QATAR UNIVERSITY

COLLEGE OF ENGINEERING

OPTIMAL RESOURCE ALLOCATION USING DEEP LEARNING-BASED ADAPTIVE COMPRESSION FOR MHEALTH APPLICATIONS

BY

ABEER ZIAD ALMARRIDI

A Thesis Submitted to

the Faculty of the College of

Engineering

in Partial Fulfillment

of the Requirements

for the Degree of

Masters of Science in Computing

January 2018

COMMITTEE PAGE

The members of the committee approve the thesis of Abeer Ziad AlMarridi defended on 12/11/2017.

Dr. Amr Mahmoud Salem Mohamed
Thesis/Dissertation Supervisor

Dr. Aiman Mahmood Erbad
Thesis/Dissertation Co-Supervisor

Prof. Abdallah Khreishah
Committee Member

Dr. Tarek El-Fouly
Committee Member

Approved:

Khalifa Al-Khalifa, Dean, College of Engineering

ABSTRACT

ALMARRIDI, ABEER, ZIAD, Masters: January : [2018], Masters of Science in Computing

Title: Optimal resource allocation using Deep learning-based adaptive compression for mHealth applications.

Supervisor of Thesis: Dr. Amr Mahmoud Mohamed

Co-supervisor of Thesis: Dr. Aiman Erbad

In the last few years the number of patients with chronic diseases that require constant monitoring increases rapidly; which motivates the researchers to develop scalable remote health applications. Nevertheless, transmitting big real-time data through a dynamic network limited by the bandwidth, end-to-end delay and transmission energy; will be an obstacle against having an efficient transmission of the data. The problem can be resolved by applying data reduction techniques on the vital signs at the transmitter side and reconstructing the data at the receiver side (i.e. the m-Health center). However, a new problem will be introduced which is the ability to receive the vital signs at the server side with an acceptable distortion rate (i.e. deformation of vital signs because of inefficient data reduction).

In this thesis, we integrate efficient data reduction with wireless networking to deliver an adaptive compression with an acceptable distortion, while reacting to the wireless network dynamics such as channel fading and user mobility. A Deep Learning (DL) approach was used to implement an adaptive compression technique to compress and reconstruct the vital signs in general and specifically the Electroencephalogram Signal (EEG) with the minimum distortion. Then, a resource allocation framework was introduced to minimize the transmission energy along with the distortion of the reconstructed signal

while considering different network and applications constraints such as the available bandwidth, distortion threshold, data rate threshold and the end to end delay.

Thereafter, this thesis evaluates the performance of using Convolutional Autoencoder (CAE) as Deep learning approach in compressing and reconstructing vital signs (i.e. EEG signals). The results show that using CAE approach in compression provides efficient distortion rate while maximizing compression ratio. However, learning makes CAE application-specific, where each CAE model is designed specifically for certain application (i.e. dataset). In some applications, CAE may provide maximum levels of compression with an acceptable distortion, eliminating the need for optimal network resource allocation, which simplifies the network layer. In other cases, network resource optimization will still be applied to complement vital sign compression and address the trade-off between the transmission energy and distortion of the reconstructed signal with respect to the network and application requirements.

Moreover, the results of the resources allocation optimization problem for multiple users illustrate that using CAE technique for compressing and reconstructing the data, while considering the bandwidth as a decision parameter, will minimize the transmission energy of the data compared with using constant equally assigned bandwidth among all users. Comparison between the results of the resource allocation using CAE and Discrete wavelet transforms (DWT)was also captured, where CAE outperforms DWT as it minimizes both the distortion and the transmission energy efficiently.

DEDICATION

To my great parents, for their endless love and support

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to my supervisor Dr. Amr Mohamed, for giving me a chance to work on such an important topic. His continued guidance, important suggestion, affectionate encouragement, critical comments and correction of the thesis are acknowledged.

Besides my supervisor, I would like to thank his research group, Eng. Ahmad Ben Said and Eng. Alaa Awad for their continuous help and availability to answer my questions and discuss the status of my work progress.

Thanks for Dr. Aiman Erbad and all my doctors and professors who taught me during my master's studies, it was a hard period filled with many academic and social challenges, I learned a lot and still, I need more as the learning phase never ends.

Furthermore, I would like to acknowledge my colleagues during this journey, Imene Mecheter, Racheal Fernandez, Yousra Regaya and Zeineb Safi for all their help, support and valuable hints.

Finally, I would like to acknowledge with gratitude, the continuous support, encouragement, and love of my family- my parents, my sisters, Alaa, Wafa, and Dody; my brother, Ahmad and my cute niece Joud.

This work was made possible by NPRP grant # 7 - 684 - 1 - 127 from the Qatar National Research Fund (a member of Qatar Foundation). The findings achieved herein are solely the responsibility of the authors.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

PUBLICATIONS

The following is the list of papers that will be published in this thesis:

I. Abeer Ziad Al-Marridi, Amr Mohamed, and Aiman Erbad, Convolutional Autoencoder Approach for EEG compression and reconstruction in m-Health Systems, submitted to International Wireless Communications and Mobile Computing Conference (IWCMC 2018), Cyprus, 2018.

II. Abeer Ziad Al-Marridi, Amr Mohamed and Aiman Erbad, Survey: Recent Data Reduction and Compression Techniques applied to Medical Data, (under preparation)

III. Abeer Ziad Al-Marridi, Amr Mohamed, and Aiman Erbad, Efficient EEG Mobile Edge Computing and Optimal Resource Allocation for Smart Health Applications, (under preparation)

CHAPTER 1: INTRODUCTION

This chapter discusses m-Health systems and their importance. It includes a motivation behind processing data at the edge level, before transmitting through the wireless network. This chapter also explains the main objectives of the thesis and shows the thesis organization.

