Applications of Deep Learning Networks

BS Electrical Engineering

Session: 2018 - 19

By:

Usman Akram	19100005
Muhammad Khizar Anjum	19100198

Supervised by: Dr. Muhammad Tahir,
Assistant Professor, EE

Co-Supervised by: Dr. Momin Ayyub Uppal,
Associate Professor, EE

Department of Electrical Engineering

Lahore University of Management Sciences

Abstract

Information processing plays a very central role in today's digitized world. It plays an important role in almost every field of life. These algorithms work behind the scenes to drive businesses, launch humans to the moon, and keep the world connected. The internet itself is a great information processing marvel. Therefore, it is imperative that we look into using new and contemporary technologies to enhance and optimize these systems, and make them faster and more robust; Deep learning is one such technology. Within a decade, the development of deep learning has shown a lot of promise for a variety of applications. This new and emerging domain in Data Science has already upended the fields of Computer Vision and Acoustics where high classification rates of around 99% for images and acoustic events –once thought a daunting, or perhaps, an impossible task was made trivial through the ingenious and innovative use of Convolutional Neural Networks and ResNets. Moreover, the use of deep learning has also shown promising results in the areas of genetics and medical sciences, where research is being done to detect –and therefore preempt- rare diseases and conditions using deep learning. Given this diversity of applications and areas where deep learning is the object of the immense focus of the research paradigm, we have decided to explore this research area for deep learning broadly. In our project, we aim to implement deep learning to investigate the potential improvements the use of Neural Networks can bring to the conventional frameworks of Information Processing. Our work is bifurcated into two broad spheres of focus; in one application, we explore the potential of the Neural networks to replace the modulation and/or demodulation blocks of traditional communication systems; in another application, we examine the prospects of using deep learning networks for early diagnosis of Parkinson's disease.

Acknowledgments

The authors wish to express sincere appreciation to Dr. Muhammad Tahir and Dr. Momin Uppal for their dedicated supervision during the course of this senior year project and in the preparation of this manuscript. Thanks also to Waleed Akbar and Hamza Ather for their unrelenting assistance for our project.

Originality Certificate

“We the undersigned certify that this submission is the original work of members of the group and meets the Faculty's Expectations of Originality”:

1. Usman Akram [2019-10-0005]
   Signed: ____________________ 19100005@lums.edu.pk

2. Muhammad Khizar Anjum [2019-10-0198]
   Signed: ____________________ 19100198@lums.edu.pk

Advisor
Dr. Muhammad Tahir

____

tahir@lums.edu.pk

Co-Advisor(s)
Dr. Momin Ayyub Uppal

____

momin.uppal@lums.edu.pk

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

The objective of our project is to explore the potential of deep neural networks in areas of information processing and medical sciences. We have observed its potential in Computer vision and acoustics, and we believe that these avenues have a lot of potential for improvement as well. Currently, these areas are saturated with hand-crafted algorithms which are closed-form mathematical solutions to messy natural phenomenon like communication-channels or clinical data. The problem with closed-form mathematical solution is that they make certain assumptions about nature which leads to discrepancies between natural phenomenon and model predictions. However, deep learning lets us approximate this messy-phenomena in the weights and non-linearities of the network, given that we have enough data. Hence, for most times, a deep-network can perform better than hand-crafted methods as it can model those irregularities in nature. Hence, we intend to improve the performance of traditional methods with our deep-learning models and make current systems better.

1.2 Problem Statement

We aim to investigate the prospects of optimizing or supplanting traditional frameworks in the field of communication systems and bioinformatics through the use of Deep Learning.

1.3 General Block Diagram

1.3.1 Application in Communication Systems

Generally, as illustrated in Figure 1.1, we use information processing to convert the data from one form to the other. We usually prefer a compact representation of the data to facilitate its transmission, storage, or even further processing. Nonetheless, the processing of data is always prone to noise and loss, and if we wish to recover the data from its processed form, we might get an error that can or cannot be neglected depending upon our application.

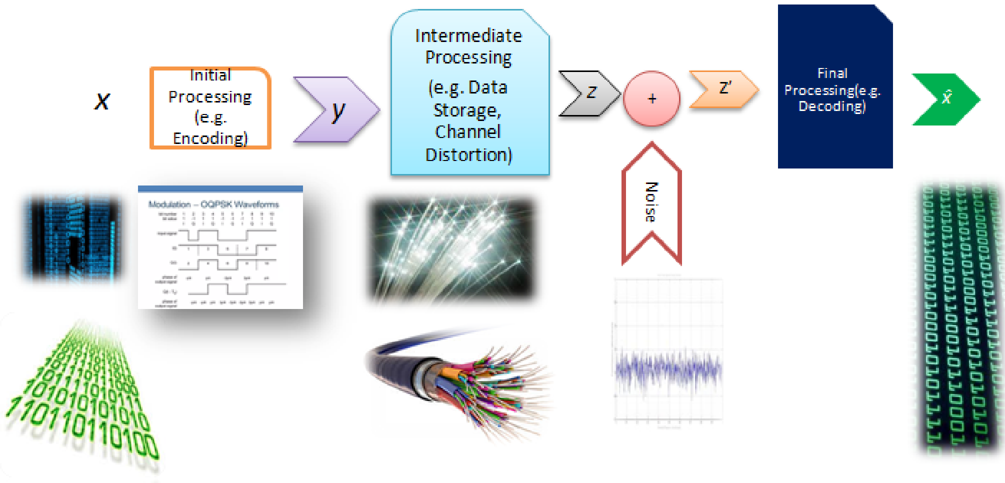

x
Initial Processing (e.g. Encoding)
y
Intermediate Processing (e.g. Data Storage, Channel Distortion)
z
+
z'
Final Processing(e.g. Decoding)
$\hat{x}$
Noise
Modulation – OQPSK Waveforms

FIGURE 1.1: The general block diagram depicting the processing and recovery of data

Ideally, we want the retrieved data $\hat{x}$ to be equal to the original data x so that the error e defined as $x - \hat{x}$ is 0; however, this is mostly unachievable due to various factors; noise, interference, hardware nonlinearities such as amplifier saturation. In addition to these external factors that are normally not in our control, there are also limitations inherent in the assumptions and models we employ that may not be adequate in capturing the complexities of the system. Hence, we try to minimize the error through various algorithms and signal processing techniques that have solid mathematical foundations. Nonetheless, these traditional techniques are as good (or bad) in capturing the complexities of our data as our models and assumptions!
The use of deep learning can greatly enhance error optimization. The Neural Network learns the model on its own by fitting the training data –labeled or unlabeled-

into the model to minimize a particular loss function; this eliminates the inefficiencies that were concomitant with the modeling the system. By iterating through the batches of training data for a sufficient number of epochs, the neural network finds the optimal sets of weights that result in minimal loss. We can view the neural network as the model in itself! But, care must be taken in generalizing this statement; the caveat here is that nonlinear models can only be learned using neural networks with nonlinear activation functions!

In communication systems, when the (modulated) data is transmitted through a channel (Copper Wires, Optical Fibers, Air, Water e.t.c), it is convolved with the channel's impulse response and corrupted by noise and disturbances. Data at the receiver is vastly different than the one that was originally transmitted. Hence, the receiver is encumbered with the additional task of deconvolving and the denoising the received data. If it were not for the noise and channel impulse response, the received data could have been demodulated to 100% accuracy (0 BER) by merely reversing the explicit processes that take place at the modulator and the transmitter. The block diagram for an OFDM system Figure 1.2 is given below:

The effects of noise and channel distortion are highly nonlinear. This is why the use of linear operations to deconvolve and de-noise the received signal only yields limited success. The ability of the neural network to learn the nonlinear features of a system can be used to great advantage. Nonlinear activation functions like ReLU, Sigmoid, Tanh enable the network to learn complex features that would otherwise be overlooked by traditional methods. In the case of OFDM, we can transmit known data bits through the channel, save the corresponding received signals, and train the neural network using this determinate data. To minimize the loss function, the neural networks updates its weights using optimizers such

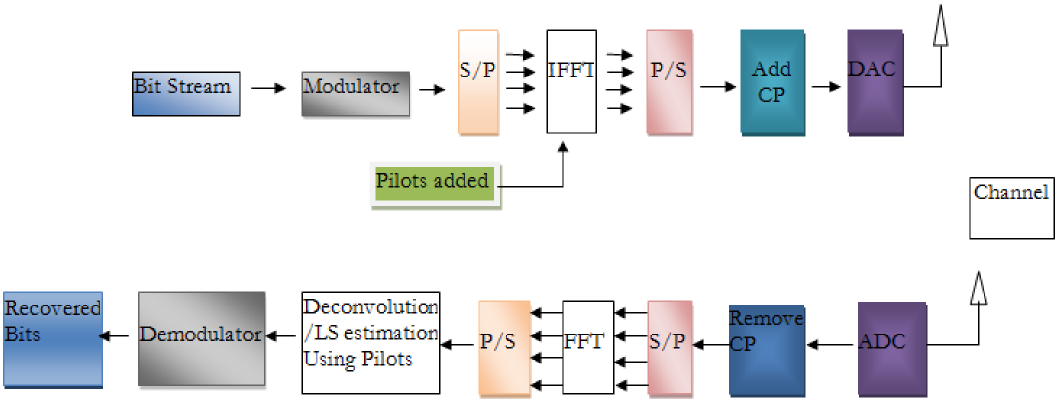


FIGURE 1.2: A traditional single-user OFDM system

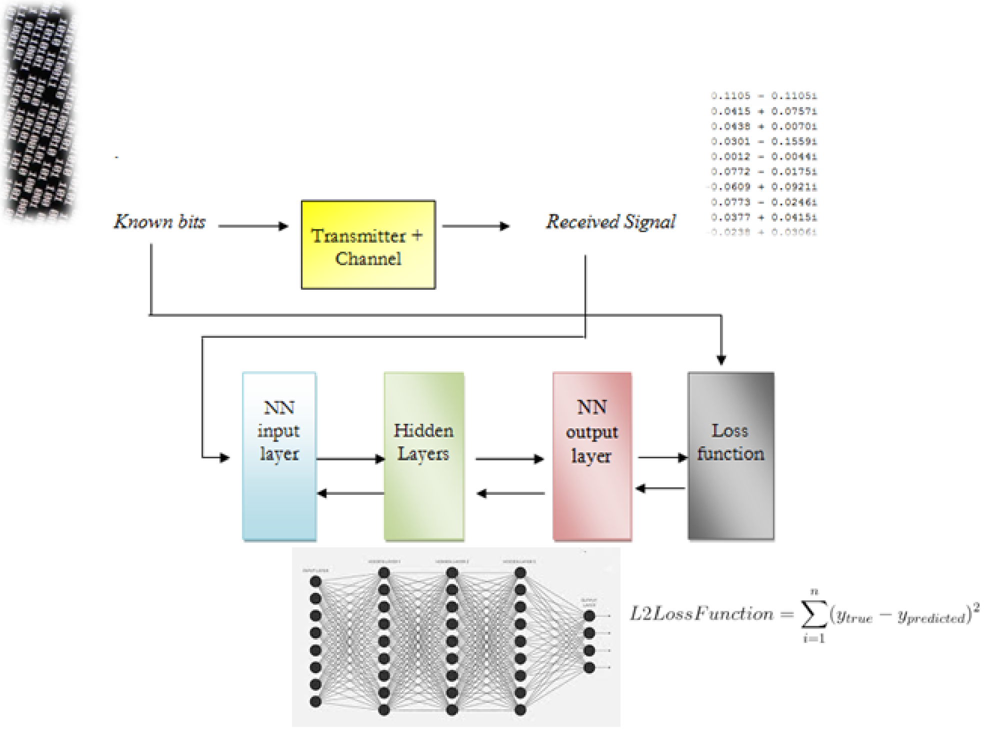


FIGURE 1.3: The Receiver is replaced by a Dense Neural Network (DNN)

as Stochastic Gradient Descent (SGD) and Adam's optimizer, and in the process, learns these two things about the system:

1. The processes that take place on the transmitter side (e.g., Modulation, CP addition, Pilots, etc.).
2. The features of the channel and the noise that distort the transmitted data

Once the loss has been minimized to a sufficient degree, we use the trained neural network to replace the entire receiver side except DAC. This is illustrated in Figure 1.3 and Figure 1.4.

We also extend the block diagram above for multi-user detection, in which a single receiver recovers the data bits transmitted from 2 independent users using the Non-Orthogonal Multiple Access (NOMA) scheme.

1.3.2 Application in Bioinformatics

For medical applications, we are working towards early diagnosis of Parkinson's disease (PD). It affects more than 6 million people worldwide and is the second most common neurodegenerative disease after Alzheimer's disease. There are a

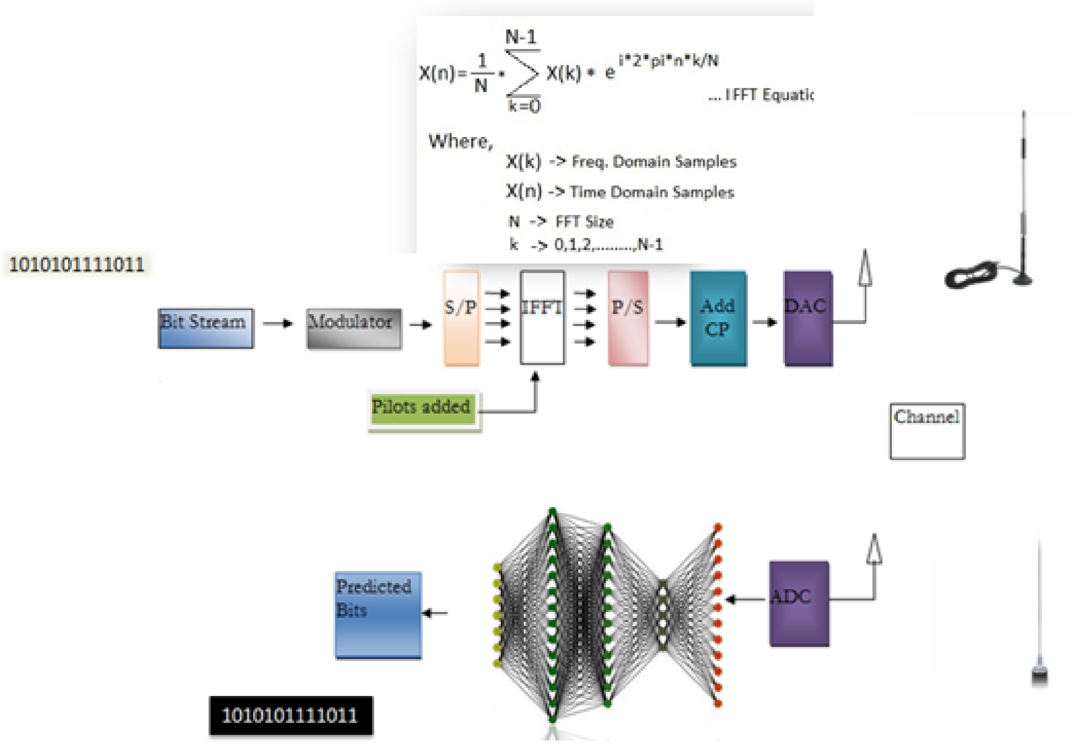


FIGURE 1.4: A complete OFDM system with a DNN as its receiver

myriad number of symptoms for Parkinson's and these symptoms of PD progressively worsen over time, leading to a stark loss in quality of life, and a significant reduction in life expectancy. A detailed account of the Parkinson's symptoms is shown in Figure 1.5. Receiving a timely and accurate diagnosis is paramount for

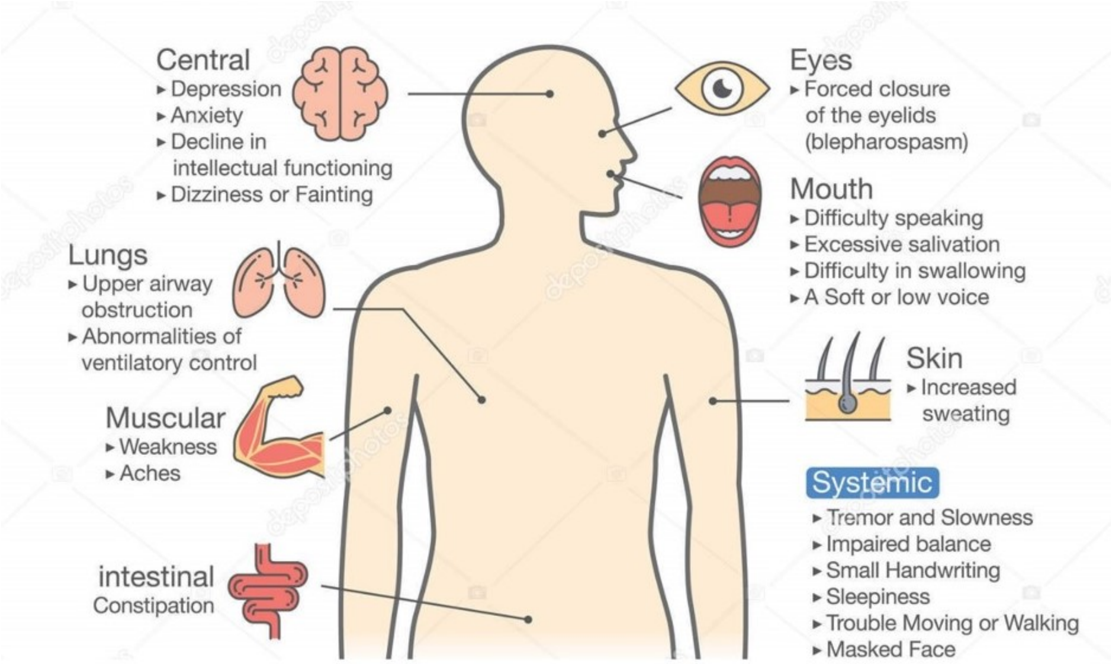


FIGURE 1.5: An Illustration of Parkinson's Symptoms

patients because access to treatments could improve their quality of life [1]. So far, the traditional methods of diagnosing PD are based on subjective clinical assessments of patient's symptoms. However, research has shown that around 25% of this diagnosis are incorrect when compared to the results of post-mortem autopsy [2]. These diagnoses are difficult because there are other diseases that may appear similar to PD and symptom severity may fluctuate over time [2]. Adding to that, patients In this project, we explore the possibility of using smartPhone data collected by SageBionetworks, and made public under the umbrella of mPower study, to diagnose people with PD [3]. Such an approach frees the diagnosis process from sporadic clinical trials and gives one the ability to do it anytime one wants. Also, this lets us explore the possibility of using deep-learning models for diagnosis task, which may improve significantly on the previous results.

