In *Signal Processing, Informatics, Communication and Energy Systems (SPICES), 2015 IEEE International Conference on*, pages 1–5. IEEE.

Afifi, S. M., GholamHosseini, H., and Poopak, S. (2015). Hardware implementations of svm on fpga: A state-of-the-art review of current practice.

Agrawal, G. P. (1997). Optical receivers. *Fiber-Optic Communication Systems*, pages 133–182.

Agrawal, G. P. (2018). Fiber-optic communication systems third edition.

Alegria, O. C., Valtierra-Rodriguez, M., Amezquita-Sanchez, J. P., Millan-Almaraz, J. R., Rodriguez, L. M., Moctezuma, A. M., Dominguez-Gonzalez, A., and Cruz-Abeyro, J. A. (2015). Empirical wavelet transform-based detection of anomalies in ulf geomagnetic signals associated to seismic events with a fuzzy logic-based system for automatic diagnosis. In *Wavelet transform and some of its real-world applications*. InTech. Available from: https://www.intechopen.com/books/wavelet-transform-and-some-of-its-real-world-applications/empirical-wavelet-transform-based-detection-of-anomalies-in-ulf-geomagnetic-signals-associated-to-se. Accessed: 2019-01-01.

Alfiad, M., van den Borne, D., Wuth, T., Kuschnerov, M., Lankl, B., Weiske, C., de Man, E., Napoli, A., and de Waardt, H. (2008). 111-gb/s polmux-rz-dqpsk transmission over 1140 km of ssmf with 10.7-gb/s nrz-ook neighbours. In *Optical Communication, 2008. ECOC 2008. 34th European Conference on*, pages 1–2. IEEE.

Anantrasirichai, N., Achim, A., Morgan, J. E., Erchova, I., and Nicholson, L. (2013). Svm-based texture classification in optical coherence tomography. In *Biomedical Imaging (ISBI), 2013 IEEE 10th International Symposium on*, pages 1332–1335. IEEE.

Antoniades, N. N., Ellinas, G., and Roudas, I. (2011). *WDM Systems and Networks: Modeling, Simulation, Design and Engineering.* Springer Science & Business Media.

Aoudia, F. A. and Hoydis, J. (2018). End-to-end learning of communications systems without a channel model. *arXiv preprint arXiv:1804.02276.*

BBC (2014a). Total internal reflection. `http://www.bbc.co.uk/schools/gcsebitesize/science/ocr_gateway_pre_2011/energy_home/3_infrared_signals4.shtml`. Accessed: 2018-07-05.

BBC (2014b). Total internal reflection. `http://www.bbc.co.uk/schools/gcsebitesize/science/ocr_gateway_pre_2011/energy_home/3_infrared_signals3.shtml`. Accessed: 2018-07-05.

Belden Inc. (2018). Cable basics: Fiber optic cable. `http://www.beldencables-emea.com/en/products/cable_basics/fiber-optic-cable/index.phtml`. Accessed: 2018-12-10.

Bernstein, G., Rajagopalan, B., and Saha, D. (2003). *Optical network control: architecture, protocols, and standards.* Addison-Wesley Longman Publishing Co., Inc.

Binjumah, W., Redyuk, A., Davey, N., Adams, R., and Sun, Y. (2015a). Reducing bit error rate of optical data transmission with neighboring symbol information using a linear support vector machine. In *Proceedings of the ECMLPKDD 2015 Doctoral Consortium(2015)*, pages 67–74. Aalto University.

Binjumah, W., Redyuk, A., Davey, N., Adams, R., and Sun, Y. (2015b). Reducing bit error rate of optical data transmission with neighboring symbol information using a linear support vector machine. *Proceedings of the ECMLPKDD*, pages 67–74.

Binjumah, W. M., Redyuk, A., Adams, R., Davey, N., and Sun, Y. (2017a). Error correction over optical transmission. In *Proceedings of the 6th International Conference on Pattern Recognition Applications and Methods - Volume 1: ICPRAM,*, pages 239–248. INSTICC, SciTePress.