1.1 Motivation

Nowadays, the spending on health care increases rapidly and it is considered as a top priority worldwide. The reason behind that is the rising number of different diseases. The United States spends around $2.6 trillion for health care, as it is the highest spending on healthcare which expected to be doubled by 2023 [1]. Lately, the number of the patients who need continuous monitoring increases swiftly, and the physical contact with the caregiver is essential; but it causes a burden for both the patients and the doctors. This limits the one-to-one relationship between the patient and the doctor, posing a real challenge for the scalability of the healthcare system.

One key solution to this problem is to exploit the fast developments of mobile, wearable devices and wireless technologies to build and improve the mobile-health systems (m-Health). The idea behind m-health is to use any communication device such as mobile phones, tablet computers and patient data aggregator along with wearable devices such as small sensors in/on or around the patient body to aggregate information about the human body. Various wearable devices were developed to sense the vital signs of the human such as electroencephalogram (EEG), Electrocardiography (ECG) and Electromyogram (EMG).

These wearable devices should send the data through the wireless network toward the m-Health center to control and diagnose the situation of the patient as soon as possible.

These systems minimize the probability of losing lives of patients if the data was sent at the right time to the caregiver. The delivery of such vital signs is constrained by the wireless bandwidth provisioned from the network, which in some cases causes long delays especially for vital signs that require intensive raw data to be delivered in a short time for timely diagnoses e.g. EEG, ECG, etc. For mobile devices with limited energy sources, communication energy is usually the highest source of energy consumption, causing the delivery of raw vital signs to be very energy consuming.

1.2 Problem Statement

Transmitting important big real-time data through the network will be hindered and influenced by the network and application constraints such as the following:

- The limited bandwidth of the network: Many users are using the network for different purposes and transmitting vital signs particularly consumes a huge amount of the network bandwidth; which is not applicable at any time.
- Power consumption: Devices with power constraints e.g. smartphone will not be able to manage the huge amount of raw vital signs.
- Delay: The application layer limitations on the end-to-end delay might be hard to achieve when dealing with huge amount of data and limited network resources, which will affect the efficiency of early detection.
- Movement of the patients with respect to the wireless access point will affect the quality of the network and therefore the efficiency of transmitting the data.

As a result, smart techniques are required to preprocess and compress the data at edge level (Patient Data Aggregator, e.g. smartphone), before transmitting the data, taking into consideration the network state and the application requirements such as distortion of the reconstructed vital sign signal. There are different compression techniques that can be applied to the data before transmission without considering the conditions of the network and the application. Our main focus is to verify the efficiency of compressing the vital signs data (like electroencephalogram EEG) using convolutional autoencoder to facilitate the transmission over the dynamic wireless network and reconstruct the data at the m-Health center with an acceptable distortion based on the application requirements.

1.3 Thesis Objective

The main objective of this thesis is to design an edge computing solution, leveraging Deep Learning (DL) based technique for adaptive vital sign compression, and cross-layer design for optimal resource allocation to adapt to the network state and dynamics; which change frequently.

Additionally, considering the application requirements while setting the appropriate compression rate, the compression approach should offer a high compression ratio with minimal distortion between the original and reconstructed data and transmission within an acceptable end-to-end delay.

1.4 Contributions

The contribution of this thesis can be summarized as follows:

1. Designing an adaptive neural network to process and compress the vital signs such as electroencephalogram (EEG), inside the network at the edge level based on the network state and reconstruct the data with the minimal distortion at the m-Health center based on the application requirements.

2. Applying Spatio-TEmporal Parametric Stepping (STEPS) mobility model for network resource optimization to define the locations of each patient with respect to the base station (wireless access point) during a certain duration of time.

3. Design an optimization problem to manage the trade-off between the transmission energy in the wireless environment and the distortion of the reconstructed compressed data under the following constraints:

    a. The amount of applied compression. Compression ratio
    b. The data rate of each user which should not exceed a certain threshold.
    c. The distortion of the reconstructed signal generated by each user should be less than the distortion threshold, based on the application requirements.
    d. The end to end delay deadline.
    e. The bandwidth consumed by all users should be less than the maximum bandwidth of the network.
    f. The location of the user with respect to the wireless access point.

4. Providing performance evaluation and a comparative study of both the compression approach and the network resource allocation.

1.5 Thesis Organization

The rest of the thesis is organized as follows. In Chapter 2, we briefly provide a background on Deep learning, EEG signals, as well as summarize the related work in the literature. In Chapter 3, the methodology of the proposed solution is explained. The obtained experimental results are analyzed and discussed in Chapter 4. Finally, In Chapter 5, the conclusion of the thesis and the future work. An Appendix is also provided to present the signals before compression and after being reconstructed at different compression ratios for certain dataset.

1.6 Summary

This Chapter has covered an introduction to the importance of m-Health systems, providing an enhancement to raise its efficiency by implementing an edge computing approach for adaptive data compression using Deep Learning based on the network state, before transmitting the data and reconstructing them at the m-Health center with respect to the application constraints. Problem statement, objectives, contributions and the thesis organization are also covered.

CHAPTER 2: LITERATURE REVIEW AND BACKGROUND

This Chapter addresses the recent approaches in data reduction and compression, different wireless network resource allocation methods and compression of the data over the edge network. Moreover, the Chapter includes a background about the main concepts that will be used in this thesis.