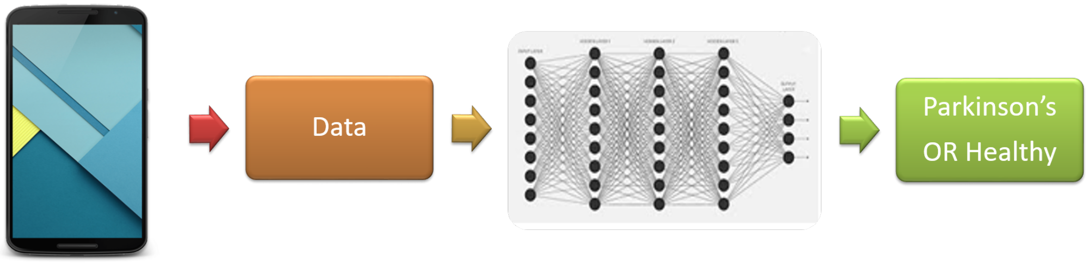


FIGURE 1.6: A General Block Diagram for the Parkinson's Diagnosis

The model that we propose for the detection for the early diagnosis of Parkinson's is to optimize a Deep Learning Model that takes in data from Smartphone (Here we use the data already collected under the mPower study) and makes predictions based on this data input as shown in Figure 1.6.

1.4 Social Benefits And Relevance

Information processing systems and health are at the heart of today's globalized world. Our project targets two very important areas of social significance. If we are able to achieve better performance in these applications, we will impact a great number of people who use these technologies in their everyday lives.

Specifically, the application in Bioinformatics has great potential to affect people's quality of life. The traditional method for Parkinson's diagnosis is clinical diagnosis which is cumbersome and even then misclassifies 25% of Parkinson's cases [2]. Also, given that Parkinson's shares many of the symptoms with other diseases, patients usually are not motivated to go for a full-fledged clinical diagnosis process, which may lead to worsening of Parkinson's until the point of no recovery. However, if we are able to achieve a good enough performance on our project, we may be able to provide a rough heuristic (if not a complete diagnosis) for people to get themselves professionally checked for PD and that kind of early diagnosis can dramatically improve people's quality of life [2].

1.5 Goals

Since this was primarily a research project, our goals were intended to evolve with time as we were to complete our initial deliverables and sift through a sufficient amount of research papers and other resource materials. For communication systems, our primary goal was to first minimize the Bit Error Rate (BER) for complex channel models in simulation for different communication systems (e.g., OFDM, NOMA), and compare them with the traditional methods of equalization; this was somewhat achieved.

With regards to application in bioinformatics, we targeted the early and easy access self-monitoring of PD using smartphones. We aimed to build towards developing deep learning models which could be deployed on smartphones and can efficiently predict if people have PD using smartphones. Specific activities designed to trigger Parkinson's symptoms were to be performed and recorded on smartphones, which were then to be input to our Deep Learning model giving us the predictions.

1.6 Objectives

Our objective was to achieve the performance of the Neural Network that would be on par with the traditional methods. Once that was attained, we added complexities to our system so that the implementation of the network would become more accurate and more practical.

1.7 Outcomes

The areas we are working on have a big part to play in the modern world. If we are able to improve on the performance of these algorithms, it will have direct implications on the Information Processing tasks and the Health of people all around the world. This could potentially mean a better world for all of us.

1.8 Deliverables (Outputs)

As our main motive is to improve on the hand-crafted algorithms in Information Processing and health, for both of our investigations: one into modulation and demodulation blocks of communication systems and the other into diagnosis of PD using smart phone data, we intend to deliver functional and trained deep neural networks which are able to perform the tasks mentioned. We also intend to compare *the performance of our frameworks with current hand-crafted algorithms.*

1.9 Timelines and the distribution of work

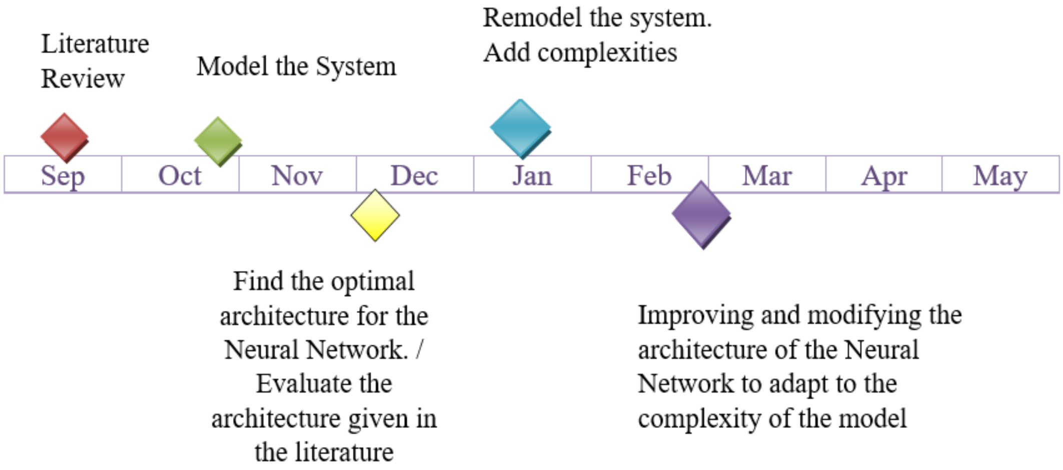

Literature Review
Model the System
Remodel the system. Add complexities
Sep
Oct
Nov
Dec
Jan
Feb
Mar
Apr
May
Find the optimal architecture for the Neural Network. / Evaluate the architecture given in the literature
Improving and modifying the architecture of the Neural Network to adapt to the complexity of the model

FIGURE 1.7: The general summary of the timeline of our work

For application in communication systems, we had modeled the OFDM system by the end of September. It took us merely a week to verify the results of the paper our work depended on which consisted of training a Dense Neural Network with the specified architecture. The only deviation from the paper was that we did not use WINNER II and instead used Real Gaussian Channel unchanging across frames. In November, we made the channel complex and increased the number of channel variations, and recorded results for different SNRs. Until December, we worked on Single User Detection. However, in January, we decided to extend our work to the multi-user system and used NOMA. By February, we had tried different architecture for the increased complexity of the system. We found out that the architecture that gave us the best results was a mere extension of the DNN we used for single user detection. We added more neurons to our networks and added a dropout layer to prevent overfitting. In March and April, we compared the performance of our Neural Network against the benchmarks set by the traditional methods.

For application in bioinformatics, we had reviewed the literature and gained access to the mPower study data. By the End of October, we were successful in restructuring the data to input it into our model. By December, we had successfully implemented the spline CNN [4] for the classification of Parkinsons's voice data recordings. Later, in January, we implemented a policy for the reclassification of data according to a fixed number of recordings per person. After this policy implementation, we moved on to the implementation of an Evidence Aggregation Model to finally output the predictions for each person. In March and April, we compared the performance of our Neural Network architecture against the benchmarks set by the traditional methods.

Chapter 2

Background

2.1 Literature Review

2.1.1 Application in Communication Systems

During recent years, researchers have experimented with various neural network architectures to achieve optimal results. The simplest of architectures that have been used is a DNN with dense, fully connected layers. Generally, the number of hidden layers is kept to a minimum (3-4) as back propagation attenuates with the increasing number of layers. In communication systems, the problem is treated as a regression one instead of a classification one; hence the loss function used to train the neural networks for relevant applications is L2 norm instead of Softmax or Cross Entropy. Recent publications corroborate that for WINNER II channel model with AWGN noise in OFDM systems, the neural networks work slightly better than LS estimation or MMSE for 64 symbol pilot, and significantly better with an 8 symbol pilot without the use of a Cyclic Prefix (CP) [5].

Nonetheless, it is important to accentuate that the DNN in these works are mostly trained offline, and the bit error rates are calculated in the simulation rather than real time. More research on the use of end to end autoencoder architecture for OFDM systems has highlighted ability of the neural network to produce results comparable to MMSE and LS without the use of explicit equalizer on a single tap channel [6]. Moreover, the use of explicit equalizer has been observed to further improve the performance of the autoencoder [6]. There is also no noticeable degradation in the performance of the autoencoder if the pilot is omitted. The autoencoder architecture is also found to be robust to Carrier Frequency Offset (CFO) [6]. However, the discontinuity posed by the channel block to the back-propagation poses a challenge to any practical implementation of the end to end autoencoders.

In [5] and [6], it is assumed that we have prior information about the channel. Specific models are used for training the neural network offline and then using it in real time. However, given the complexities of the channel, there are little prospects for improvements neural networks can bring to real-time workings of communication systems. Nonetheless, recent research into Generative Adversarial Net (GAN) is very encouraging; GAN enables the use of end to end auto encoders without any prior information about the channel [7].

2.1.2 Application in Bioinformatics

Parkinson's disease is a neurodegenerative disease that can affect a person's movement, speech, dexterity, and cognition. Physicians primarily diagnose Parkinson's disease by performing a clinical assessment of symptoms. However, misdiagnoses are common. Research has shown that around 25% of these diagnoses are incorrect when compared to the results of post-mortem autopsy [2]. One factor that contributes to misdiagnoses is that the symptoms of Parkinson's disease may not be prominent at the time the clinical assessment is performed [8]. Another problem is the cumbersome process that discourages people from getting a clinical Parkinson's diagnosis, which may lead to worsening of Parkinson's until the point of no recovery. However, if we are able to achieve a good enough performance on our project, we may be able to provide a rough heuristic (if not a complete diagnosis) for people to get themselves professionally checked for PD and that kind of early diagnosis can dramatically improve people's quality of life [2].

We are working on a deep learning approach to distinguish healthy patients from Parkinson's patients using open-source data from mPower study [3]. Machine Learning algorithms have already been applied to diagnose other diseases as well for example, Breast Cancer [9], cardiac risk factors [10], skin cancer [11] and depression [12].

The data we are using consists of four different activities which are walking, tapping, memory and voice [3]. Previous work on this data has achieved very impressive performance, i.e., 0.85 area under characteristic curve (AUC) [8] using all modes of data and only 0.56 AUC using only voice data. This previous work uses expert hand-crafted features [13] which may be limiting the full potential of this data as these features can be suboptimal. Our goal is to implement an end-to-end deep learning algorithm to explore the options for better discrimination between healthy and Parkinson's patients.

Chapter 3

Methodology and Tools

3.1 System Level Design

3.1.1 Application in Communication Systems

Traditional methods for data recovery include channel estimation through the use of pilots. In that, LS estimation or MMSE is used. At first, we investigate the performance of Deep Learning architecture for a constant, single tap channel with white noise at specified SNR.

Then, we increase the number of channel taps to 8 and make a Gaussian channel. By varying channel parameters and types (No of Channel Realizations, Fading or Non fading etc.), we repeat the analysis.

We then use DNN to recover data bits transmitted from two different users with non-orthogonal resource allocation, at a single receiver

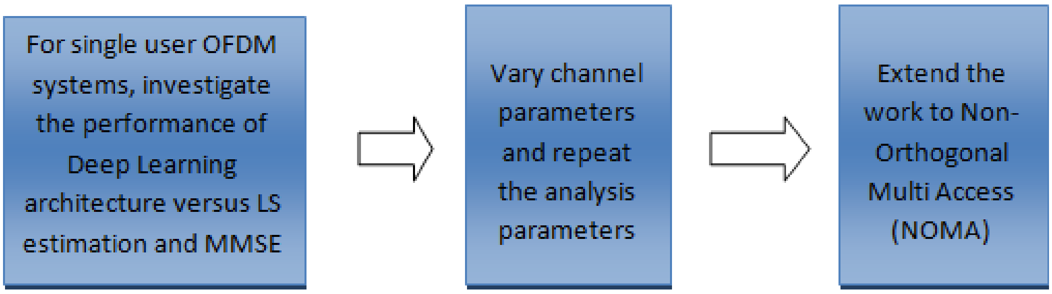

For single user OFDM systems, investigate the performance of Deep Learning architecture versus LS estimation and MMSE
Vary channel parameters and repeat the analysis parameters
Extend the work to Non-Orthogonal Multi Access (NOMA)

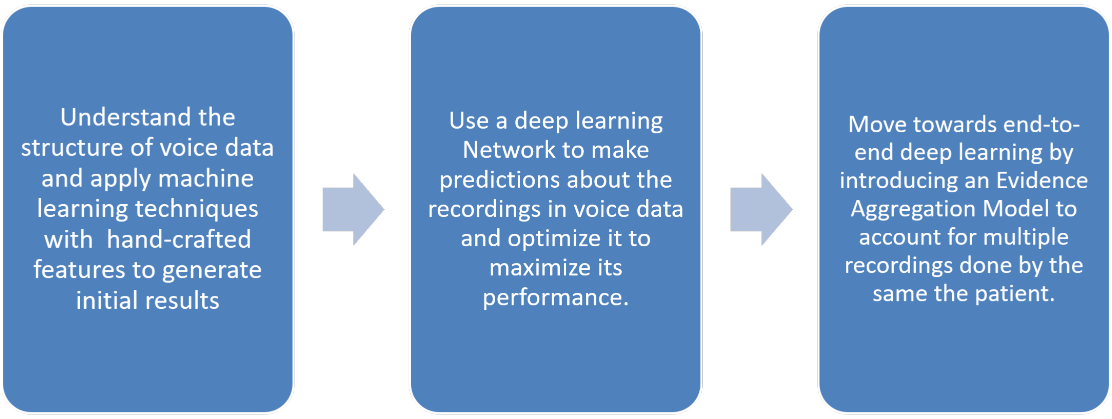
Figure 3.1: System level design for Communication Systems

3.1.2 Application in Bioinformatics

Traditional methods for Parkinson's diagnosis involve a subjective clinical diagnosis. However, through the use of mPower data [3], we intend to augment that process with an easy-to-use predictive Neural Network architecture. We do that by understanding the data provided to us first and then apply Machine Learning techniques with hand-crafted people to generate initial results.

Then we replace those networks with more sophisticated Deep Learning Models and improve upon those results.

After that, we move towards End-to-End Deep Learning which removes all the steps required to prepare features for input to our model.



Understand the structure of voice data and apply machine learning techniques with hand-crafted features to generate initial results

Use a deep learning Network to make predictions about the recordings in voice data and optimize it to maximize its performance.

Move towards end-to-end deep learning by introducing an Evidence Aggregation Model to account for multiple recordings done by the same the patient.