Binjumah, W. M., Redyuk, A., Adams, R., Davey, N., and Sun, Y. (2017b). Investigating optical transmission error correction using wavelet transforms. In *ESANN 2017 proceedings, 25th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning - (Bruges (Belgium), 26-28 April 2017)*, pages 447–452. ESANN.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning.*

Bochkarev, V. V., Shevlyakova, A. V., and Solovyev, V. D. (2015). The average word length dynamics as an indicator of cultural changes in society. *Social Evolution & History*, 14(2):153–175.

Borkowski, R., Zibar, D., Caballero, A., Arlunno, V., and Monroy, I. T. (2013). Stokes space-based optical modulation format recognition for digital coherent receivers. *IEEE Photonics Technology Letters*, 25(21):2129–2132.

Bozinovic, N., Yue, Y., Ren, Y., Tur, M., Kristensen, P., Huang, H., Willner, A. E., and Ramachandran, S. (2013). Terabit-scale orbital angular momentum mode division multiplexing in fibers. *science*, 340(6140):1545–1548.

BrainVoyager QX (2015). Support vector machines (svms). `http://www.brainvoyager.com/bvqx/doc/UsersGuide/MVPA/SupportVectorMachinesSVMs.html`. Accessed: 2015-03-06.

Bronshtein, A. (2017). Train/test split and cross validation in python. `https://towardsdatascience.com/train-test-split-and-cross-validation-in-python-80b61beca4b6`. Accessed: 2017-09-03.

Bustio-Martínez, L., Cumplido, R., Hernández-Palancar, J., and Feregrino-Uribe, C. (2010). On the design of a hardware-software architecture for acceleration of svm's training phase. In *Mexican Conference on Pattern Recognition*, pages 281–290. Springer.

Cao, K.-k., Shen, H.-b., and Chen, H.-f. (2010). A parallel and scalable digital architecture for training support vector machines. *Journal of Zhejiang University-Science C*, 11(8):620–628.

Chang, C.-C. and Lin, C.-J. (2011). Libsvm: a library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3):27.

Chughtai, M. N. (2012). *Study of physical layer impairments in high speed optical networks.* PhD thesis, KTH Royal Institute of Technology.

Chuma, E. L., Meloni, L. G., Iano, Y., and Roger, L. L. B. (2017). Fpga implementation of a de-noising using haar level 5 wavelet transform. In *the Proceedings of XXXV SIMPÃSIO BRASILEIRO DE TELECOMUNICAÃÃES E PROCESSAMENTO DE SINAIS, SBrT2017*, pages 3–6. DE SETEMBRO DE.

Chun-Lin, L. (2010). A tutorial of the wavelet transform. *NTUEE, Taiwan.*

CISCO (2005). Introduction to optical fibers, db, attenuation and measurements. `https://www.cisco.com/c/en/us/support/docs/optical/synchronous-digital-hierarchy-sdh/29000-db-29000.html`. Accessed: 2018-12-10.

Cutajar, M., Gatt, E., Grech, I., Casha, O., and Micallef, J. (2013). Hardware-based support vector machine for phoneme classification. In *EUROCON, 2013 IEEE*, pages 1701–1708. IEEE.

Davis, C. C. (n.d.). Fiber optic technology and its role in the information revolution. `https://www.ece.umd.edu/~davis/optfib.html`. Accessed: 2018-04-20.

Destrero, A., Mosci, S., De Mol, C., Verri, A., and Odone, F. (2009). Feature selection for high-dimensional data. *Computational management science*, 6(1):25–40.

Dhanalakshmi, P., Palanivel, S., and Ramalingam, V. (2009). Classification of audio signals using svm and rbfnn. *Expert systems with applications*, 36(3):6069–6075.

Dobre, O. A., Abdi, A., Bar-Ness, Y., and Su, W. (2005). Blind modulation classification: a concept whose time has come. In *Advances in Wired and Wireless Communication, 2005 IEEE/Sarnoff Symposium on*, pages 223–228. IEEE.