2.1 Literature Review

2.1.1 Data Reduction and Compression

Due to recent trends and technologies such as the Internet of Things (IoT), the amount of data needed to be processed, transmitted and analyzed increase rapidly. As a result, different approaches were proposed to manage big data delivery using efficient data reduction techniques for analysis, and efficient transmission over bandwidth constrained networks. Data reduction is one technique to change the representation of the data and it could be done in different ways. Many researchers proposed different methods which aim to reduce the size of the data [2,3,4,5]. Further to this, some researchers have taken into considerations the network, and application requirements and constraints such as [6].

A fuzzy data reduction technique was proposed in [6] to obtain the most representative electroencephalogram (EEG) samples and neglecting redundant ones without loss of knowledge. The authors used fuzzy Formal Concept Analysis (FCA) with Smart Sensing (SS) approach to optimize the complexity, reduce the size of the stored data and maximize the lifetime of the battery-operated devices that need to run for a long time without replacement [6].

The authors in [7] modify Block Sparse Bayesian Learning-BO BSBL-BO method to manage both linear and nonlinear dependency structure of EEG raw signal. Moreover, a vector representation was applied on multichannel signal to achieve improved block sparsity structure. The experimental results show that even in high compression ratios the algorithm produces low reconstruction error. However, the technique has high complexity, where the sensing and processing power need to be optimized in order to be used on Wireless Body Area Networks (WBAN).

Compression is one type of data reduction methods, which will be used in this work. Generally, lossless and lossy compression are two main types of data compression. Lossless compression ensures reconstructing the original data from the compressed version without distortion. However, this will add a limitation on the compression ratio; as it would be hard to apply high compression ratios. On the other hand, the lossy approach can achieve high compression ratios, but it introduces distortion on the reconstructed signal.

We will discuss different proposed compression technique using different approaches in the coming subsections.

2.1.1.1 Lossless/Near Lossless Compression Techniques

A novel near lossless compression algorithm for EEG signals was introduced in [8] to satisfy low power, latency, and low complexity by combining different state-of-the-art tools for data compression and signal processing. The encoder and decoder use the previous encoded value for prediction. The prediction is defined as a tree where the parents give the predicted value to the children and the root get its predicted value using past samples. Recursive Least Squares (RLS) algorithm was used to compute all (least-squares optimal) predictions efficiently.

The authors in [9] enhance the algorithm proposed by [8] to take into consideration the distortion threshold while looking for the best compression rates. The enhancement was by replacing RLS predictor with a multi-channel predictor that uses integer addition and multiplication. Approximating the exponential weighting was used to remove the floating-point arithmetic. The improved method reduces all the computation requirements without affecting the compression ratio. In the modified algorithm, the authors find the set of predictors empirically, where they use four main predictors to improve the cost performance at certain compression rate.

Moreover, neural networks were used for compression in [10, 11, 12, 13, 14, 15, 16]. A combination of Discrete Cosine Transform (DCT) and Artificial Neural Network (ANN) was used to develop a near lossless EEG compression approach. The authors in [10] first apply DCT and then perform ANN on the DCT coefficients. Two main benefits of applying ANN are reducing the dimensions of the coefficients of EEG data and estimating the original coefficients in the reconstruction phase. Arithmetic coding was used to quantize the difference between the original coefficients and the estimated ones to improve the reconstruction performance. Figure 1 presents the procedure of compression used.

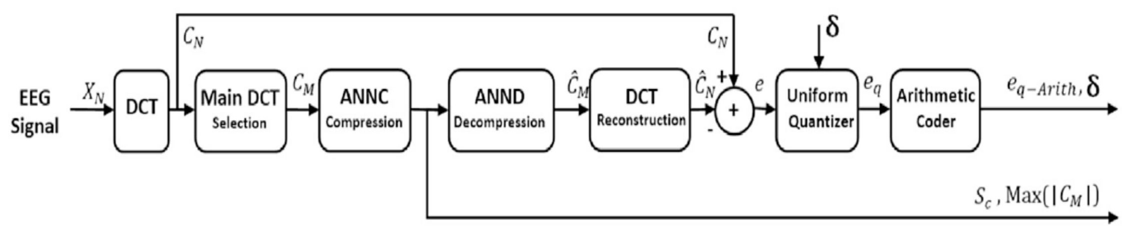

*Figure 1:*The Diagram of Encoder in the Proposed Compression Method in [10]

The experimental results show that the technique in [10] can give better compression ratios and less distortion compared to state-of-the-art. The authors did not use ANN directly on the EEG datasets; as training, will require more data and it would be hard to get the same reconstruction accuracy as what they achieve. Also, the two-stage compression approach tends to have high complexity to estimate the DCT coefficients, and apply ANNC for large data dimensions.

Another near lossless EEG compression approach was proposed by [11] using the neural network predictors, where the author tried four models of neural network and compare them with LMS adaptive linear and autoregressive (AR) model. The author found that single layer perceptron predictor produces the best results. In [12] the authors enhance the proposed near lossless EEG compression technique used in [11] by adding a preprocessing stage using the concept of correlation dimension (CD).