FIGURE 3.2: System level design for Bioinformatics

3.2 Tools/Instruments

3.2.1 Simulation Software Packages

We used the following sofwares/libraries for our applications:

- MATLAB
- Python
- Tensorflow
- Keras
- Scipy
- Numpy

- Scikit-Learn

3.2.2 Hardware Instruments

We used **SOHRAB** Cluster at LUMS for our our simulation process. Its specifications are:
4 way SMP 48 Core E7-4830 v3 @ 2.10GHz, 256 GB RAM (Raid 10) 5*1.2 Tb - 3.3 Tb storage, Grid K40M GPU

Chapter 4

Project Implementation and Simulation

4.1 Simulation Setup

4.1.1 Application in Communication Systems

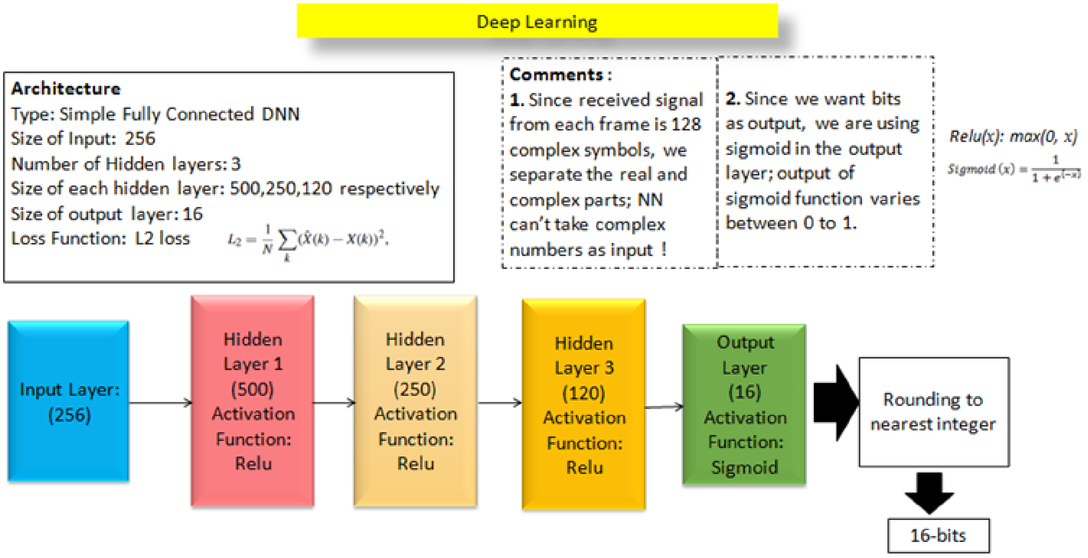
4.1.1.1 OFDM Systems

Firstly, the data is generated in Matlab for traditional OFDM frames. There are 64 subcarriers. The data and pilot bits, both of size 128, are generated independently and are then mapped onto QPSK modulation scheme resulting in sequences of size 64 symbol. After converting them to time-domain by taking their 64-point IFFT, the cyclic prefix of size 16 symbols is added to both sequences after which they are concatenated. The frame of size 180 symbols is convolved with the channel impulse response, and then the Additive White Gaussian Noise is added to it. The cyclic prefix is then removed, and the pilot and data sequences are extracted. The resulting frame size of 128 symbols is ready for analysis.
We investigate three techniques used to recover the original data:

1. **Least Squares (LS) Estimation:** In this approach, the pilots are explicitly used for Channel State Information (CSI) estimation. The DFT of the channel is obtained by dividing the 64-point FFT of the pilot in the received frame by the 64 point FFT of the transmitted frame. The estimated modulated data is then obtained by dividing the 64-point FFT of the data portion from the received frame and the estimated DFT of the channel impulse response. The estimated modulated data can then be mapped back to 128 bits using QPSK constellation diagram.

2. **Time Domain Processing:** This is similar to the above-mentioned scheme except for the method of estimating H. Using our assumption on the number of channel taps, the convolution can be used to derive a matrix equation linking the channel impulse response, transmitted pilot sequence, and the received pilot sequence. Once h[n], its DFT can be computed, and the remainder of the process is the same as before.

3. **Deep Learning:** Here, we use a neural network whose architecture is explained in the diagrams below. The neural network is trained using Keras libraries for Tensorflow in python. The details of the Neural Network Architecture are illustrated in Figure 4.1 and Figure 4.2.



Architecture
Type: Simple Fully Connected DNN
Size of Input: 256
Number of Hidden layers: 3
Size of each hidden layer: 500,250,120 respectively
Size of output layer: 16
Loss Function: L2 loss $L_2 = \frac{1}{N}\sum_k (\hat{X}(k) - X(k))^2,$

Comments :
1. Since received signal from each frame is 128 complex symbols, we separate the real and complex parts; NN can't take complex numbers as input !

2. Since we want bits as output, we are using sigmoid in the output layer; output of sigmoid function varies between 0 to 1.

$Relu(x): max(0, x)$

$Sigmoid\,(x) = \frac{1}{1+e^{(-x)}}$

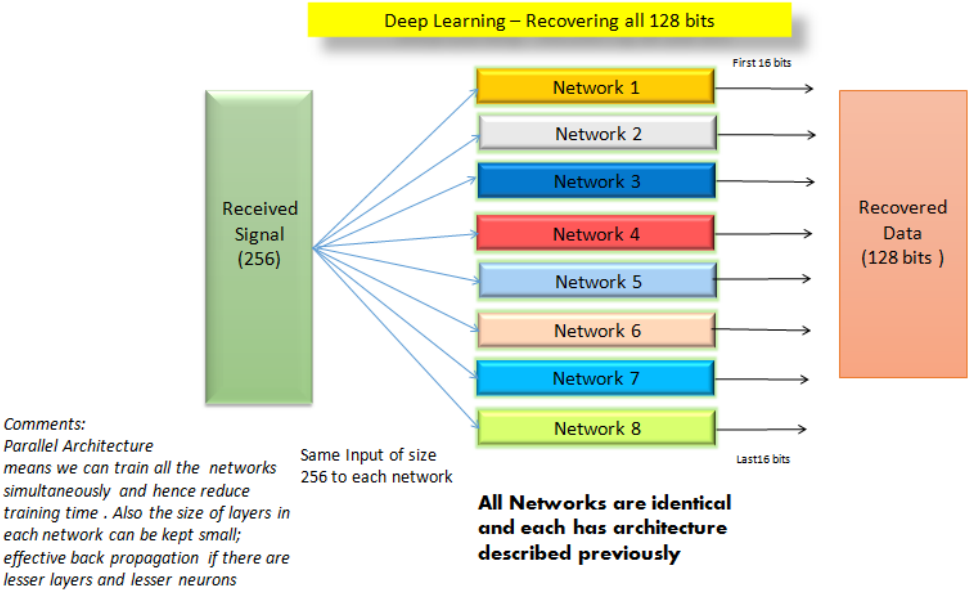
Input Layer: (256)
Hidden Layer 1 (500) Activation Function: Relu
Hidden Layer 2 (250) Activation Function: Relu
Hidden Layer 3 (120) Activation Function: Relu
Output Layer (16) Activation Function: Sigmoid
Rounding to nearest integer
16-bits

FIGURE 4.1: A single Neural Network block to recover 16 bits

4.1.1.2 NOMA Systems

Using the structure for simulation for OFDM system as a skeleton, we extend our work to Non-Orthogonal Multiple Access (NOMA) system. We keep our work restricted to 2 users only. However, since the interference from two users increases the complexity of the problem at hand, we start with 16 subcarriers and a 3 tap channel to observe the trends, and then simulate for 64 subcarriers and 8 tap channel systems to confirm those trends.

The frame structure for NOMA is sufficiently similar to the one for OFDM system. For 16 subcarrier system, 32-bit data is converted to 16 complex symbols using QPSK modulation. Then 16 point IFFT is carried out on these symbols. Moreover, as in the case of OFDM, the pilot bits, equal in length to the data bits


Deep Learning – Recovering all 128 bits
Received Signal (256)
First 16 bits
Network 1
Network 2
Network 3
Network 4
Network 5
Network 6
Network 7
Network 8
Last16 bits
Recovered Data (128 bits)
Same Input of size 256 to each network
All Networks are identical and each has architecture described previously

Comments:
Parallel Architecture means we can train all the networks simultaneously and hence reduce training time . Also the size of layers in each network can be kept small; effective back propagation if there are lesser layers and lesser neurons

FIGURE 4.2: The whole architecture that recovers all 128 bits

(true for most of our work) is QPSK modulated and converted to time-domain. The pilot is then concatenated with time domain data. However, this time space is reserved for the pilot of the other user. This ensures that while the data of the two users are non-orthogonal, the pilots are orthogonal in time. This is illustrated in figure 4.3.

After interference, Gaussian noise is added to signal before it is passed onto the

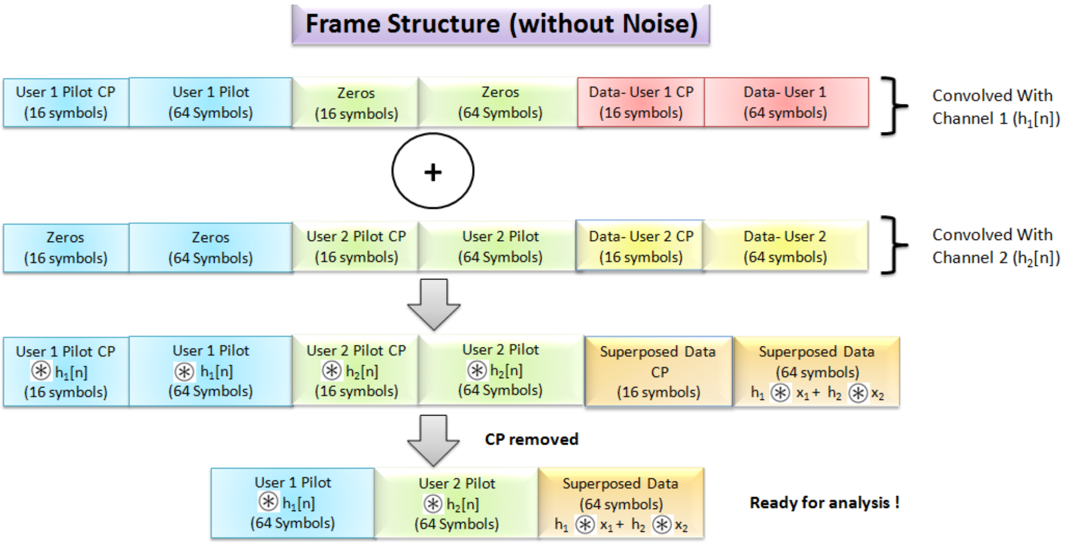

Frame Structure (without Noise)
User 1 Pilot CP (16 symbols)
User 1 Pilot (64 Symbols)
Zeros (16 symbols)
Zeros (64 Symbols)
Data- User 1 CP (16 symbols)
Data- User 1 (64 symbols)
Convolved With Channel 1 ($h_1[n]$)
+
Zeros (16 symbols)
Zeros (64 Symbols)
User 2 Pilot CP (16 symbols)
User 2 Pilot (64 Symbols)
Data- User 2 CP (16 symbols)
Data- User 2 (64 symbols)
Convolved With Channel 2 ($h_2[n]$)
User 1 Pilot CP $\circledast h_1[n]$ (16 symbols)
User 1 Pilot $\circledast h_1[n]$ (64 Symbols)
User 2 Pilot CP $\circledast h_2[n]$ (16 symbols)
User 2 Pilot $\circledast h_2[n]$ (64 Symbols)
Superposed Data CP (16 symbols)
Superposed Data (64 symbols) $h_1 \circledast x_1 + h_2 \circledast x_2$
CP removed
User 1 Pilot $\circledast h_1[n]$ (64 Symbols)
User 2 Pilot $\circledast h_2[n]$ (64 Symbols)
Superposed Data (64 symbols) $h_1 \circledast x_1 + h_2 \circledast x_2$
Ready for analysis !

FIGURE 4.3: Frame Structure for NOMA system: the pilot symbols are preserved despite interference

receiver. Unless otherwise specified, the simulations must be assumed to have been carried out at 20 SNR.

The method for theoretical estimation of BER is somewhat different than the one used for OFDM systems. As before, we estimate the DFT of the channel impulse response using the pilots. We then use Maximum Likelihood estimation at a subcarrier level in frequency domain to infer which of the four possible symbols was transmitted. This can be illustrated in the following equations:

$$Y_{(data)} = H_1 X_{1(data)} + H_2 X_{2(data)} + Z$$

where,

$$X_1, X_2 \in [e^{j(2k+1)\frac{\pi}{4}}] \mid k \in \{-2, -1, 0, 1\}$$

$$F^{-1}(Z) \in \mathcal{N}(0, \sigma^2)$$

The Conditional Distribution on Y is given by:

$$f(Y \mid X_1, X_2, H_1, H_2, \sigma^2) \propto e^{-\frac{|Y - H_1 X_1 - H_2 X_2|^2}{2\sigma^2}}$$

$$\implies \hat{X}_1 = \arg \max_{x_1 \in QPSK} f(Y \mid X_1 = x_1) = \arg \max_{x_1 \in QPSK} \sum_{x_2 \in QPSK} f(Y \mid X_1 = x_1, X_2 = x_2)$$

$$\implies \hat{X}_1 = \arg \max_{x_1 \in QPSK} \sum_{x_2 \in QPSK} e^{-\frac{|Y - H_1 x_1 - H_2 x_2|^2}{2\sigma^2}}$$

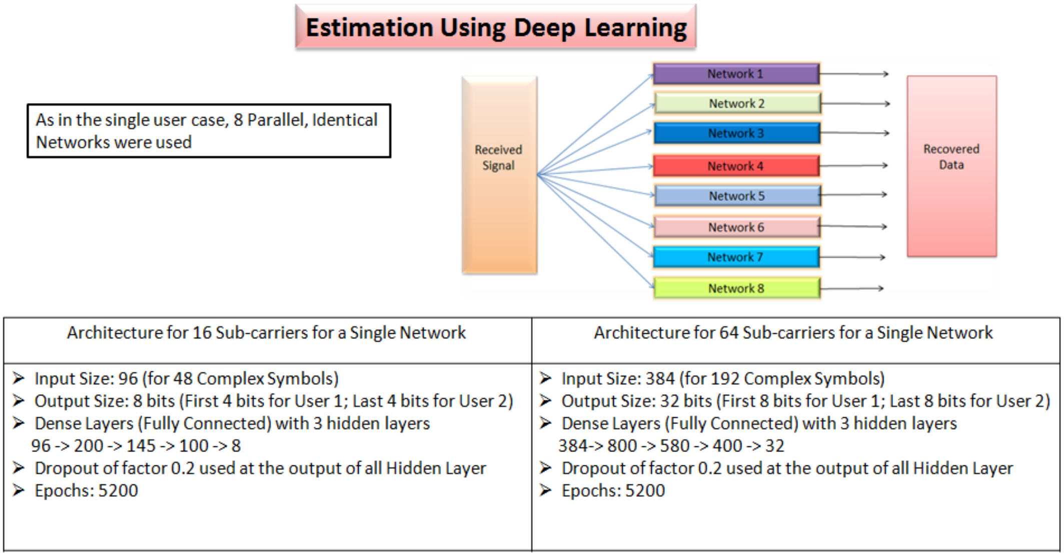
Similarly,

$$\hat{X}_2 = \arg \max_{x_2 \in QPSK} \sum_{x_1 \in QPSK} e^{-\frac{|Y - H_1 x_1 - H_2 x_2|^2}{2\sigma^2}}$$

To get a tighter theoretical limit on the BER, we also used separately the actual CSI (stored in a csv file) rather than the CSI estimate obtained from the pilots.
As far as the Deep Learning architecture is concerned, we investigated several models including the auto-encoders. However, the architecture that gave us the best results was merely a modification of the Dense Neural Network used in the single user case. We carried out our experiments for 16- Subcarrier systems and 64-Subcarrier Systems, architecture for which is shown in figure 4.4

4.1.2 Application in Bioinformatics

For this application, we first need to understand the structure of the data, and then we will move onto devising architectures for End-to-End Deep Learning.



Architecture for 16 Sub-carriers for a Single Network	Architecture for 64 Sub-carriers for a Single Network
➢ Input Size: 96 (for 48 Complex Symbols) ➢ Output Size: 8 bits (First 4 bits for User 1; Last 4 bits for User 2) ➢ Dense Layers (Fully Connected) with 3 hidden layers 96 -> 200 -> 145 -> 100 -> 8 ➢ Dropout of factor 0.2 used at the output of all Hidden Layer ➢ Epochs: 5200	➢ Input Size: 384 (for 192 Complex Symbols) ➢ Output Size: 32 bits (First 8 bits for User 1; Last 8 bits for User 2) ➢ Dense Layers (Fully Connected) with 3 hidden layers 384-> 800 -> 580 -> 400 -> 32 ➢ Dropout of factor 0.2 used at the output of all Hidden Layer ➢ Epochs: 5200

FIGURE 4.4: Deep Learning architectures used for NOMA systems

4.1.2.1 Struture of mPower Data

The mPower data [3] is consists of four types of activities: Walking, Voice, Tapping and Memory. Figure 4.5 illustrates the structure of mPower data. For our purposes, we limit our attention to Voice data. These four types of activities are specifically designed to trigger Parkinson's symptoms response. Each activity is recorded via the following processes:

- **Walking**: Subjects walk 10 steps with a phone in their pocket. Phone's Gyroscopic measurements are recorded during the whole activity. This activity is intended to capture irregularities in walking if any.
- **Voice**: Subjects record 10 seconds of voice into the microphone of their cellphone. This data is intended to capture the jitter or shimmer in the voice if any.
- **Tapping**: Subjects tap on two points on the phone repeatedly. The tapping positions and intervals are recorded. This data is intended to record tremor in hand motions if any.
- **Memory**: Subjects play a guessing game. In this game, the results of pattern matching are recorded. This activity is intended to capture if the memory is in good shape since one of the symptoms of Parkinson's is loss of memory.