Dong, Z., Khan, F. N., Sui, Q., Zhong, K., Lu, C., and Lau, A. P. T. (2016). Optical performance monitoring: A review of current and future technologies. *Journal of Lightwave Technology*, 34(2):525–543.

Dutton, H. J. (1998). *Understanding optical communications*. Prentice Hall PTR New Jersey.

Essiambre, R.-J., Kramer, G., Winzer, P. J., Foschini, G. J., and Goebel, B. (2010). Capacity limits of optical fiber networks. *Journal of Lightwave Technology*, 28(4):662–701.

Evgeniou, T., Figueiras-Vidal, A. R., and Theodoridis, S. (2008). Emerging machine learning techniques in signal processing.

Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R., and Lin, C.-J. (2008). Liblinear: A library for large linear classification. *Journal of machine learning research*, 9(Aug):1871–1874.

Faruque, S. (2017). Radio frequency modulation made easy. `https://popularelectronics.technicacuriosa.com/2017/03/08/radio-frequency-modulation-made-easy/`. Accessed: 2018-03-20.

Fehske, A., Gaeddert, J., and Reed, J. H. (2005). A new approach to signal classification using spectral correlation and neural networks. In *New Frontiers in Dynamic Spectrum Access Networks, 2005. DySPAN 2005. 2005 First IEEE International Symposium on*, pages 144–150. IEEE.

Glesk, I. (2010). From morse code to terabit communications. In *Telecommunications Conference (HISTELCON), 2010 Second IEEE Region 8 Conference on the History of*, pages 1–6. IEEE.

Gonzalez, N. G., Zibar, D., and Monroy, I. T. (2010). Cognitive digital receiver for burst mode phase modulated radio over fiber links. In *Optical Communication (ECOC), 2010 36th European Conference and Exhibition on*, pages 1–3. IEEE.

Goodfellow, I., Bengio, Y., Courville, A., and Bengio, Y. (2016). *Deep learning*, volume 1. MIT press Cambridge.

Groleat, T., Arzel, M., and Vaton, S. (2012). Hardware acceleration of svm-based traffic classification on fpga. In *Wireless Communications and Mobile Computing Conference (IWCMC), 2012 8th International*, pages 443–449. IEEE.

Hardin, R. (1973). Applications of the split-step fourier method to the numerical solution of nonlinear and variable coefficient wave equations. *Siam Rev.*, (15):423.

Hecht, J. (2011). Ultrafast fibre optics set new speed record. https://www.newscientist.com/article/mg21028095-500-ultrafast-fibre-optics-set-new-speed-record/. Accessed: 2018-04-23.

Heittola, T., Çakır, E., and Virtanen, T. (2018). The machine learning approach for analysis of sound scenes and events. In *Computational Analysis of Sound Scenes and Events*, pages 13–40. Springer.

Hosseinifard, B., Moradi, M. H., and Rostami, R. (2013). Classifying depression patients and normal subjects using machine learning techniques and nonlinear features from eeg signal. *Computer methods and programs in biomedicine*, 109(3):339–345.

Huang, T.-L. (2010). *Comparison of L2-regularized multi-class linear classifiers*. PhD thesis, Masterâs thesis, Department of Computer Science and Information Engineering, National Taiwan University.

Hubbard, B. B. (1998). *The world according to wavelets: the story of a mathematical technique in the making*. AK Peters/CRC Press.

Hunt, S., Sun, Y., Shafarenko, A., Adams, R., Davey, N., Slater, B., Bhamber, R., Boscolo, S., and Turitsyn, S. K. (2009). Adaptive electrical signal post-processing with varying representations in optical communication systems. In *International Conference on Engineering Applications of Neural Networks*, pages 235–245. Springer.

Hunt, S., Sun, Y., Shafarenko, A., Adams, R., Davey, N., Slater, B., Bhamber, R., Boscolo, S., and Turitsyn, S. K. (2010). Correcting errors in optical data transmission using neural networks. In *International Conference on Artificial Neural Networks*, pages 448–457. Springer.

Hunt, S., Sun, Y., Shafarenko, A., Davey, N., Boscolo, S., and Turitsyn, S. (2008). Using simple neural networks to correct errors in optical data transmission.