In [15], the neural network was used for EEG data compression, where a complete study about the effectiveness of this compression approach with respect to the overall energy consumption was presented. Their compression approach used Stacked AutoEncoder SAE which has mainly two functions encoding and decoding. The encoder transforms the original data to the compressed version with lower dimensionality using certain activation function and the decoder construct the data using another activation function. The bottleneck layer is the compressed data. Training is an expensive operation in SAE in terms of time, due to that, it was done offline at the server level to obtain the optimal set of weights and biases that will be used on the PDA. The authors' model is represented using three main layers, where the second layer is the bottleneck, and the compression ratio can be described by the number of neurons in this layer [15].

In [16], the authors consider the correlation among multiple modalities, by proposing deep learning approach using Stacked AutoEncoder (SAE) for compression and feature extraction as it considers both the intra-correlation and inter-correlation of the data from multiple modalities. The model implemented in [16] involves EEG and EMG pathways where each path can be considered as a separate SAE dedicated to learn the intra-modality correlation of the data and at a certain level, a layer that merges the features of both joint layers will be used. Training the model is done offline on the server side and the optimal configurations are obtained and leveraged by the devices attached to the patients (PDA), to apply the right compression using the pre-calculated weights and biases for different network conditions. DEAP dataset was used in their experiments and a comparison between SAE, DWT, CS and the 2D compression approach was done. The results show that having high compression ratios SAE would outperform the state-of-the-art methods.

2.1.1.2 Lossy Compression Techniques

The main idea behind lossy compression algorithms is to remove low energy (i.e. less important) coefficients, as a result of transforming the data into a specific domain, based on specific criteria, while keeping the most significant coefficients. Lossy compression has three main layers: transformation, quantization, and encoding. Discrete Fourier Transform (DFT), Discrete Cosine Transform (DCT) and Discrete Wavelet Transform (DWT) are examples of transformation algorithms that change the original signal to the frequency domain coefficient, which has the important coefficients that will be used in the later stages. Compressing a range of values into discrete values is done by the quantization layer, DCT and DWT are also used for quantization. Predictive coding, arithmetic encoder or Set Partitioning In Hierarchical Trees (SPIHT) can be used to support the last layer which is the encoding.

1.5-D Multi-Channel EEG compression algorithm was proposed by [17] to manage the transmission of EEG signals through the network with low complexity and high reliability. The algorithm combines DWT with 1D to reduce the computational complexity and No List Set Partitioning In Hierarchical Trees (NLSPIHT) algorithm to perform better compression rate. At the same compression ratio, the proposed algorithm achieves better results than 2-D NLSPIHT algorithm; while 2-D SPIHT algorithm has better performance in lower compression ratios. Preprocessing the raw data in [17] is done using 1D DWT, then the wavelet coefficients were reshaped into a matrix and used by 2-D NLSPIHT encoding algorithm to manage the compression.

Table 2-1 shows the relation between compression ratio and distortion using the three methods, where the 1.5-D Multi-Channel EEG compression algorithm has an effective performance, especially with high compression ratios.

A combination of DWT and Compressive Sensing (CS) in wireless sensors was proposed in [18] as a lossy compression technique to control both compression and energy consumption of encoder and transmitter, having certain energy threshold. However, the authors in [16] believe that CS is not efficient for EEG compression.

Table 2- 1

PRD (%) vs CR Comparison between 2-D SPIHT, 2-D NLSPIHT and 1.5-D NLSPIHT [17]

Algorithm	Tile Size	CR=2	4	6	8	10	12	14	16	18	20	22
2-D NLSPIHT	32×32	2.16	6.19	13.38	19.03	23.98	28.38	32.43	36.22	39.62	42.63	45.34
	64×64	1.94	5.52	11.94	17.05	21.57	25.66	29.29	32.75	35.85	38.58	41.04
	128×128	1.67	4.81	10.41	14.94	18.99	22.58	25.94	29.09	31.78	34.14	36.33
	256×256	1.50	4.31	9.37	13.65	17.29	20.65	24.01	26.73	29.13	31.38	33.39
	512×512	1.45	4.09	8.96	13.26	16.67	20.16	23.32	26.02	28.27	30.44	32.57
2-D SPIHT	32×32	0.34	5.47	12.72	18.64	23.60	28.10	32.29	36.22	39.86	43.16	46.16
	64×64	0.30	4.75	11.08	16.31	20.70	24.64	28.29	31.64	34.73	37.55	40.09
	128×128	0.26	4.12	9.60	14.18	18.06	21.59	24.82	27.79	30.51	32.99	35.23
	256×256	0.23	3.74	8.68	12.85	16.46	19.74	22.74	25.51	28.04	30.33	32.38
	512×512	0.23	3.62	8.39	12.41	15.96	19.16	22.09	24.81	27.29	29.51	31.51
This work	32×32	2.15	5.49	11.90	17.07	21.30	24.82	27.83	30.52	32.96	35.14	37.12
	64×64	1.94	4.78	10.40	14.99	18.69	21.84	24.49	26.84	28.97	30.87	32.61
	128×128	1.67	4.15	9.19	13.44	17.06	20.09	22.70	25.17	27.42	29.44	31.26
	256×256	1.50	3.76	8.32	12.20	15.46	18.13	20.55	22.80	24.83	26.55	28.14
	512×512	1.45	3.60	8.01	11.76	14.88	17.45	19.75	21.83	23.83	25.53	26.98

Real-time lossy EEG compression technique was proposed by Daou, Labeauin [19]. The method uses DWT and dynamic reference lists to get the decorrelated sub-band coefficients. SPIHT was used as source coder. The algorithm relies on the redundancy between different frequency sub-bands presented in different EEG segments of one channel in different time segments. The experiments expose the efficiency of the method in both compression and detecting seizure activity.