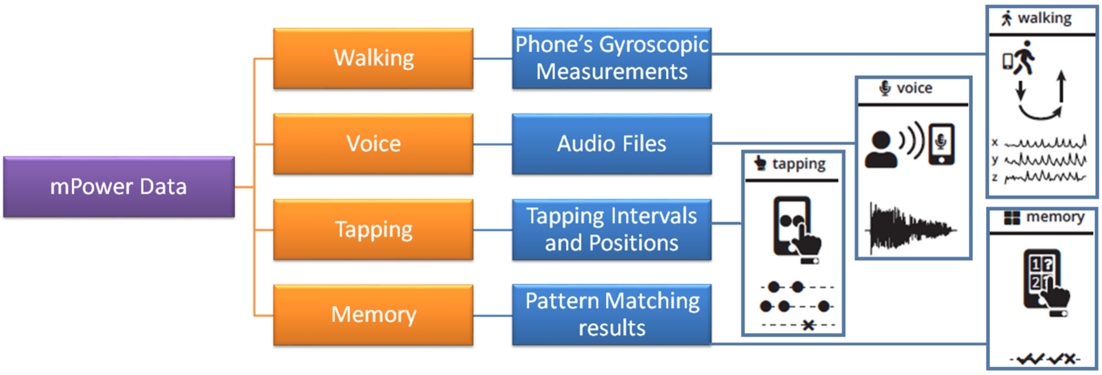


FIGURE 4.5: The Structure of mPower Data

4.1.2.2 Structure of Voice Data

The voice data contains 65,022 audio files in total which are organized on the basis of the time of their recording as illustrated by Figure 4.6. We ignored the recordings of Parkinson's patients done just after medication, as we believe that those recordings would not provide information about Parkinson's symptoms as Parkinson's medications, suppress these symptoms [8].

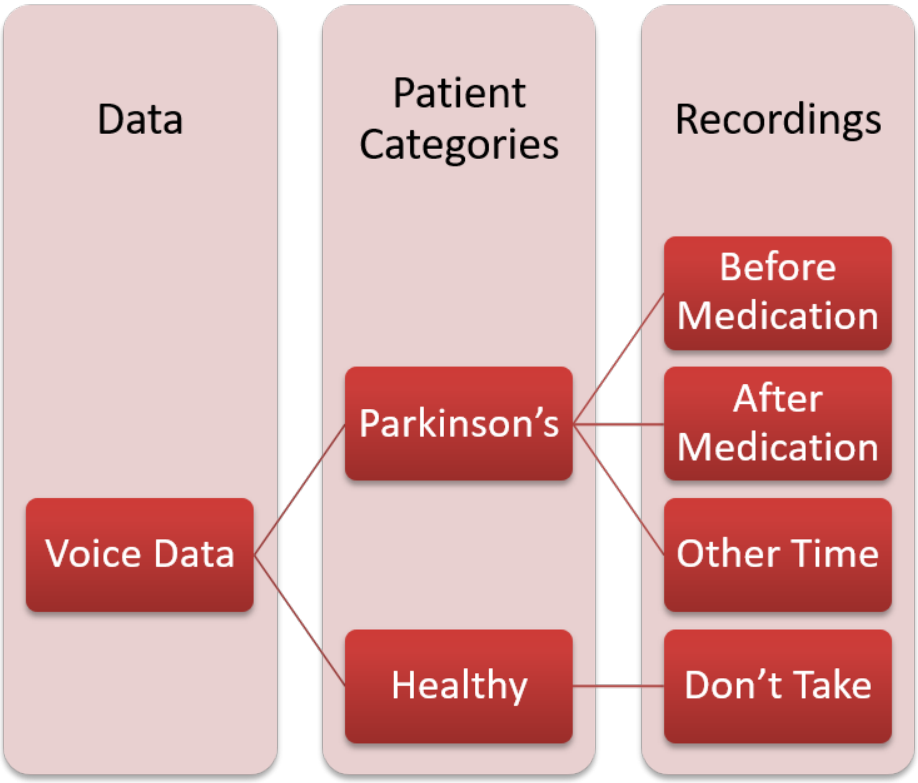


FIGURE 4.6: Different types of recordings in Voice Data

However, the data available to us is imbalanced in terms of the number of people participating in the study. We observe that only 20% of the people are clinically diagnosed by Parkinson's Disease, while an overwhelming 58% of the recordings are attributed to that 20

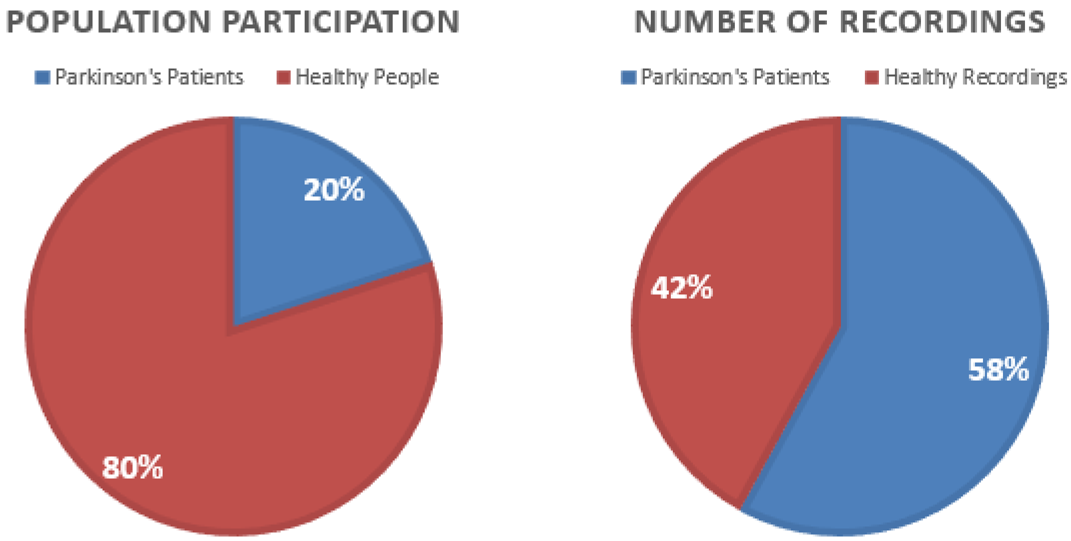


FIGURE 4.7: The Vioce Data Distribution according to PD indication

4.1.2.3 Initial Investigation with Machine Learning Models

Initially, we assumed every recording sample independent of each other and designed a recording-level classifier. The pipeline designed for such a classifier is illustrated in Figure 4.8. The following features were used for training these classifiers:

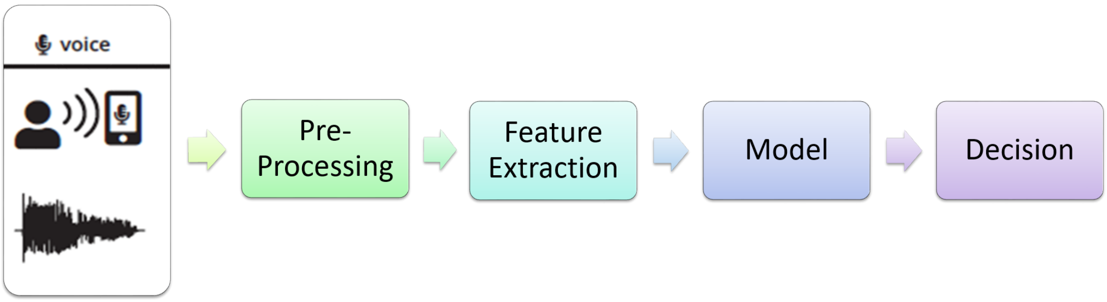


FIGURE 4.8: Initial Pipleline Designed for a Recording-level Classifier

- Detrended fluctuation analysis [13]
- Mean Teager-Kaiser energy operator [13]
- Mel-Frequency Cepstal Co-efficients [13]

By training these classifiers, we were able to achieve an Area Under Curve measure of 0.88 as illustrated in Figure 4.9, but our assumptions were wrong are there was high correlation present between two audio-files recorded by the same patient and treating them as independent was misleading.

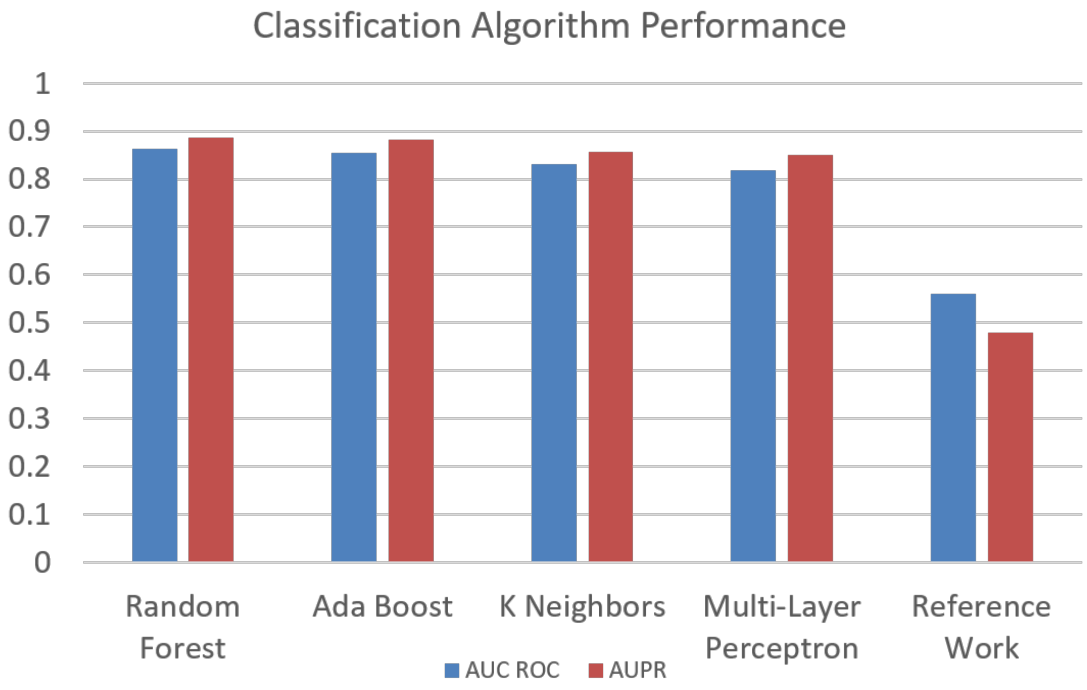


Figure 4.9: Results of Machine Learning Algorithms with Incorrect Assumption of Independence

4.1.2.4 Selection of a Fixed Number of Recordings per Person

After the correction of our assumption of independence, we limited all the recordings from one particular person to either train set or the test set. Hence, a high correlation between two recordings does not affect our performance measures as all of those recordings are located entirely on one side of the split. It turns out when we limit all the recordings by one patient into either test or train dataset; the metrics drop rapidly.

The contribution of recordings by different people is very asymmetric. Some people have too many recordings, some too few. This causes a bias when training a recording level classifier as well as a person level classifier. As shown in Figure 4.10, we see that the mode number of recordings is 3. Almost 31% of the people have done 3 recordings, and the mean number of recordings is 11.57. Therefore, we devise a policy of selecting a fixed number (let's say k) of recordings per person. This is done for the following reasons:

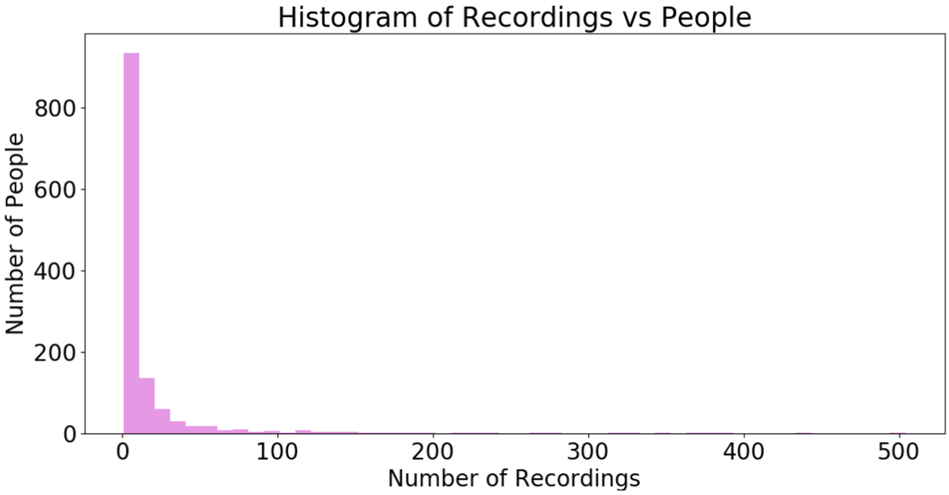


FIGURE 4.10: Asymmetry in the Number of Recordings done by Different People

- People with lesser number of recordings become under-represented and provide us very less information.
- A large number of recordings from the same person provides redundant information as the correction between two recordings by the same person is very high. This costs us in the form of un-required computation.

Therefore, we implemented this selection and tested our performance measure by selecting 2,3,4 and 5 number of recordings per person for each architecture.

4.1.2.5 Spline Convolutional Neural Network Architecture

We use a deep neural network architecture [4] to classify the recordings because this architecture was designed to detect and learn features for audible data. The detailed structure for such an architecture is given in Figure 4.11.

The spline CNNs are a special form of regular CNNs with the change that convolutional filters are initialized as band-limited filters with several center frequencies covering the whole Frequency-space. The CNN is called spline CNN because Hermite-cubic splines are used for filter construction. The filters are constructed by first designing 1 ‘mother-filter’ in Frequency Domain and then shifting it to cover the whole Frequency range. In our case, we use a filter of 100 Hz Bandwidth and shift it with a frequency of ∼10 Hz to make 200 band-limited filters. The sampling frequency we use is 2205 Hz. These 200 filters are initialized as convolutional filters in our network and then learned as we train the network. For illustration, two sample filters out of the 200 filters have been shown in Figure 4.12

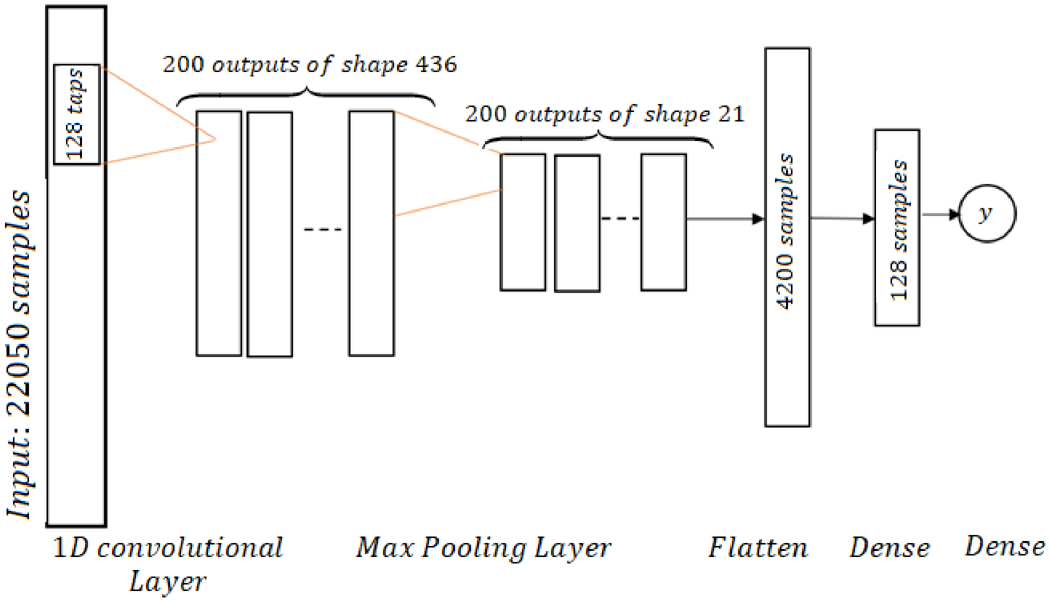


FIGURE 4.11: The Architecture of Spline Convolutional Neural Network

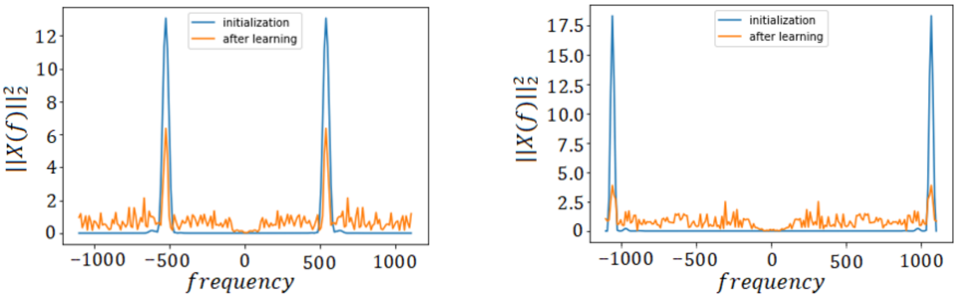


FIGURE 4.12: Sample filters for Spline-CNN

4.1.2.6 Evidence Aggregation Model (EAM)

We introduced another Deep Learning Model into our pipeline to output a final prediction for each person based upon their predictions from the recording-based classifier. This updates our pipeline to the one shown in Figure 4.13 EAM is implemented as a Deep Bi-directional LSTM network, as shown in Figure 4.14. The work of EAM was to aggregate results from more than one models as we move towards other activities as well. However, for this setting, where we are only working with the voice data, the EAM converges to mode function.