Isautier, P., Pan, J., DeSalvo, R., and Ralph, S. E. (2015). Stokes space-based modulation format recognition for autonomous optical receivers. *Journal of Lightwave Technology*, 33(24):5157–5163.

Kanprachar, S. (1999). Modeling and analysis of the effects of impairments in fiber optic links.

KDAG (2015). Svm simplified. https://kgpdag.wordpress.com/2015/08/12/svm-simplified/. Accessed: 2017-11-06.

Keim, R. (2016). Understanding quadrature phase shift keying (qpsk) modulation. https://www.allaboutcircuits.com/technical-articles/quadrature-phase-shift-keying-qpsk-modulation/. Accessed: 2018-04-04.

Koide, T., Hoang, A.-T., Okamoto, T., Shigemi, S., Mishima, T., Tamaki, T., Raytchev, B., Kaneda, K., Kominami, Y., Miyaki, R., et al. (2014). Fpga implementation of type identifier for colorectal endoscopie images with nbi magnification. In *Circuits and Systems (APCCAS), 2014 IEEE Asia Pacific Conference on*, pages 651–654. IEEE.

Koike-Akino, T., Duan, C., Parsons, K., Kojima, K., Yoshida, T., Sugihara, T., and Mizuochi, T. (2012). High-order statistical equalizer for nonlinearity compensation in dispersion-managed coherent optical communications. *Optics express*, 20(14):15769–15780.

Komorkiewicz, M., Kluczewski, M., and Gorgon, M. (2012). Floating point hog implementation for real-time multiple object detection. In *Field Programmable Logic and Applications (FPL), 2012 22nd International Conference on*, pages 711–714. IEEE.

Lee, S.-H. and Lim, J. S. (2012). Parkinson's disease classification using gait characteristics and wavelet-based feature extraction. *Expert Systems with Applications*, 39(8):7338–7344.

Li, M., Yu, S., Yang, J., Chen, Z., Han, Y., and Gu, W. (2013). Nonparameter nonlinear phase noise mitigation by using m-ary support vector machine for coherent optical systems. *IEEE Photonics Journal*, 5(6):7800312–7800312.

Library of Congress (n.d.). Library of congress, today in history - may 24, what hath god wrought? https://www.loc.gov/item/today-in-history/may-24/. Accessed: 2018-04-15.

Lin, C.-C., Chen, S.-H., Truong, T.-K., and Chang, Y. (2005). Audio classification and categorization based on wavelets and support vector machine. *IEEE Transactions on Speech and Audio Processing*, 13(5):644–651.

Liu, Z., Cao, H., Chen, X., He, Z., and Shen, Z. (2013). Multi-fault classification based on wavelet svm with pso algorithm to analyze vibration signals from rolling element bearings. *Neurocomputing*, 99:399–410.

Machhout, M. and Tourki, R. (2017). Fpga implementation of svm for nonlinear systems regression. *International journal of advanced computer science and applications*, 8(8).

Maliuk, D., Stratigopoulos, H.-G., and Makris, Y. (2010). An analog vlsi multilayer perceptron and its application towards built-in self-test in analog circuits. In *On-Line Testing Symposium (IOLTS), 2010 IEEE 16th International*, pages 71–76. IEEE.

Massa, N. (2000). Fiber optic telecommunication. `https://spie.org/publications/fundamentals-of-photonics-modules?SSO=1`. Accessed: 2018-12-10.

Matsumoto, M. (2013). Information-rate analysis of a fiber-optic transmission system including 2r signal regenerators. *Optics Express*, 21(22):26762–26773.

McCool, R. (n.d.). Introduction to fibre optics. `http://www.jb.man.ac.uk/research/fibre/intro2fibre.htm#dispersion`. Accessed: 2018-04-20.

Metaxas, A., Redyuk, A., Sun, Y., Shafarenko, A., Davey, N., and Adams, R. (2013). linear support vector machines for error correction in optical data transmission. In *International Conference on Adaptive and Natural Computing Algorithms*, pages 438–445. Springer.