The authors stated that their approach is unlike the previous techniques which use the same time segment to extract the correlation and ignore the characteristic of EEG in which different EEG segments can give similar features that can be eliminated for the case of compression.

In [20], a lossy compression approach for an authentication system was developed by combining SPIHT compression algorithm with Discrete Wavelet to compress and reconstruct the signals. On the other hand, the approach used by [21] depends on pre-processing the parts of EEG signal, which need to be transmitted and stored by performing spectral band separation, then Discrete Wavelets Transform with the appropriate wavelets was performed to achieve better results compared with Discrete Fourier Transform.

2.1.1.3 Hybrid Compression Techniques

Discrete cosine transforms, and Huffman coding was used in [22] to outcome a hybrid lossless EEG compression technique; which could outperform 1-D SPIHT,2-D SPIHT-AC, 2-D SPIHT and JPEG2000. Another hybrid approach was proposed in [23] where reshaping of the signals was applied before performing the compression to utilize the channel correlation. Two representations of the data were proposed, image (matrix) or volumetric data (tensor). The authors use these two representations into two-stage compression, starting by wavelet-based lossy coding layer, followed by arithmetic coding on the residual as a lossless fashion, which allows having restrictions on the distortion of the residual; and facilitate the signal reconstruction.

In [24], the authors propose lossy and lossless techniques to achieve good compression and reconstruction of the EEG signal. The method starts by applying a lossy method either DCT or DWT based on the user decision, to get rid of the redundancy in the data; since all the coefficients below a certain selected threshold will be replaced by zeros.

At this stage, a lossless compression using either Run Length Encoding (RLE) or Arithmetic Encoding should be applied to the data to get high compression ratios without distortion.

Our work regarding compression will enhance the work done in [15], where convolutional autoencoder will be used instead of staked autoencoder, to respect the correlations presented in EEG signals, by arranging the EEG data into 2D formations that take into consideration the spatio-temporal correlation amongst the EEG samples.

Table 2-2 and Table 2-3 present all mentioned compression techniques, the datasets used for evaluating and performance metric if applicable.

Abbreviations used in Table 2-2 and Table 2-3:

- SNR: Signal to Noise Ratio. [8]
- MAE: Mean Absolute Error. [8]
- CR: Compression Ratio. [15]
- PRD: Percent-root mean square distortion. [15]
- MSE: Mean Square Error. [21]
- PSNR: Peak signal to noise ratio. [21]

Table 2- 2

Summary of Different Compression Techniques

Approach	Compression type	Dataset used	Performance metric	Comments
Stacked autoencoder neural networks (SAE) [15, 16]	Lossless	• BCI-IV-2a • BCI-IV-2b • DEAP dataset	• PRD • CR	Deep learning approach outperforms other techniques in high CR, but DWT is the best in low CR
DWT and dynamic reference lists [19]	Lossy	• Dataset 1- MIT dB • Dataset 2- MNI dB.	• PRD • CR	
Statistical modeling and encoding [8, 9, 25]	Lossless	• DB1a, b from BCI2000 system. • DB2a, b from BCI Competition III • DB3 from BCI Competition IV2 • DB4 from BCI Competition IV4 • Diagnostic ECG DB	• CR • MAE • SNR	
DWT in 1D and No List Set Partitioning in Hierarchical Trees (NLSPIHT) algorithm [17]	Lossy	CHB-MIT Scalp EEG Database	• PRD • CR • MAE	The proposed algorithm has an effective performance especially with high compression ratios
Wavelet-based lossy coding layer, and arithmetic coding [23]	Hybrid	• EEG-MMI • BCI3-MI • BCI4-MI	• CR • PRD • MAE	

Table 2- 3

Summary of Different Compression Techniques

Approach	Compression type	Dataset used	Performance metric	Comments
Spectral band separation with DWT [21]	Lossy	• Partial EEG • Epileptic EEG datasets	• *CR* • *PSNR* • *MSE*	Compared to Discrete Fourier Transform, this approach gives better results.
DWT and SPIHT [20]	Lossy	• Graz dataset A and B in the BCI Competition • Australian EEG Database recorded at the John Hunter Hospital	• *CR* • *PRD*	
Discrete cosine transforms, and Huffman coding [22]	Hybrid	• Five datasets (A-E) from Bone University database	• *CR* • *PRD*	Outperforms 1-D SPIHT,2-D SPIHT+AC, 2-D SPIHT and JPEG2000
DCT and ANN [10]	Near-lossless	• BCI2 Dataset-IV • BCI3 Dataset-II • BCI4 Dataset-I • Bonn university Dataset	• *CR* • *PRD*	

2.1.2 Wireless Network Resource Allocation

To facilitate patient's remote monitoring while the patient is mobile, wireless technologies are commonly used to deliver the patient's vital signs through dynamic wireless channel, which has lots of channel impairments such as fading, path loss, etc. Therefore, resource allocation is one of the most important techniques that guarantees maximum channel utilization, while providing Quality of Service (QoS) for multiple users. Due to this importance of network resource allocation; different approaches were addressed in the literature. In [26] these approaches were classified into four main classes, cost-function based, decision-making processes using game theory, Markov decision processes (MDPs), and optimization based.

Performing cost based resource allocation in the network to find the optimal solution such as [27]. The authors use distributed Lagrangian method to allocate a certain amount of resources for each node while minimizing the cost and satisfying the constraints of each node. However, using this approach in some cases would be hard as in [28], especially if it's not the only factor affecting the allocation. The authors in [29, 30] propose a technique using a game theoretic approach, where a competition for getting the bandwidth from different network occurs between the users. The computational complexity of this approach can be considered as a drawback that needs to be resolved. Also, getting the optimal solution is not a must in this approach where an inefficient utilization may occur [31].