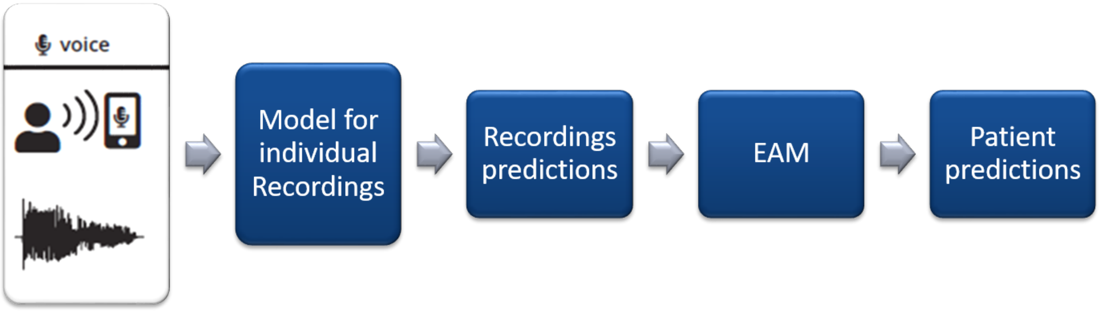


Figure 4.13: New PipeLine Model after adding Evidence Aggregation Model

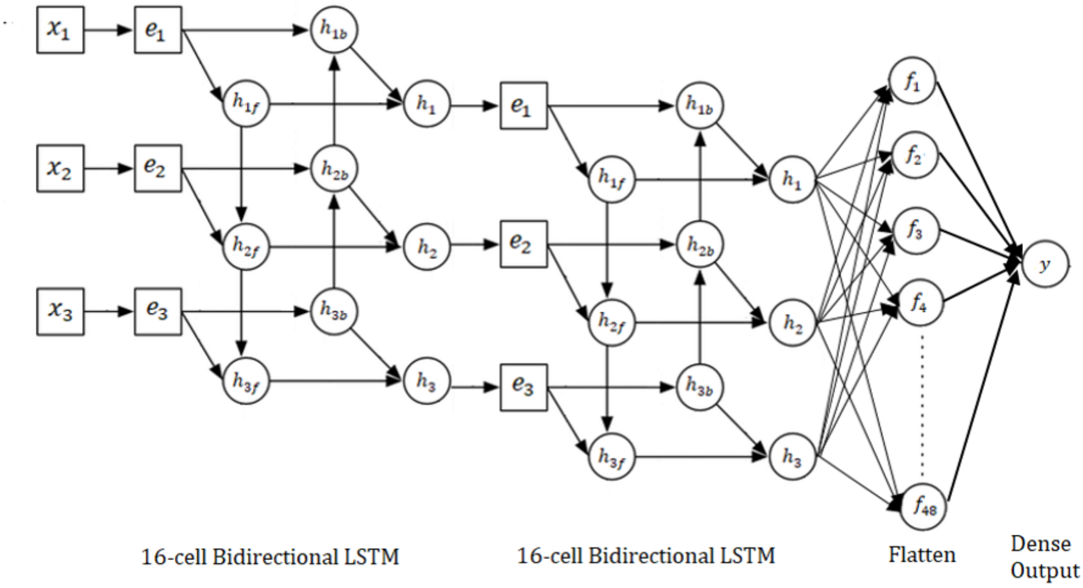


Figure 4.14: Structure of Evidence Aggregation Model

4.2 Simulation Results

4.2.1 Application in Communication Systems

4.2.1.1 OFDM Systems

Figure 4.15 shows results for a single tap channel with Additive White Gaussian Noise (AWGN). The training data consisted of 50,000 messages while testing data consisted of 10,000 messages. The channel was unchanging across frames.
We can see that Least Squares (LS) estimation fares the worst while time domain processing fares better since it utilizes the knowledge of number of channel taps

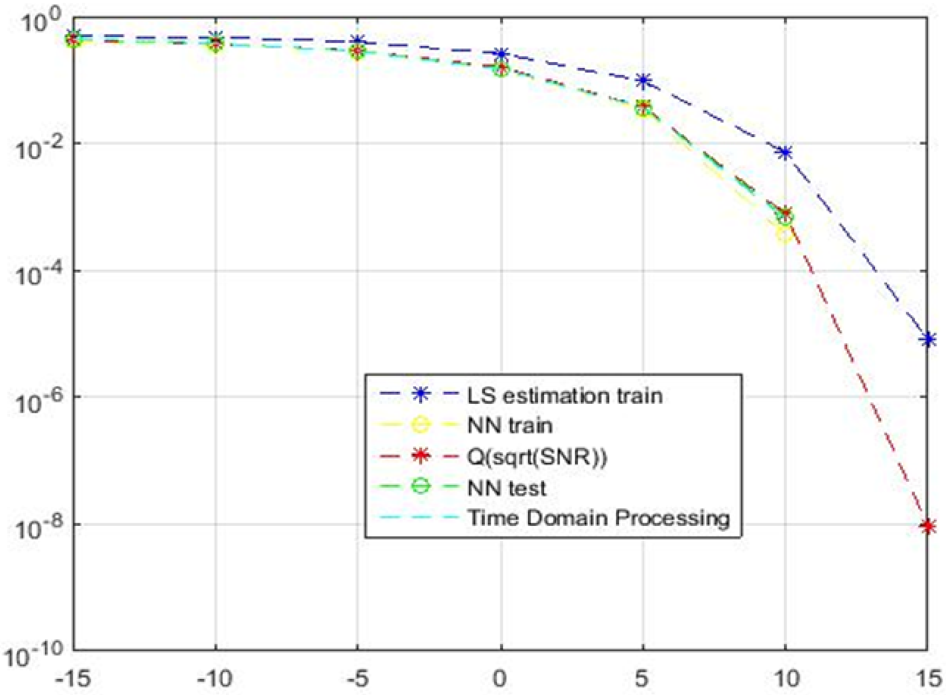


FIGURE 4.15: Results for AWGN Channel

used. We see that Neural Network performed the best and performed at par with Theoretical limit given by:

$$Q(\sqrt{10^{\frac{SNR(dB)}{10}}})$$

Figure 4.16 shows results for 8 tap real channel drawn from a Gaussian distribution. Again, the training data consisted of 50,000 messages while testing data consisted of 10,000 messages. The channel was unchanging across frames. We again see that Neural Network performed the best.

Figure 4.17 shows results for 8-tap real channel drawn from a Gaussian distribution. As before, the training data consisted of 50,000 messages while testing data consisted of 10,000 messages. This time, the channel is changed every frame so that we have 50,000 instances of the channel. This has led to degradation in the performance of the Neural Network causing it to lag behind theoretical methods. One possible explanation could be that the neural network does not have enough examples to learn the distribution channels are drawn from.

Figure 4.18 shows results for 8 tap complex, exponentially fading channel. Both real and imaginary parts are drawn from a Gaussian distribution and have equal energies. The training data consisted of 50,000 messages while testing data consisted of 10,000 messages. Also, the channel is changed every frame so that we

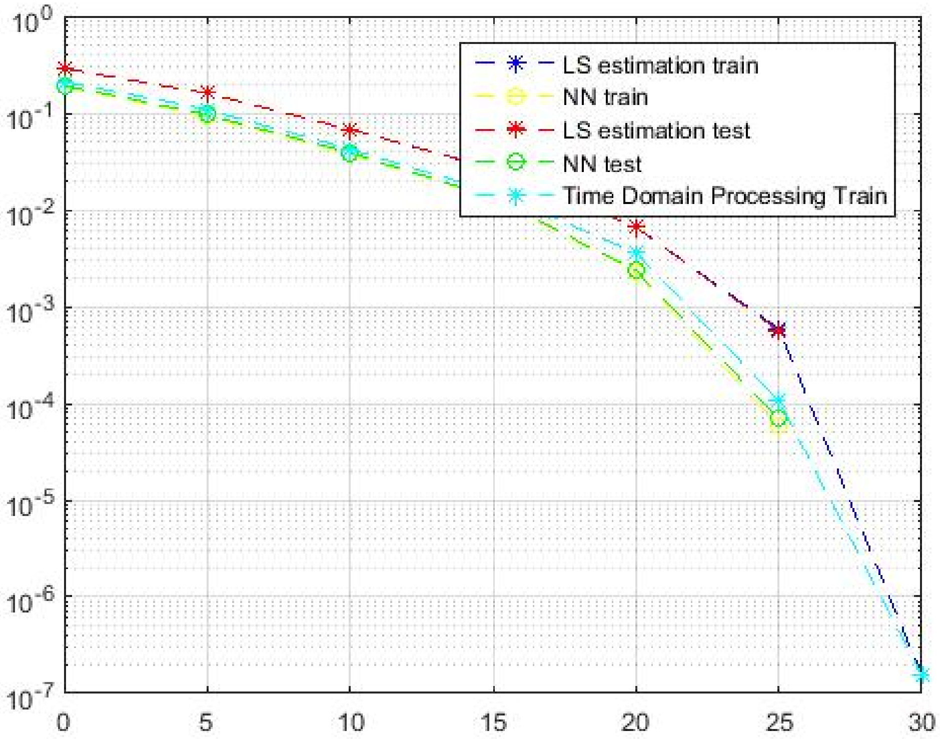


FIGURE 4.16: Results for 8 tap real channel which is unchanging across frames

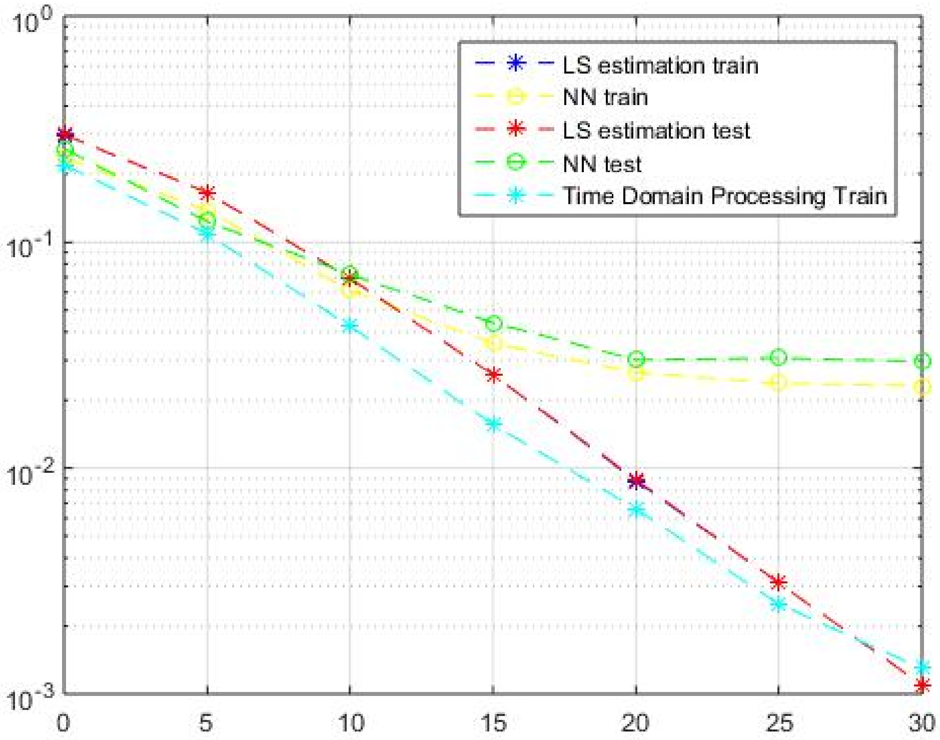


FIGURE 4.17: Results for 8-tap real channel which is changing across frames

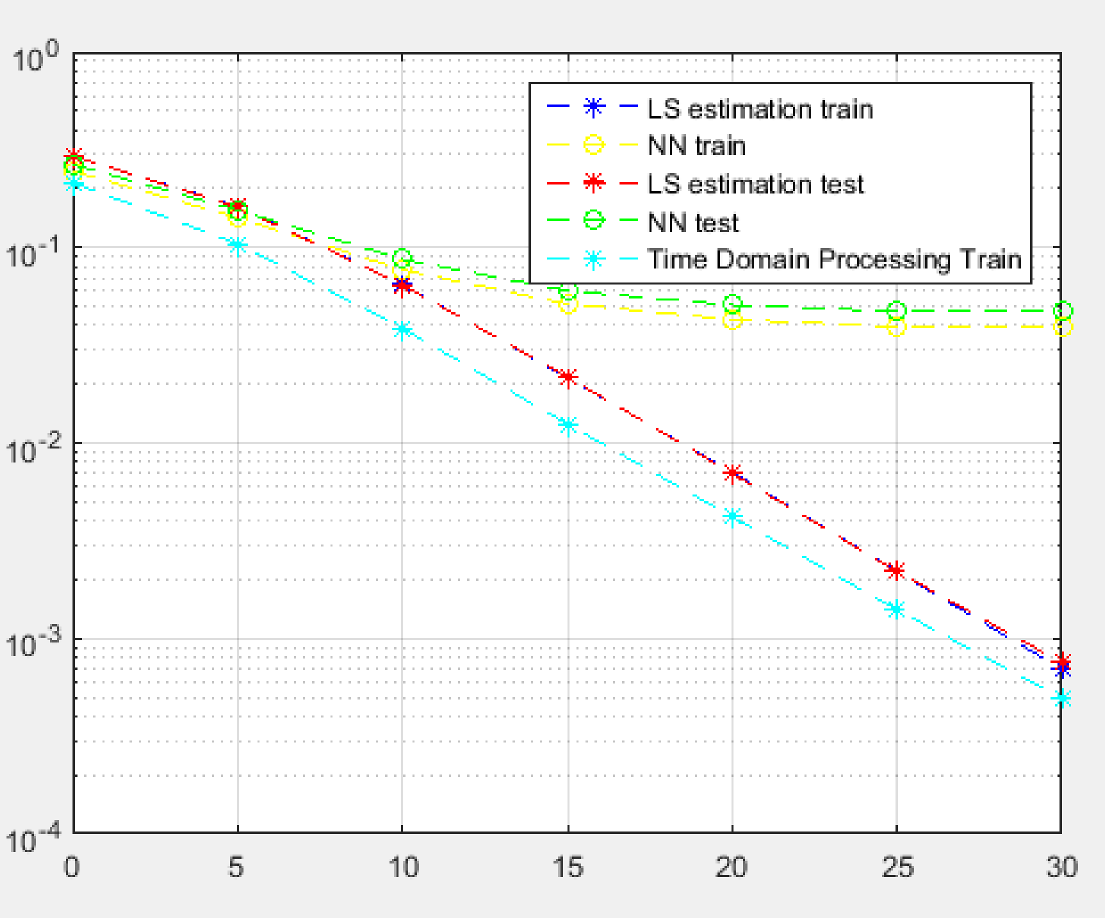


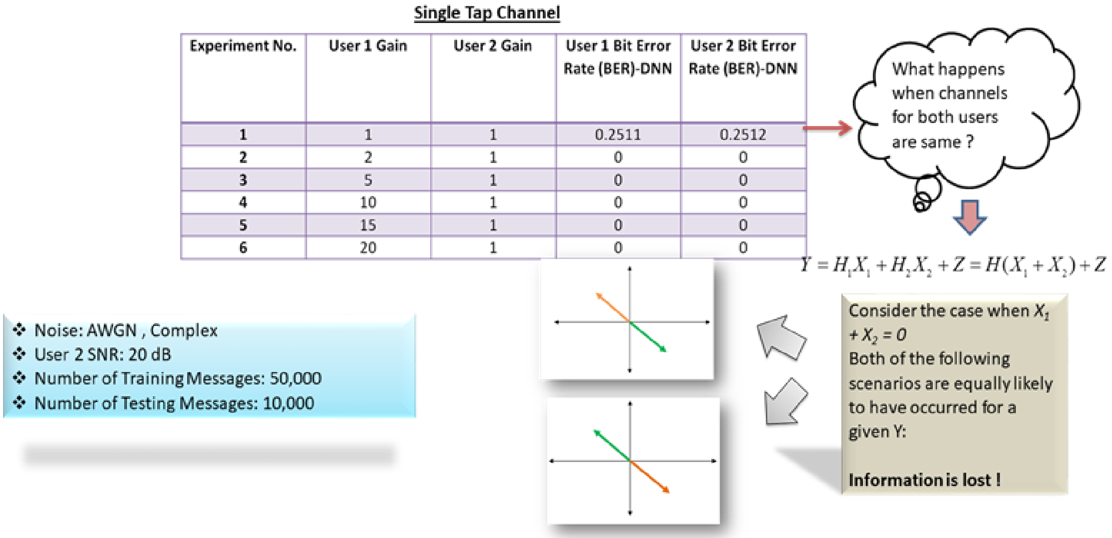
FIGURE 4.18: Results for 8 tap complex channel which is changing across frames

have 50,000 instances of the channel. The performance of Neural Network deteriorates while the performance LS estimation and Time domain processing improves slightly.