Miller, K. (2011). Chromatic dispersion in optical fibers. `http://www.m2optics.com/blog/bid/61431/Chromatic-Dispersion-in-Optical-Fibers`. Accessed: 2018-06-26.

Mishra, S. (2017). Unsupervised learning and data clustering. `https://towardsdatascience.com/unsupervised-learning-and-data-clustering-eeecb78b422a`. Accessed: 2018-01-05.

Mizuochi, T., Miyata, Y., Kobayashi, T., Ouchi, K., Kuno, K., Kubo, K., Shimizu, K., Tagami, H., Yoshida, H., Fujita, H., et al. (2004). Forward error correction based on block turbo code with 3-bit soft decision for 10-gb/s optical communication systems. *IEEE Journal of Selected Topics in Quantum Electronics*, 10(2):376–386.

Nasreen, N., Lu, D., and Arshad, M. (2018). Optical soliton solutions of nonlinear schrödinger equation with second order spatiotemporal dispersion and its modulation instability. *Optik*, 161:221–229.

NTT, News Release (2006). 14 tbps over a single optical fiber: Successful demonstration of world's largest capacity. `http://www.ntt.co.jp/news/news06e/0609/060929a.html`. Accessed: 2018-04-23.

Peach, M. (2013). Nec and corning achieve petabit optical transmission. `http://optics.org/news/4/1/29`. Accessed: 2018-04-24.

Peng, C.-H., Chen, B.-W., Kuan, T.-W., Lin, P.-C., Wang, J.-F., and Shih, N.-S. (2014). Rec-sta: Reconfigurable and efficient chip design with smo-based training accelerator. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 22(8):1791–1802.

R. Sathya, R. Iswarya, V. R. R. S. (2014). Design and implementation of low cost maritime boundary identification system using fiber optic technology. *International Journal for Research in Applied Science and Engineering Technology*, 2:394–402. ISSN: 2321-9653.

Rajbhandari, S. (2010). *Application of wavelets and artificial neural network for indoor optical wireless communication systems.* PhD thesis, Northumbria University.

Rajbhandari, S., Ghassemlooy, Z., and Angelova, M. (2009a). Bit error performance of diffuse indoor optical wireless channel pulse position modulation system employing artificial neural networks for channel equalisation. *IET optoelectronics*, 3(4):169–179.

Rajbhandari, S., Ghassemlooy, Z., and Angelova, M. (2009b). Effective denoising and adaptive equalization of indoor optical wireless channel with artificial light using the discrete wavelet transform and artificial neural network. *Journal of Lightwave technology*, 27(20):4493–4500.

Redyuk, A., Nanii, O., Treshchikov, V., Mikhailov, V., and Fedoruk, M. (2014). 100 gb s- 1 coherent dense wavelength division multiplexing system reach extension beyond the limit of electronic dispersion compensation using optical dispersion management. *Laser Physics Letters*, 12(2):025101.

Regniers, O., Bombrun, L., Lafon, V., and Germain, C. (2016). Supervised classification of very high resolution optical images using wavelet-based textural features. *IEEE Transactions on Geoscience and Remote Sensing*, 54(6):3722–3735.

Rhoads, M. (2016). Cisco community: Understanding tx rx light level. `https://community.cisco.com/t5/optical-networking/understanding-tx-rx-light-level/td-p/2794532`. Accessed: 2018-08-01.

Rouse, M. (2007). carrier signal. `https://searchtelecom.techtarget.com/definition/carrier-signal`. Accessed: 2018-05-01.

Ruiz-Llata, M., Guarnizo, G., and Yébenes-Calvino, M. (2010). Fpga implementation of a support vector machine for classification and regression. In *Neural Networks (IJCNN), The 2010 International Joint Conference on*, pages 1–5. IEEE.

Sanjeevi, M. (2017). Different types of machine learning and their types. `https://medium.com/deep-math-machine-learning-ai/different-types-of-machine-learning-and-their-types-34760b9128a2`. Accessed: 2018-04-26.