Studying network switching between different RATs can be done using Markov and Semi-Markov decision processes [32, 33, 34]. Nevertheless, the size of the network has a great impact on the complexity where the larger the network, the harder it is to get a solution [31].

Optimization based approach would be hard to accomplish in some cases as finding an optimal resource allocation and user association with certain constraints might be an NP-hard problem; which could be managed by either relaxing the optimization or online adaptive learning for network selection as presented in [35].

In [36] the authors develop a user-centric approach which integrate both the network and application requirements such as transmission energy, application quality of service and monetary cost to find the suitable RAN(s) that allows all users to meet the system obligations. An enhancement was added to [36] by implementing adaptivity to the system using dynamic weights update mechanism to maximize the operating time of user equipment (UE) and finding the optimal decision even under dynamic changes to the network [26].

The authors in [37, 38] study the allocation problem to maximize the energy efficiency. In [37] a combination of four allocation schemes: antenna selection, time allocation, subcarrier and power allocation to improve the energy efficiency (EE) of the system was proposed. While in [38], an integration between Visible Light Communication (VLC) and Radio Frequency (RF)-based network in the wireless environment, was proposed to enhance the EE.

Characterizing the comparative advantage of several Virtual Network Function (VNF) was used to allocate the computing resources efficiently [39]. Some researchers focus on uplink scheduling such as [40, 41, 42, 43]. In [40] the authors propose an efficient use of the uplink radio resources to the machine-to-machine (M2M) and a user equipment

(UE) by applying an extended random access (RA) scheme in two stages. The authors in [41] used uplink scheduling to allocate the resources for a user equipment (UE) with high-priority data while performing an acceptable performance for other users.

A fast-distributed gradient method (FGM) was proposed by [44] to solve the network utility maximization (NUM) problem. The authors found that their approach overcomes all the sub-gradient methods, which suffer from a certain rate of convergence.

In [45], the authors proposed a multi-objective framework to manage the resource allocation while minimizing the transmission power, maximizing the energy efficiency and considering the Quality of Service (QoS) requirements and the limitations of the channel state information at the transmitter side.

2.1.3 In-Network Processing for Data Reduction and Resource Allocation

In-network processing focuses on processing vital signs at the edge network adaptively to respond to wireless network dynamics while addressing application-level requirements. Data reduction plays an important role in network resource allocation as optimizing the network and maximizing the use of resources is always a goal. Recently processing and transmitting medical data through the network became very active research area; due to the great enhancement of wireless and mobile communication technologies.

Many researchers went through different resource allocation schemes; since the challenges behind the design of Mobile-Health (m-Health) systems with Body Area Sensor Networks (BASNs) need to be considered; such as the limitation of small sensors, which get affected by the consumed energy, storage and computational resources, let alone addressing the application level requirements.

The design of the wireless sensor was considered by researchers through minimizing the size and complexity of the signal compression techniques. For example, the authors in [46, 47] studied the practical performance of different compressive sensing implementations when applied to scalp EEG signals. [48] discussed their proposed low-cost quadratic level compression algorithm to reduce the encoding delay, hardware cost of the minimized sensors along with the transmission energy. However, others concentrated on the in-network processing to manage the transmission of huge real-time data through the network. In [49], an encoding distortion model of DWT-based compression was proposed to analyze, controlling, and optimizing the behavior of the wireless EEG monitoring systems. The main decision parameters that affect distortion at the receiver side are compression ratio, wavelet filter length, and the channel models.

An efficient user-centric network association mechanism over Device-to-Device (D2D) communication integrated into heterogeneous wireless networks to enhance the system performance and support reliable connectivity was proposed in [50]. The solution composed of three main stages to identify the RAN(s) and/or inner node(s) that optimize the user's objective, while allowing all users to meet the system constraints. The authors in [51] propose ProbCache resource management algorithm for in-network caching, to allocate spaces based on the length of the path from the source to destination in order to reduce the caching redundancy and network traffic. However, their approach adds high computation complexity to the system.

Prolong the lifetime of the wireless BASNs can be done by minimizing the total consumption energy. In [52], the authors claim to optimize the Energy-Cost-Distortion using Lagrangian duality theory, where at the edge level (Patient Data Aggregator-PDA) the algorithm should run to find the optimal transmission rate and compression ratio, which satisfy the application layer constraints while providing the optimal trade-off among energy consumption, monetary cost and signal distortion. The proposed distributed cross-layer solution can be applied in heterogeneous wireless m-health systems. The authors in [53] developed an Energy–Rate-Distortion cross-layer framework which performs an optimal resource allocation for all users. The total energy consumption was minimized, having data delay deadline and distortion threshold as constraints. An Energy-Delay–Distortion cross-layer framework was proposed as an enhancement to the work in [53,54] to guarantee a good quality transmission of medical signals while having limited power and computation resources [55]. In [56] the authors introduced a cross-layer framework to minimize and adapt the total transmission time with respect to Bit error rate (BER), source coding distortion, encoding energy and transmission energy by applying a dynamic time-frequency slot allocation, instead of the conventional Time Division Multiple Access (TDMA) scheme which uses constant bandwidth allocation.