4.2.1.2 NOMA systems

Since we now have to do a joint estimation of both users' data the complexity of the problem has increased. Consequently, we had to use more neurons per layer, and we had to increase the number of epochs by a factor of 52. Therefore, we considered it prudent to carry out most of our experiments for 16 subcarrier systems and then verify them for 64 subcarrier systems.

Table 4.1 shows results for a Single tap channel. The noise added was complex. The Signal to Noise ratio (SNR) for user 2 was maintained at 20 dB. The gain for user 1 was varied which was akin to changing its SNR. The number of training messages used was 50,000 while the number of testing messages used was 10,000. As we can see here, the neural network achieved perfect decoding of the received data. However, it was observed that when the channel for both users was same,



Single Tap Channel

Experiment No.	User 1 Gain	User 2 Gain	User 1 Bit Error Rate (BER)-DNN	User 2 Bit Error Rate (BER)-DNN
1	1	1	0.2511	0.2512
2	2	1	0	0
3	5	1	0	0
4	10	1	0	0
5	15	1	0	0
6	20	1	0	0

What happens when channels for both users are same ?

$$Y = H_1X_1 + H_2X_2 + Z = H(X_1 + X_2) + Z$$

Consider the case when $X_1 + X_2 = 0$
Both of the following scenarios are equally likely to have occurred for a given Y:

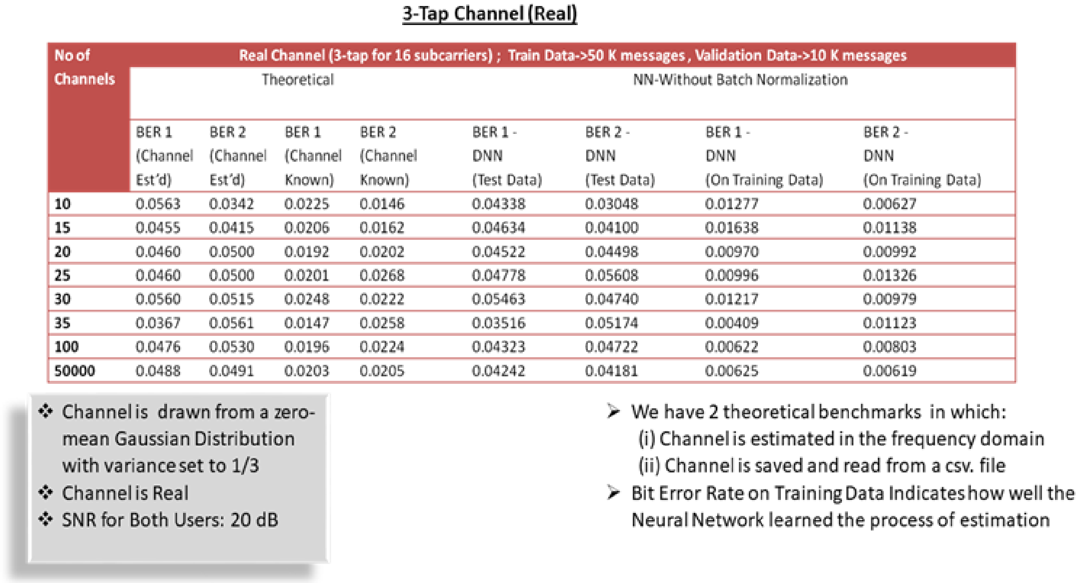
Information is lost !

- Noise: AWGN , Complex
- User 2 SNR: 20 dB
- Number of Training Messages: 50,000
- Number of Testing Messages: 10,000

Table 4.1: Results for Single tap Channel (NOMA)

the Neural Network failed because QPSK modulation uses sets of symbols that are 180^o apart in phase, that cancel out resulting in a loss of information.

Table 4.2 and Figure 4.19 demonstrate results when the number of Channel taps



3-Tap Channel (Real)

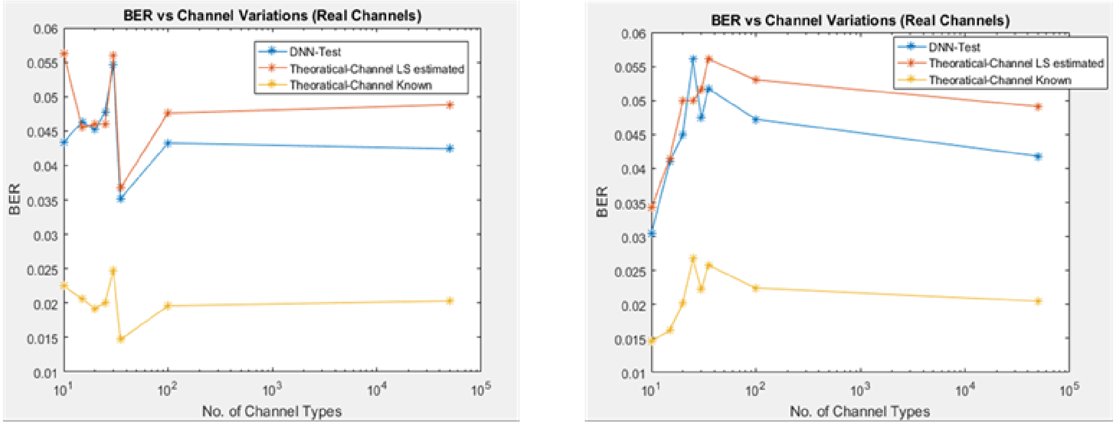
No of Channels	Real Channel (3-tap for 16 subcarriers) ; Train Data->50 K messages , Validation Data->10 K messages							
	Theoretical				NN-Without Batch Normalization			
	BER 1 (Channel Est'd)	BER 2 (Channel Est'd)	BER 1 (Channel Known)	BER 2 (Channel Known)	BER 1 - DNN (Test Data)	BER 2 - DNN (Test Data)	BER 1 - DNN (On Training Data)	BER 2 - DNN (On Training Data)
10	0.0563	0.0342	0.0225	0.0146	0.04338	0.03048	0.01277	0.00627
15	0.0455	0.0415	0.0206	0.0162	0.04634	0.04100	0.01638	0.01138
20	0.0460	0.0500	0.0192	0.0202	0.04522	0.04498	0.00970	0.00992
25	0.0460	0.0500	0.0201	0.0268	0.04778	0.05608	0.00996	0.01326
30	0.0560	0.0515	0.0248	0.0222	0.05463	0.04740	0.01217	0.00979
35	0.0367	0.0561	0.0147	0.0258	0.03516	0.05174	0.00409	0.01123
100	0.0476	0.0530	0.0196	0.0224	0.04323	0.04722	0.00622	0.00803
50000	0.0488	0.0491	0.0203	0.0205	0.04242	0.04181	0.00625	0.00619

- Channel is drawn from a zero-mean Gaussian Distribution with variance set to 1/3
- Channel is Real
- SNR for Both Users: 20 dB

- We have 2 theoretical benchmarks in which:
  (i) Channel is estimated in the frequency domain
  (ii) Channel is saved and read from a csv. file
- Bit Error Rate on Training Data Indicates how well the Neural Network learned the process of estimation

Table 4.2: Results for 3-tap Real Channel (NOMA)

is increased to 3. The Channel is kept real and drawn from a zero-mean Gaussian distribution. This time, the SNR for both of the users is maintained at 20 dB. We used 2 theoretical benchmarks to gauge the performance of the Neural Network. The first method uses Least Squares Estimation in the frequency domain to estimate the channel. In the second method, the channel is saved and loaded from a



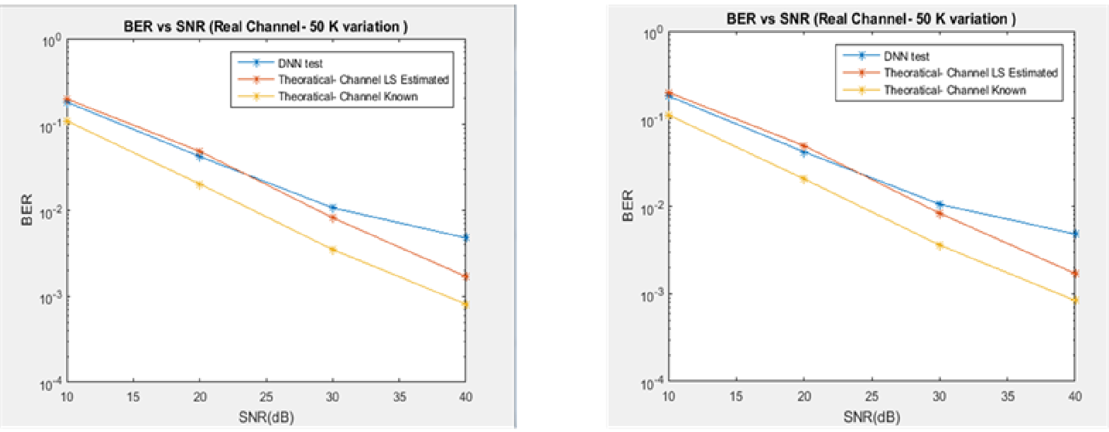
Plots for User 1

Plots for User 2

FIGURE 4.19: Plots of BER for both users for 3-tap Real Channel against No. of Channel (NOMA)

.csv file and used to recover the data. Since the major hurdle in the data recovery is channel estimation, this method gives us the theoretical limit below which the bit error rate cannot possibly go. We investigated the BER against the increase in Number of Channel types or instances. The plots show that the neural network performs slightly better than the theoretical method that uses Least squares estimation of the channel. Also, we see that contrary to the case of single user detection; the neural network has also managed to learn the channel distribution as the performance of the neural network does not degrade as we increase the number of channel types. Similarly, figure 4.20 below shows that BER decreases with increasing SNR.

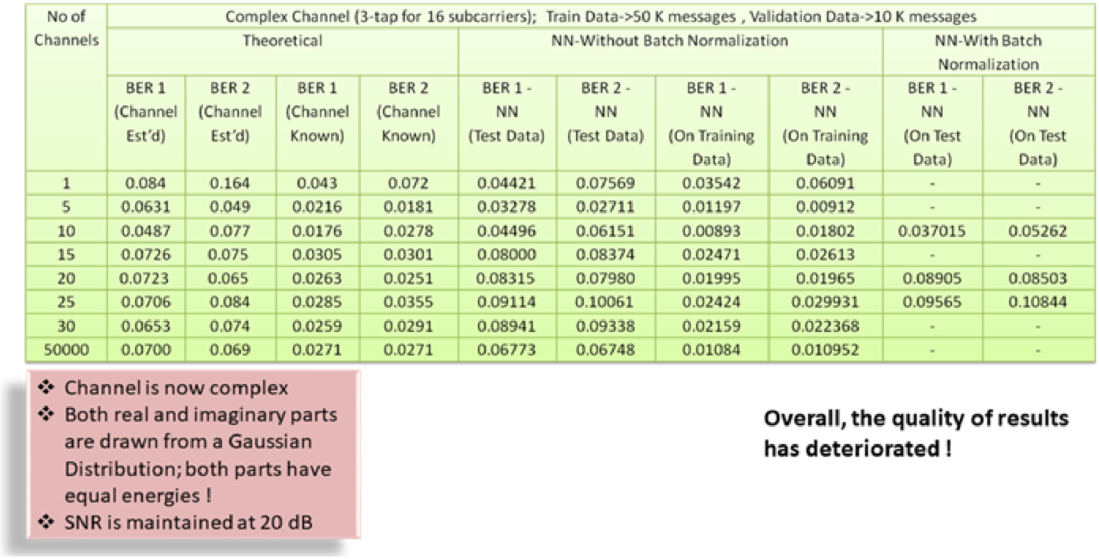
Table 4.3 and Figure 4.21 demonstrate what happens when we make the channel



Plots for User 1

Plots for User 2

FIGURE 4.20: Plots of BER for both users for 3-tap Real Channel against SNR (NOMA)



No of Channels	Complex Channel (3-tap for 16 subcarriers); Train Data->50 K messages , Validation Data->10 K messages									
	Theoretical				NN-Without Batch Normalization				NN-With Batch Normalization	
	BER 1 (Channel Est'd)	BER 2 (Channel Est'd)	BER 1 (Channel Known)	BER 2 (Channel Known)	BER 1 - NN (Test Data)	BER 2 - NN (Test Data)	BER 1 - NN (On Training Data)	BER 2 - NN (On Training Data)	BER 1 - NN (On Test Data)	BER 2 - NN (On Test Data)
1	0.084	0.164	0.043	0.072	0.04421	0.07569	0.03542	0.06091	-	-
5	0.0631	0.049	0.0216	0.0181	0.03278	0.02711	0.01197	0.00912	-	-
10	0.0487	0.077	0.0176	0.0278	0.04496	0.06151	0.00893	0.01802	0.037015	0.05262
15	0.0726	0.075	0.0305	0.0301	0.08000	0.08374	0.02471	0.02613	-	-
20	0.0723	0.065	0.0263	0.0251	0.08315	0.07980	0.01995	0.01965	0.08905	0.08503
25	0.0706	0.084	0.0285	0.0355	0.09114	0.10061	0.02424	0.029931	0.09565	0.10844
30	0.0653	0.074	0.0259	0.0291	0.08941	0.09338	0.02159	0.022368	-	-
50000	0.0700	0.069	0.0271	0.0271	0.06773	0.06748	0.01084	0.010952	-	-

- Channel is now complex
- Both real and imaginary parts are drawn from a Gaussian Distribution; both parts have equal energies !
- SNR is maintained at 20 dB

Overall, the quality of results has deteriorated !

Table 4.3: Results for 3-tap Complex Channel (NOMA)

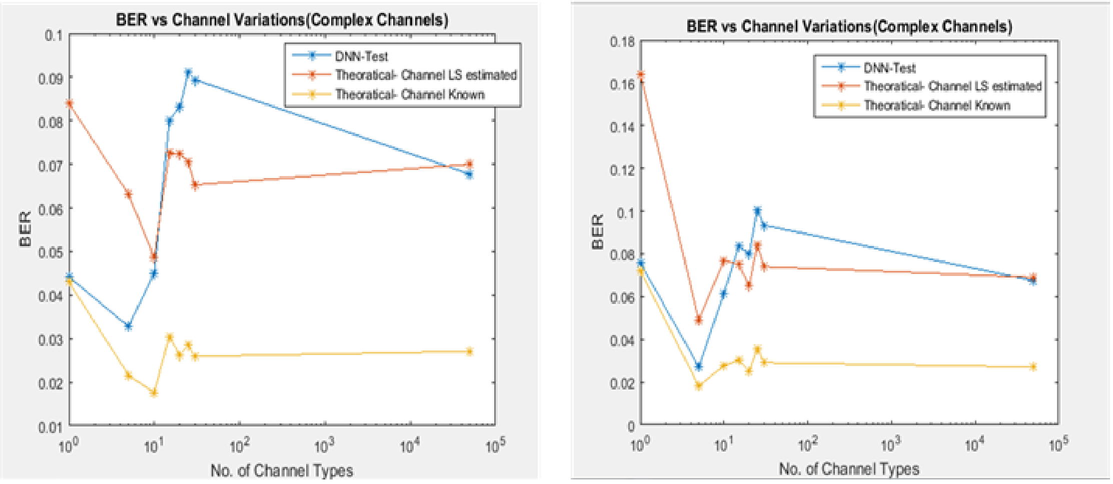
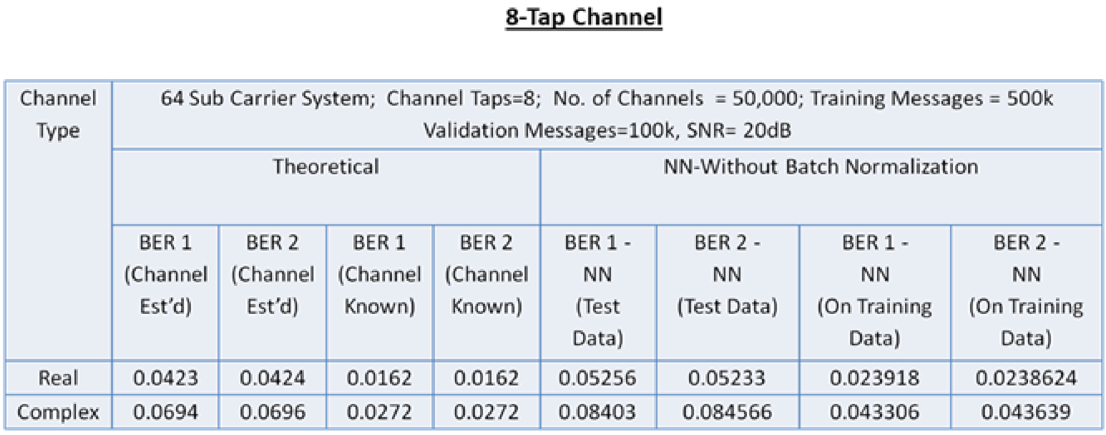


Figure 4.21: Plots of BER for both users for 3-tap Complex Channel against No. of Channel (NOMA)

complex. Both the real and imaginary parts of the channel have been drawn from a Gaussian distribution, and the energies of both parts are kept equal. We can see that the bit error rates have gone up and the overall quality of results have deteriorated. However, the Neural Network still performs at par with the theoretical method that estimates the channel using the Least Squares.