Scholkopf, B. and Smola, A. J. (2001). *Learning with kernels: support vector machines, regularization, optimization, and beyond.* MIT press.

Schütze, H., Manning, C. D., and Raghavan, P. (2008). *Introduction to information retrieval*, volume 39. Cambridge University Press.

Shahi, S. N., Kumar, S., and Liang, X. (2014). Analytical modeling of cross-phase modulation in coherent fiber-optic system. *Optics express*, 22(2):1426–1439.

Shaik, A. (n.d.). Amplitude modulation. `http://www.physics-and-radio-electronics.com/blog/amplitude-modulation/`. Accessed: 2018-08-26.

Shigemi, S., Mishima, T., Hoang, A.-T., Koide, T., Tamaki, T., Raytchev, B., Kaneda, K., Kominami, Y., Miyaki, R., Matsuo, T., et al. (2013). Customizable hardware architecture of support vector machine in cad system for colorectal endoscopic images with nbi magnification. In *Proc. of the 18th Workshop on Systhesis And System Integration of Mixed Information Technologies (SASIMI2013)*, pages 298–303.

Smola, A. J. and Schölkopf, B. (1998). *Learning with kernels.* Citeseer.

StackExchange (2017). What is the span of an optical fiber? `https://electronics.stackexchange.com/questions/299803/what-is-the-span-of-an-optical-fiber`, note = Accessed: 2018-08-01.

Sun, Y., Shafarenko, A., Adams, R., Davey, N., Slater, B., Bhamber, R., Boscolo, S., and Turitsyn, S. K. (2008). Adaptive electrical signal post-processing in optical communication systems. In *In: Procs of the Artificial Neural Networks and Intelligent Information Processing Workshop (ANNIP 2008).* INSTICC (Inst. Syst. Technologies Information Control and Communication).

Tan, A. S., Wymeersch, H., Johannisson, P., Agrell, E., Andrekson, P., and Karlsson, M. (2011). An ml-based detector for optical communication in the presence of nonlinear phase noise. In *Communications (ICC), 2011 IEEE International Conference on*, pages 1–5. IEEE.

Tan, M. C., Khan, F. N., Al-Arashi, W. H., Zhou, Y., and Lau, A. P. T. (2014). Simultaneous optical performance monitoring and modulation format/bit-rate identification using principal component analysis. *Journal of Optical Communications and Networking*, 6(5):441–448.

The MathWorks (1994-2018). Run test for randomness (runstest). `https://uk.mathworks.com/help/stats/runstest.html#bubrl2u-2`, note = Accessed: 2018-06-28.

The Pennsylvania State University (2018). S.3.2 hypothesis testing (p-value approach). `https://newonlinecourses.science.psu.edu/statprogram/reviews/statistical-concepts/hypothesis-testing/p-value-approach`. Accessed: 2018-12-20.

Thrane, J., Wass, J., Piels, M., Diniz, J. C., Jones, R., and Zibar, D. (2017). Machine learning techniques for optical performance monitoring from directly detected pdm-qam signals. *Journal of Lightwave Technology*, 35(4):868–875.

Varma, T., Chitre, V., and Patil, D. (2012). The haar wavelet and the biorthogonal wavelet transforms of an image. *International Journal of Engineering Research and Applications*, 288:291.

Walker, J. S. (2008). *A primer on wavelets and their scientific applications.* CRC press.

Wang, C., Gong, L., Yu, Q., Li, X., Xie, Y., and Zhou, X. (2017). Dlau: A scalable deep learning accelerator unit on fpga. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 36(3):513–517.

Wang, D., Zhang, M., Li, Z., Cui, Y., Liu, J., Yang, Y., and Wang, H. (2015). Nonlinear decision boundary created by a machine learning-based classifier to mitigate nonlinear phase noise. In *Optical Communication (ECOC), 2015 European Conference on*, pages 1–3. IEEE.

Wang, H., Guo, J., Wang, T., Zhang, Q., and Shao, J. (2012). Physical layer design for free space optical communication. In *Control Engineering and Communication Technology (ICCECT), 2012 International Conference on*, pages 978–981. IEEE.