Unlike the work discussed above, we have proposed an Energy- Distortion cross-layer framework, where the data rate, compression, bandwidth and end-to-end delay are the decision parameters of the resource allocation problem, while considering practical mobility pattern of the patient to study the impact of mobility as a source of network dynamics on the in-network processing and resource allocation techniques. In Chapter 3, a detailed explanation is provided.

2.2 Background

2.2.1 Deep Learning

Deep learning is a subfield of machine learning, which aims to introduce artificial intelligence in machine learning. This intelligence can be achieved by learning, which can be supervised, partially supervised or unsupervised. Usually, deep neural network composes of multiple layers for feature extraction, where the output of a layer is used as an input for the successive layer and each layer consists of a certain number of neurons. Neurons work like the human brain to learn without the need for human input. The neuron in the first layer extracts the simple feature of the input and passes them to the next layer, which learns with time more detailed features about the input. The more the layers, the more feature extraction, which may in some cases affect the classification accuracy [57]. We call the layers between the input and the output as hidden layers; *Figure 2* shows a simple representation of the neural network.

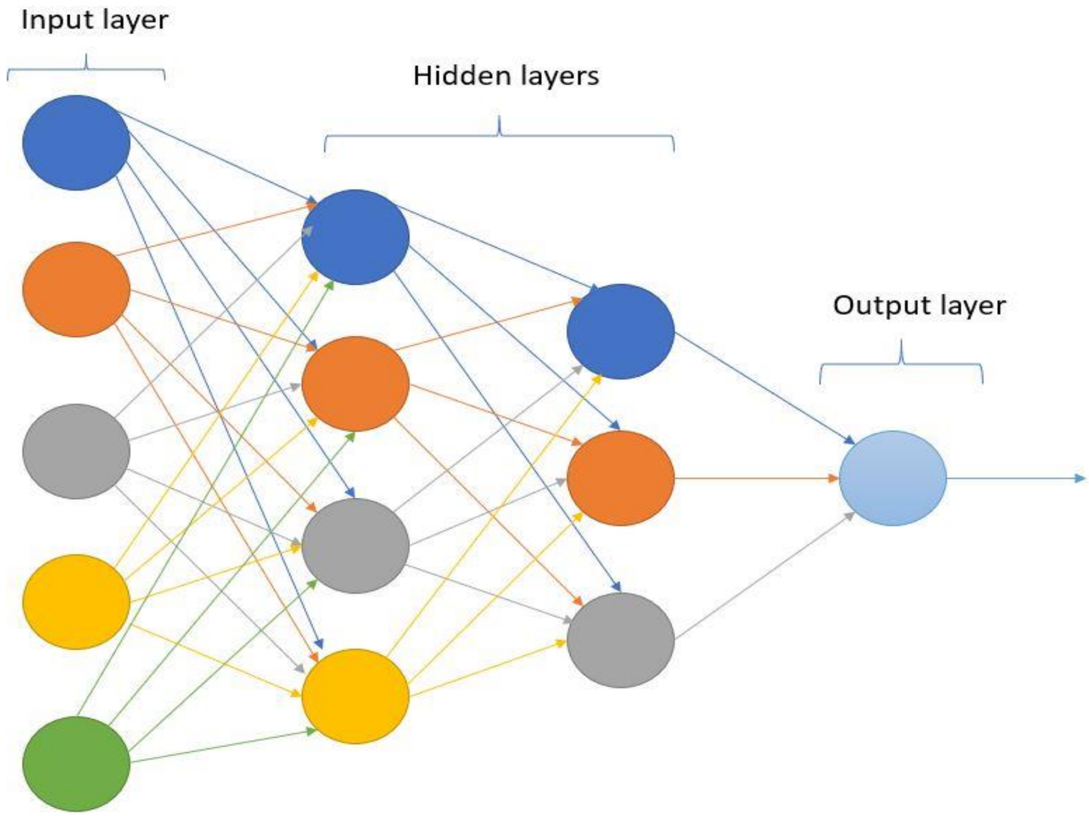

Figure 2: Simple Representation of a Neural Network, Where the Circles Corresponds to Neurons

Deep neural network is not a new field in the research. In 1986 back-propagation was introduced for the first time by Rumelhart. However, most neural networks were a single layer due to the cost of computation and availability of data. In July 2006 Hinton and Salakhutdinov, introduce the multi-layer neural network to reduce the dimensionality of the data in order to facilitate the classification, visualization, communication, and storage of high-dimensional data. In 2010, the first strong work has been published since 2006 by Abdel-rahman Mohamed, George Dahl, and Geoffrey Hinton, where they used deep neural networks in acoustic modeling. There are four major architectures of deep neural networks where all the layers could apply three main operations: forward propagation, backward propagation, and update which performs updates to the weights of the layer during training.

I. Recurrent Neural Networks (RNN)

Usually, RNN used with the data that have good interdependencies to maintain information about what happened in all the previous layers, where the output of a layer depends on the previous computations. This type of neural network does not go through the update operation of the weights; since it uses the same weighs across all the layers. Due to that, the total number of parameter that the network needs to learn will be reduced. It’s worth to mention that RNN has great achievements in language modeling, bioinformatics and speech recognition applications [58].

II. Autoencoder (AE): will be explained in the coming subsection.

III. Convolutional Neural Networks (CNN): will be explained in the coming subsection.

IV. Recursive Neural Networks:

Like Recurrent Neural networks as it has a shared weight matrix, however, it uses a variation of backpropagation called backpropagation through the structure (BPTS) since it uses binary tree structure in learning [59].