To confirm the robustness of our results, we extend our experiments to 64 subcarrier system, the results of which are shown in Table 4.4. Since the Number of training, messages have been increased 10 folds, and the neurons in each layer have also been increased, the training took longer than the training in the 16-subcarrier system. Results show that the Neural Network performs slightly poorer than the theoretical methods.



8-Tap Channel

Channel Type	64 Sub Carrier System; Channel Taps=8; No. of Channels = 50,000; Training Messages = 500k Validation Messages=100k, SNR= 20dB							
	Theoretical				NN-Without Batch Normalization			
	BER 1 (Channel Est'd)	BER 2 (Channel Est'd)	BER 1 (Channel Known)	BER 2 (Channel Known)	BER 1 - NN (Test Data)	BER 2 - NN (Test Data)	BER 1 - NN (On Training Data)	BER 2 - NN (On Training Data)
Real	0.0423	0.0424	0.0162	0.0162	0.05256	0.05233	0.023918	0.0238624
Complex	0.0694	0.0696	0.0272	0.0272	0.08403	0.084566	0.043306	0.043639

Table 4.4: Results for an 8-tap complex channel varying across frames for 64 sub-carrier system

4.2.2 Application in Bioinformatics

We compare our results to the state-of-the-art work on the same dataset by P.Schwab [8] who have used Traditional Machine Learning techniques to classify between Parkinson's patients and Healthy persons.

4.2.2.1 Results using Random Forest

After the correction of our independence assumption, we were able to improve upon the reported result by using more robust hand-crafted features as shown in Figure 4.22. Using Random Forest with EAM, improved our results to an AUC of 0.76 in comparison of 0.56 obtained by P.Schwab [8] on Voice Data as shown in Figure 4.23

4.2.2.2 Results using Spline CNN

The next logical step was to replace recording-level with a CNN to skip the pre-processing and feature Extraction step. This improved our measure to an AUC ROC of 0.89 with EAM for 3 recordings per patient as shown in Figure 4.24

4.2.2.3 Results Using Different Values of Recordings per Person

The number of recordings per person were also changed and the results were recorded. As we can see that this change does not affect the performance very significantly,yet maximum performance was obtained at 3 recordings per person as shown in Figure 4.25. The data composition was also changed when we changed the number of recordings per person which is illustrated in Figure 4.26

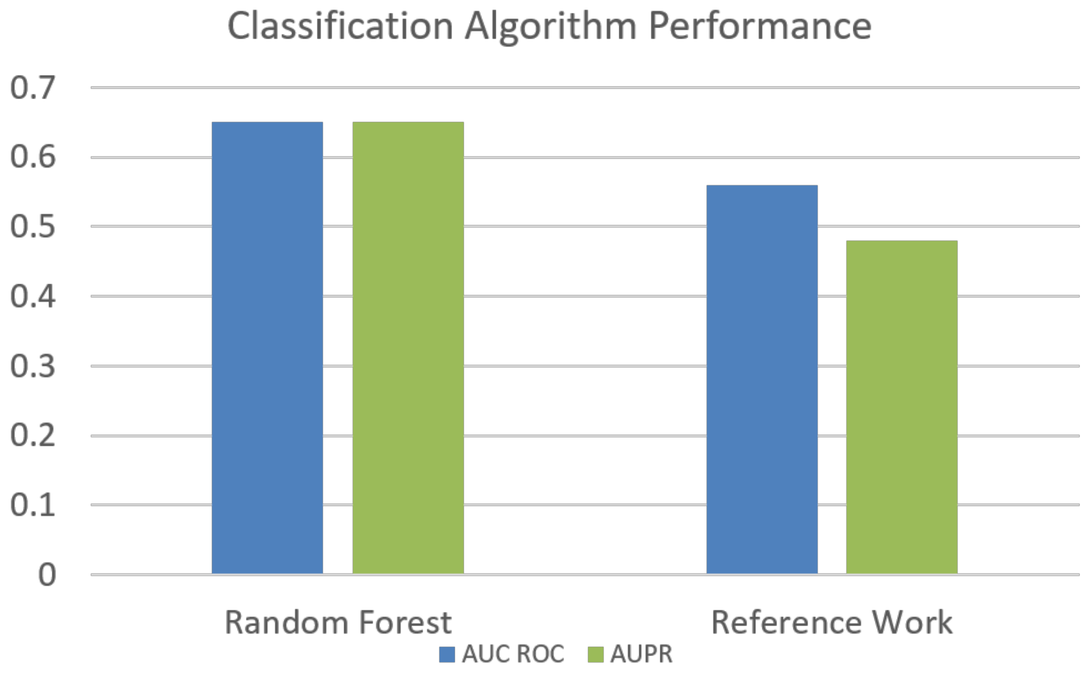


Figure 4.22: Results from a simple Random Forest Classifier

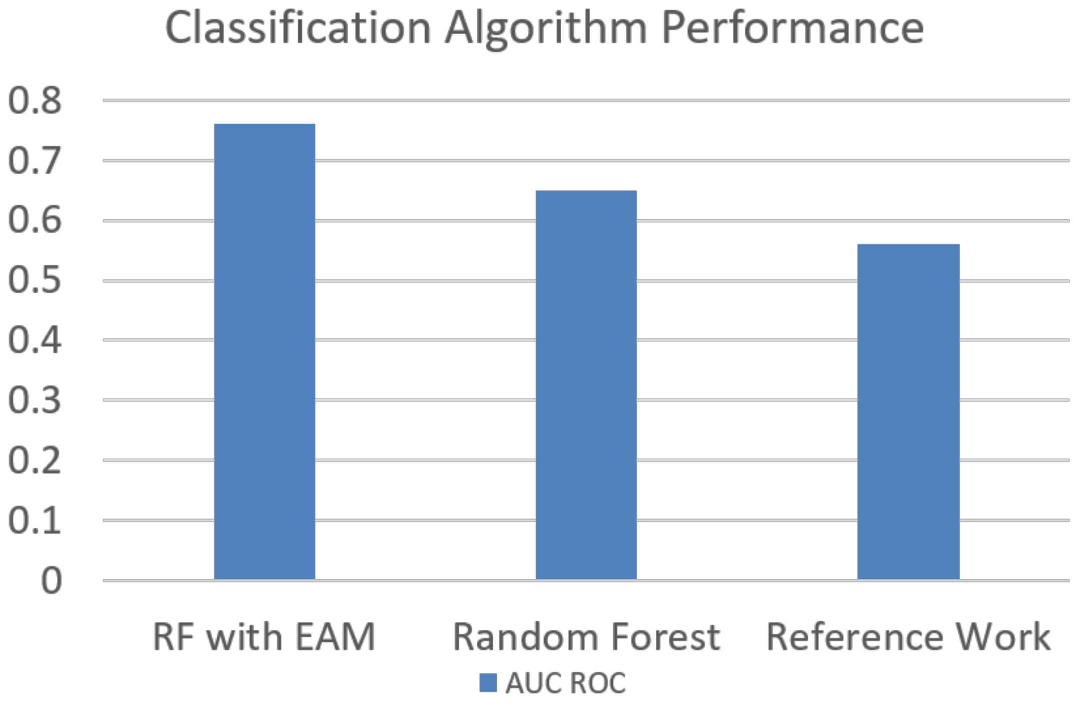


Figure 4.23: Results from a simple Random Forest Classifier Alongwith EAM

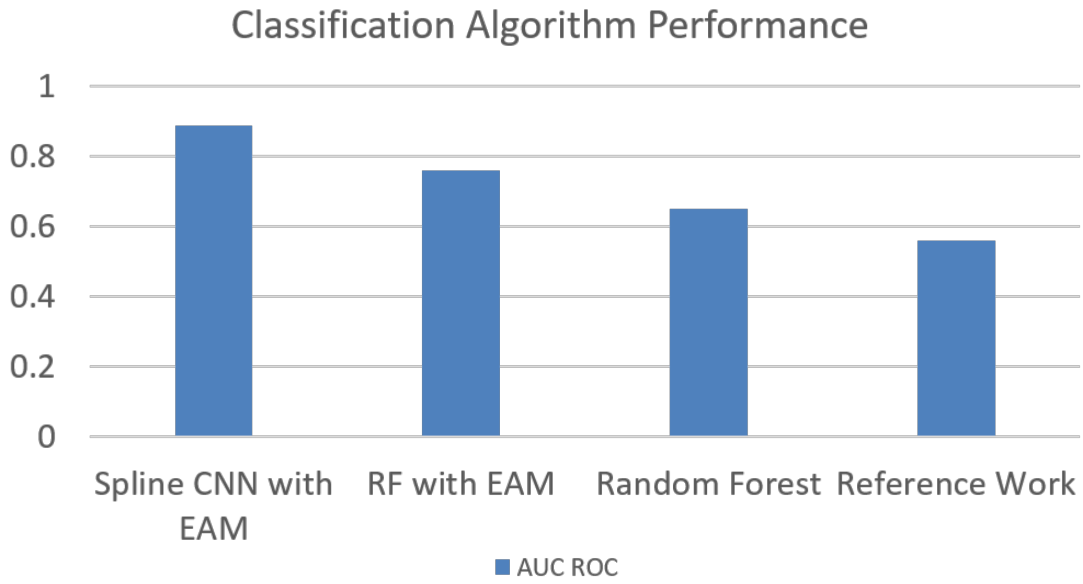


FIGURE 4.24: Results from a Spline CNN Alongwith EAM

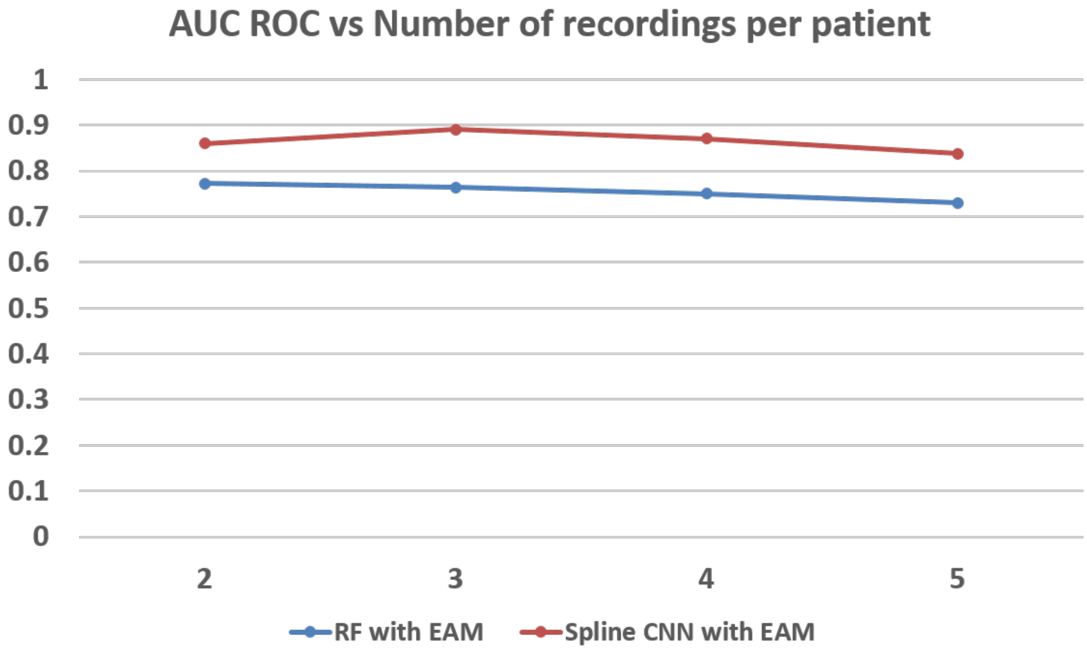


FIGURE 4.25: Change in Performace with Respect to Number of Recordings per Person

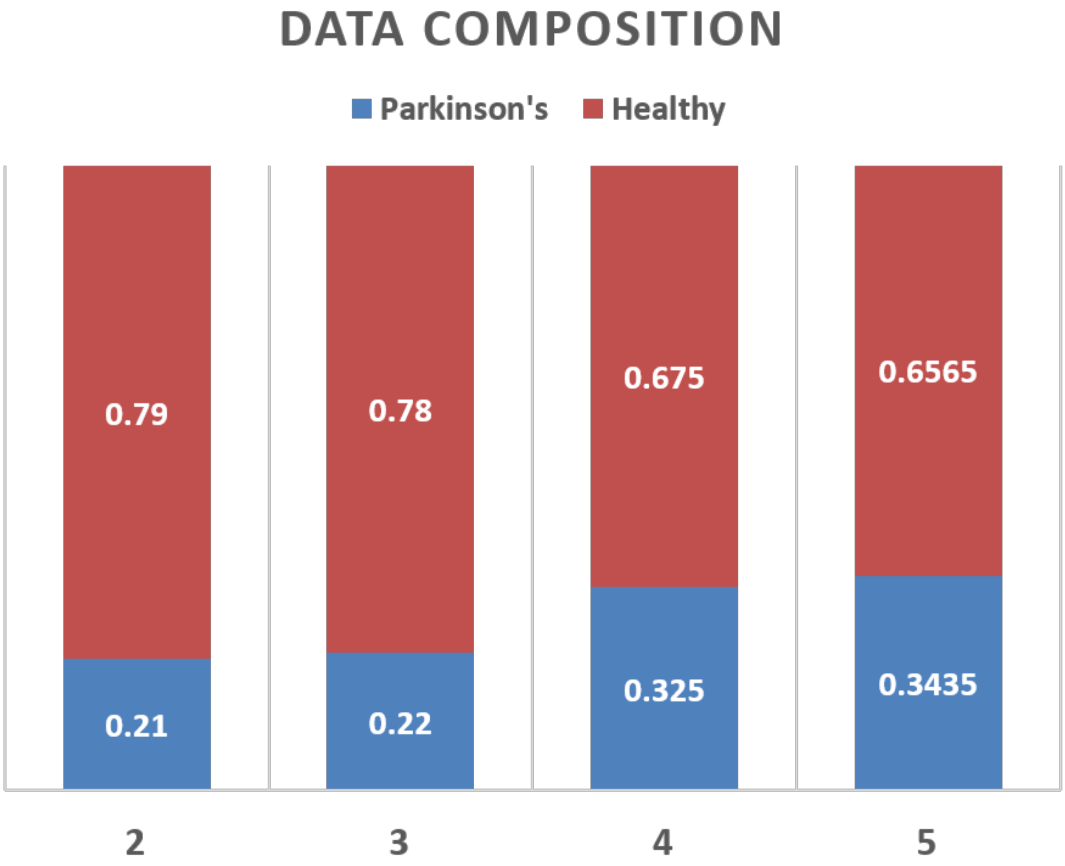


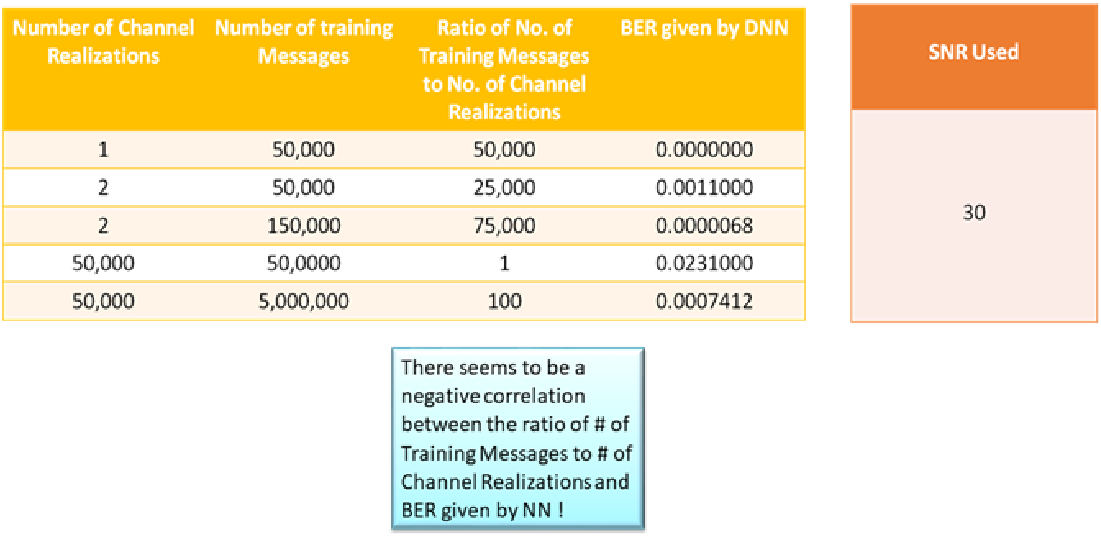
FIGURE 4.26: Data Composition as we change the number of recordings per patient

4.3 Result Analysis And Outlook

4.3.1 Application in Communication Systems

The major stumbling block identified by our results is the need for training data. Our generalization error is made up of 2 components: Bias Error and Variance Error. While bias error can be decreased by making the model more complex, variance error cannot be decreased without increasing the amount of training data. For single user detection, when the channel was unvarying across frames, our neural network did not need to learn the channel distribution, and therefore a small number of training messages sufficed. However, as we increased the number of channel instances, the DNN needed to learn the underlying distribution of the channel and consequently required a large amount of training data. That is why for an unchanging, channel, our DNN performed the best, while for a changing channel it performed the worst. Table 4.5 shows that as we increase the amount of training data, the performance of the neural network starts to improve.