Wang, J., Lin, J., and Wang, Z. (2018). Efficient hardware architectures for deep convolutional neural network. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 65(6):1941–1953.

Wass, J., Thrane, J., Piels, M., Diniz, J. C., Jones, R., and Zibar, D. (2016). Machine learning for optical performance monitoring from directly detected pdm-qam signals. In *ECOC 2016; 42nd European Conference on Optical Communication; Proceedings of*, pages 1–3. VDE.

Wavelet Toolbox (n.d.). Wavelet shapes. `http://radio.feld.cvut.cz/matlab/toolbox/wavelet/ch06_ad7.html#36137`. Accessed: 2018-12-19.

Wikipedia (2017). Field-programmable gate array — wikipedia, the free encyclopedia. Accessed: 20-June-2018.

Wikipedia (2018a). Baud. `https://en.wikipedia.org/wiki/Baud`. Accessed: 2018-07-26.

Wikipedia (2018b). Carrier wave. `https://en.wikipedia.org/wiki/Carrier_wave`. Accessed: 2018-08-01.

Wikipedia (2018c). dbm. `https://en.wikipedia.org/wiki/DBm`. Accessed: 2018-12-10.

Wikipedia (2018d). Optical fiber. `https://en.wikipedia.org/wiki/Optical_fiber`. Accessed: 2018-07-25.

Wikipedia (2018e). Signal-to-noise ratio. `https://en.wikipedia.org/wiki/Signal-to-noise_ratio`. Accessed: 2018-07-26.

Wikipedia (2018f). Wavelength-division multiplexing. `https://en.wikipedia.org/wiki/Wavelength-division_multiplexing`. Accessed: 2018-05-26.

Williams, J. R. and Amaratunga, K. (1994). Introduction to wavelets in engineering. *International Journal for Numerical Methods in Engineering*, 37(14):2365–2388.

Woodford, C. (2018). Fiber optics (types of fiber-optic cables). `http://www.explainthatstuff.com/fiberoptics.html`. Accessed: 2018-08-22.

Woodward, B. (2014). *Fiber Optics Installer (FOI) Certification Exam Guide*. John Wiley & Sons.

Wu, X., Jargon, J. A., Paraschis, L., and Willner, A. E. (2011). Ann-based optical performance monitoring of qpsk signals using parameters derived from balanced-detected asynchronous diagrams. *IEEE Photonics Technology Letters*, 23(4):248–250.

Wu, X., Jargon, J. A., Skoog, R. A., Paraschis, L., and Willner, A. E. (2009). Applications of artificial neural networks in optical performance monitoring. *Journal of Lightwave Technology*, 27(16):3580–3589.

Yadav, A. R., Anand, R., Dewal, M., and Gupta, S. (2015). Performance analysis of discrete wavelet transform based first-order statistical texture features for hardwood species classification. *Procedia Computer Science*, 57:214–221.

Yao, S. (2003). Polarization in fiber systems: Squeezing out more bandwidth. *The Photonics Handbook*, page 1.

Yuan, G.-X., Ho, C.-H., and Lin, C.-J. (2012). Recent advances of large-scale linear classification. *Proceedings of the IEEE*, 100(9):2584–2603.

Zibar, D., de Carvalho, L. H. H., Piels, M., Doberstein, A., Diniz, J., Nebendahl, B., Franciscangelis, C., Estaran, J., Haisch, H., Gonzalez, N. G., et al. (2015). Application of machine learning techniques for amplitude and phase noise characterization. *Journal of Lightwave Technology*, 33(7):1333–1343.

Zibar, D., Winther, O., Franceschi, N., Borkowski, R., Caballero, A., Arlunno, V., Schmidt, M. N., Gonzales, N. G., Mao, B., Ye, Y., et al. (2012). Nonlinear impairment compensation using expectation maximization for dispersion managed and unmanaged pdm 16-qam transmission. *Optics express*, 20(26):B181–B196.

Zou, H., Hastie, T., and Tibshirani, R. (2006). Sparse principal component analysis. *Journal of computational and graphical statistics*, 15(2):265–286.