Similarly, Table 4.6 convincingly suggests at lower amounts training data, the performance of the neural network is at par with the theoretical method that estimates the channel with Least Squares. However if we increase the number of training messages, the performance of the DNN improves, and it tries to approach the theoretical limit calculated by using the exact channel response which had been saved and loaded from a .csv file.

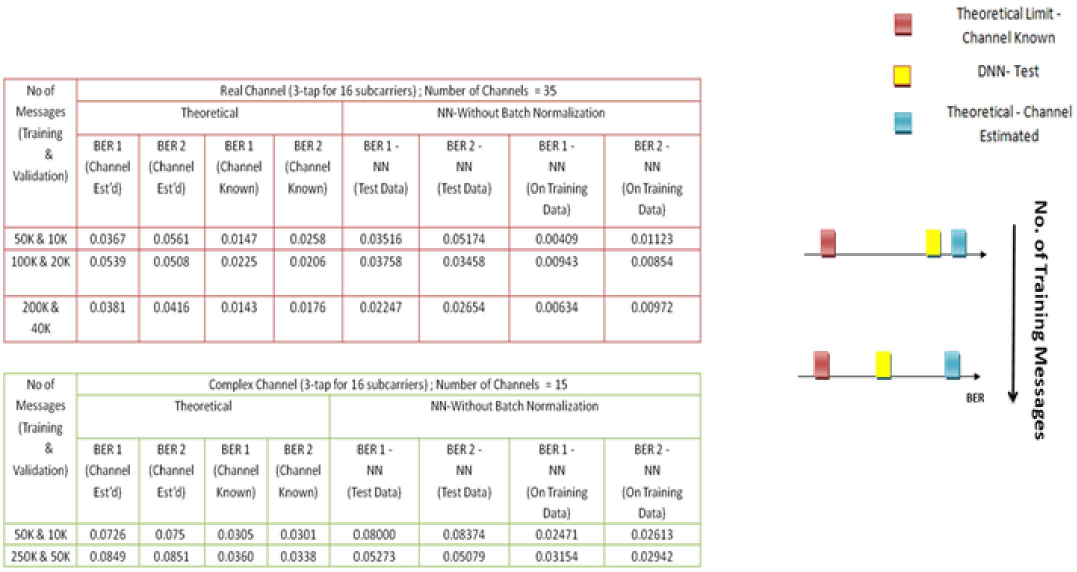


Number of Channel Realizations	Number of training Messages	Ratio of No. of Training Messages to No. of Channel Realizations	BER given by DNN
1	50,000	50,000	0.0000000
2	50,000	25,000	0.0011000
2	150,000	75,000	0.0000068
50,000	50,0000	1	0.0231000
50,000	5,000,000	100	0.0007412

SNR Used
30

There seems to be a negative correlation between the ratio of # of Training Messages to # of Channel Realizations and BER given by NN !

Table 4.5: Effect of increasing the Training Data for Single User Detection



No of Messages (Training & Validation)	Real Channel (3-tap for 16 subcarriers) ; Number of Channels = 35							
	Theoretical				NN-Without Batch Normalization			
	BER 1 (Channel Est'd)	BER 2 (Channel Est'd)	BER 1 (Channel Known)	BER 2 (Channel Known)	BER 1 - NN (Test Data)	BER 2 - NN (Test Data)	BER 1 - NN (On Training Data)	BER 2 - NN (On Training Data)
50K & 10K	0.0367	0.0561	0.0147	0.0258	0.03516	0.05174	0.00409	0.01123
100K & 20K	0.0539	0.0508	0.0225	0.0206	0.03758	0.03458	0.00943	0.00854
200K & 40K	0.0381	0.0416	0.0143	0.0176	0.02247	0.02654	0.00634	0.00972

No of Messages (Training & Validation)	Complex Channel (3-tap for 16 subcarriers) ; Number of Channels = 15							
	Theoretical				NN-Without Batch Normalization			
	BER 1 (Channel Est'd)	BER 2 (Channel Est'd)	BER 1 (Channel Known)	BER 2 (Channel Known)	BER 1 - NN (Test Data)	BER 2 - NN (Test Data)	BER 1 - NN (On Training Data)	BER 2 - NN (On Training Data)
50K & 10K	0.0726	0.075	0.0305	0.0301	0.08000	0.08374	0.02471	0.02613
250K & 50K	0.0849	0.0851	0.0360	0.0338	0.05273	0.05079	0.03154	0.02942

Theoretical Limit - Channel Known
DNN- Test
Theoretical - Channel Estimated
No. of Training Messages
BER

Table 4.6: Effect of increasing the Training Data for Multi-User Detection

4.3.2 Application in Bioinformatics

The major takeaway from our results is the importance of End-to-End Deep Learning as we have been able to improve quite a lot on the previous results by using the Spline-CNN. The main advantage of Spline-CNN is that it does not need us to compute convolutional filters, but instead learns those filters based on data itself; we just initialize the filters as band-limited filters with the help of splines, and the rest is learned by the Network itself.

However, for End-to-End Deep Learning, here, we also face the same problem of the requirement of large amounts of data. This trend can be observed in Figure 4.25 because as we increase the number of recordings per person, the results show a trend of decreasing performance. This is due to the decreasing amount of data we have available when we increase the number of recordings per person as there are fewer people who have recorded that many number of recordings or more. We can observe this trend in the number of recordings histogram in Figure 4.10.

Chapter 5

Cost Analysis, Conclusion, and Future Recommendations

5.1 Application in Communication Systems

One of the major drawbacks of using the Neural Networks is that when the applications require a large amount of training data, the power and the specifications of the hardware required to go up exponentially. The more complexity we add to our model of the communication system, the more complicated becomes the architecture required. For example, in multi-user detection, when we moved from 16 subcarrier system to a more standard 64 sub-carrier systems, we had to increase the number of neurons by a factor of 4, and we had to increase the amount of training data by a factor of 10. The training required High-Performance Computing, and despite that, it took at least a week. One positive outcome of this is that out of 2128 (≈ 1038) possible training messages which could be fed to our 64 subcarrier system, we at most require a very small subset of about tens of million distinct messages. This means that our network indeed learns the inverse of the transformation which mapped the bits to the received signal. Also, in terms of cost, big companies that are working in the field of communication do have powerful hardware at their disposal, which means that they will be able to train the network without the cost of any new hardware installation or purchase. One area of concern may be the power consumption of the hardware while training the network. But we must remember that we have to train the network only once to obtain the optimal weights of the model, which we can copy to and store in millions of devices. In other words, once we have trained the network, we can use a microcontroller or a chipset of negligible cost and power to do inference.

Nonetheless, it would be pertinent to mention that the system model our network uses is very rudimentary and simple. The neural network is trained at a constant SNR; the noise is AWGN, and the channel is drawn from a single Gaussian distribution; the bits are generated randomly too, and so the resultant symbols in the frame don't have any covariance between them (On second thought, things could have been less complicated as covariance between symbols would have allowed us to use LSTM network who have a superior performance for sequence-2-sequence transformation.) To train our neural network for practical use, we could use more realistic channel models such as WINNER II. Also, we could change the SNR gradually but continuously to mimic real-time use of communication systems. Moreover, instead of generating the bits randomly, we could use data bits obtained from video, audio, or other documents to make the network more accurate.

One area that has the potential for investigation is to decrease in pilot length. For Single-User Detection, when the pilot length is small, the performance of the neural network degrades less than that of traditional methods [5]. Unfortunately, we failed to observe this trend for multi-user detection. Our future work aims to find improvements in the architecture, and find the optimal amount of training data so that we obtain this trend in the multi-user detection as well.

5.2 Application in Bioinformatics

Our systems may have performed exceptionally well in terms of Area Under Receiver Operating Curve (AUC) as illustrated in Figure 4.24, however, if we look at the sensitivity and specificity measures of the data as defined below:

$$Sensitivity = \frac{TP}{TP + FN}$$

$$Specificity = \frac{TN}{TN + FP}$$

In our context:

- TP = True Positives = Number of Parkinson's Patients predicted as Parkinson's Patients
- TN = True Negatives = Number of Healthy People predicted as Healthy People
- FP = False Positives = Number of Parkinson's Patients predicted as Healthy People

- FN = False Negatives = Number of Healthy People predicted as Parkinson's Patients

Using this measure, we obtain the results illustrated in Figure 5.1. We see that they are not performing very well as the maximum sensitivity we were able to obtain is 0.41 which is not a very good result for providing a very good measure for providing Parkinson's predictions. We believe that this is due to the following reasons:

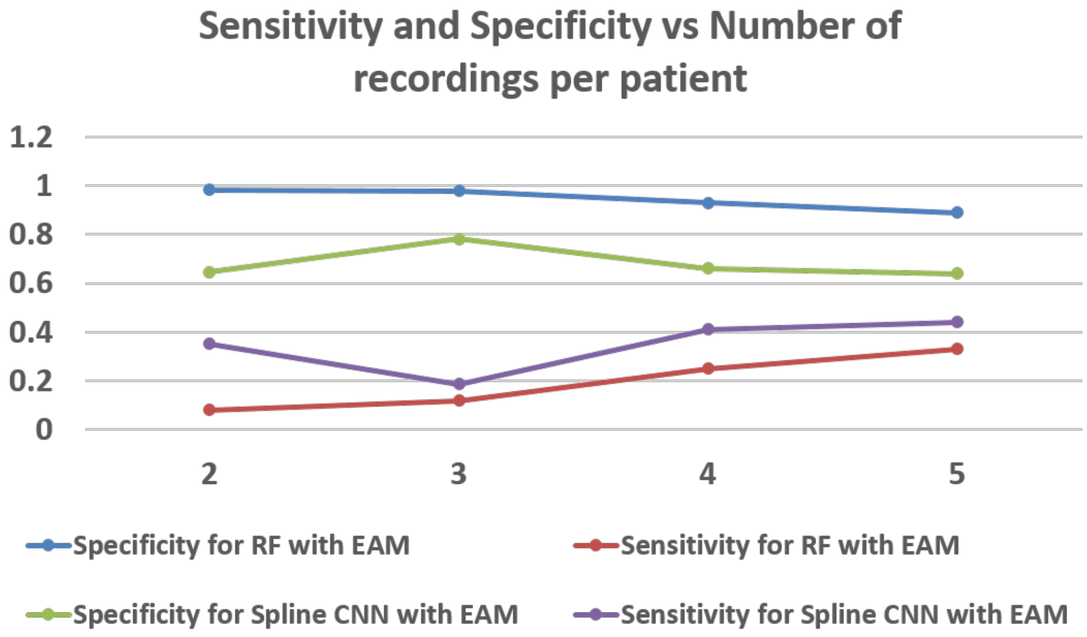


FIGURE 5.1: Change in Sensitivity and Specificity as we change the number of recordings per person

- The data is highly imbalanced and stays so even when we change the recordings per patient as illustrated in Figure 4.26.
- When we increase the number of recordings per patient, it seems to become less imbalanced, but at the same time, we have a lesser amount of data available, as there are fewer people who have recorded that many number of recordings or more. We can observe this trend in the number of recordings histogram in Figure 4.10.

We aim to fight this trend either by optimizing our models for more sensitivity or by adding other modes (types) of activities data in the architecture as well. This way, the other activities data can help augment our predictions and improve them giving us a more robust model in terms of sensitivity and specificity as well.

References

[1] G. P. D. S. G. S. Committee, "Factors impacting on quality of life in parkinson's disease: results from an international survey," *Movement Disorders*, vol. 17, no. 1, pp. 60–67, 2002.

[2] R. Pahwa and K. E. Lyons, "Early diagnosis of parkinson's disease: recommendations from diagnostic clinical guidelines," *Am J Manag Care*, vol. 16, no. 4, pp. 94–99, 2010.

[3] B. M. Bot, C. Suver, E. C. Neto, M. Kellen, A. Klein, C. Bare, M. Doerr, A. Pratap, J. Wilbanks, E. R. Dorsey *et al.*, "The mpower study, parkinson disease mobile data collected using researchkit," *Scientific data*, vol. 3, p. 160011, 2016.

[4] R. Balestriero, R. Cosentino, H. Glotin, and R. Baraniuk, "Spline filters for end-to-end deep learning," in *International Conference on Machine Learning*, 2018, pp. 373–382.

[5] H. Ye, G. Y. Li, and B.-H. Juang, "Power of deep learning for channel estimation and signal detection in ofdm systems," *IEEE Wireless Communications Letters*, vol. 7, no. 1, pp. 114–117, 2018.

[6] A. Felix, S. Cammerer, S. Dörner, J. Hoydis, and S. Ten Brink, "Ofdm-autoencoder for end-to-end learning of communications systems," in *2018 IEEE 19th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*. IEEE, 2018, pp. 1–5.

[7] H. Ye, G. Y. Li, B.-H. F. Juang, and K. Sivanesan, "Channel agnostic end-to-end learning based communication systems with conditional gan," in *2018 IEEE Globecom Workshops (GC Wkshps)*. IEEE, 2018, pp. 1–5.

[8] P. Schwab and W. Karlen, "Phonemd: Learning to diagnose parkinson's disease from smartphone data," *arXiv preprint arXiv:1810.01485*, 2018.

[9] B. Zheng, S. W. Yoon, and S. S. Lam, “Breast cancer diagnosis based on feature extraction using a hybrid of k-means and support vector machine algorithms,” *Expert Systems with Applications*, vol. 41, no. 4, pp. 1476–1482, 2014.

[10] J. J. Oresko, Z. Jin, J. Cheng, S. Huang, Y. Sun, H. Duschl, and A. C. Cheng, “A wearable smartphone-based platform for real-time cardiovascular disease detection via electrocardiogram processing,” *IEEE Transactions on Information Technology in Biomedicine*, vol. 14, no. 3, pp. 734–740, 2010.

[11] A. Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau, and S. Thrun, “Dermatologist-level classification of skin cancer with deep neural networks,” *Nature*, vol. 542, no. 7639, p. 115, 2017.

[12] Y. Suhara, Y. Xu, and A. Pentland, “Deepmood: Forecasting depressed mood based on self-reported histories via recurrent neural networks,” in *Proceedings of the 26th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 2017, pp. 715–724.

[13] S. Arora, V. Venkataraman, A. Zhan, S. Donohue, K. M. Biglan, E. R. Dorsey, and M. A. Little, “Detecting and monitoring the symptoms of parkinson’s disease using smartphones: a pilot study,” *Parkinsonism & related disorders*, vol. 21, no. 6, pp. 650–653, 2015.