2.2.1.1 Autoencoder

Autoencoder (AE) is a neural network usually used for unsupervised learning as it aims to recreate the input rather than classify it under certain classes [60]. The number of neurons in the input layer equals to the number of neurons in the output layer. Unlike other neural networks, the hidden layers have a smaller number of neurons compared to input/output layers. This is because autoencoder proposed to encode the data with lower dimensionality and extract the discriminative features. The number of hidden layers depends on the dimensionality of the input as single hidden layer will not be sufficient to represent high dimensionality data, and in this case, we call the model as deep Autoencoder model [58]. General representation of autoencoder presented in *Figure 3*.

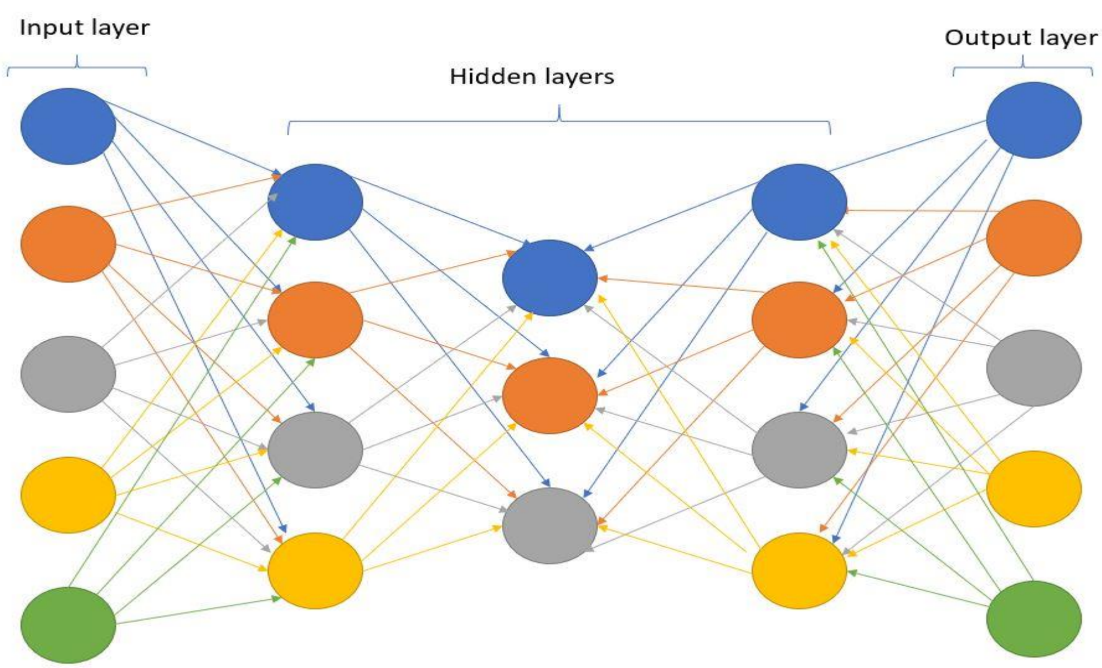

Figure 3: General Representation of Autoencoder

2.2.1.2 Convolutional Neural Networks

It was proposed in 2012 by Krizhevsky, Sutskever, and Hinton in [61], where they called it AlexNet. Many other configurations were developed after that such as Clarifai, VGG, and GoogleNet.

Convolutional Neural Network (CNN) is an efficient artificial neural network approach, which was proposed to manage the data that has local correlations while minimizing the number of training parameters. It was called convolutional because it performs a complex operation using convolutional filters on the entire image instead of using neurons in its layers. Using filters decrease the number of connections between the layers. CNN was able to outperform the state-of-the-art techniques in advanced computer vision and natural language processing tasks [62].

There are three main types of layers used in a convolutional neural network, which are the following:

I. Fully connected layer:

They are usually used as the last layer after the convolutional and pooling layers to convert the 2D input into 1D output. Nonlinear activation function could be used in these layers in order to get the class prediction of the output [62].

II. Convolutional layer:

Consist of filters with certain dimensions to extract the features from the input. Filters are another representation of neurons, which generate an output value of a weighted input. The filter should move through the input and capture the features. If the size of the input is not divisible by the size of the filter; then padding technique should be performed on the input. Convolutional layers can expect as an input either normal pixels values if it is an input layer or feature map if it is a hidden layer. Feature map is the output of a filter in the previous layer, where we might get multiple feature maps from the previous layer based on the number of filters used [62].

III. Pooling layer:

It is called down-sampling layer, which usually located after the convolutional layer to reduce the spatial dimensions of the input for the next convolutional layers. Pooling layers usually used to minimize the following [62]:

- Computational overhead.
- Computational costs; since the number of parameters will be reduced.
- Chance of getting over-fitting

2.2.1.3 Convolutional Autoencoder

A combination of Convolutional Neural Networks (CNN) and Autoencoder produce Convolutional AutoEncoder model (CAE). The fully connected layers in SAE will be replaced by convolutional layers in both encoder and decoder sides. This type of models usually used in image processing; because of having convolutional layers. .

We will use this model in our proposed compression solution as the encoder will apply compression of the EEG signal on the sender side (PDA) and the decoder will reconstruct the signal at the receiver side (m-Health center). The main advantage of using Convolutional Autoencoder model is applying high compression ratios since the number of filters in the bottleneck layer corresponds to the compression ratio and better reconstruction of the EEG signal as CNN respects the correlation occurred between and within the input, and EEG signals have spatio-temporal correlation amongst it samples. In this thesis, the power of convolutional layers will be used to consider this aspect.

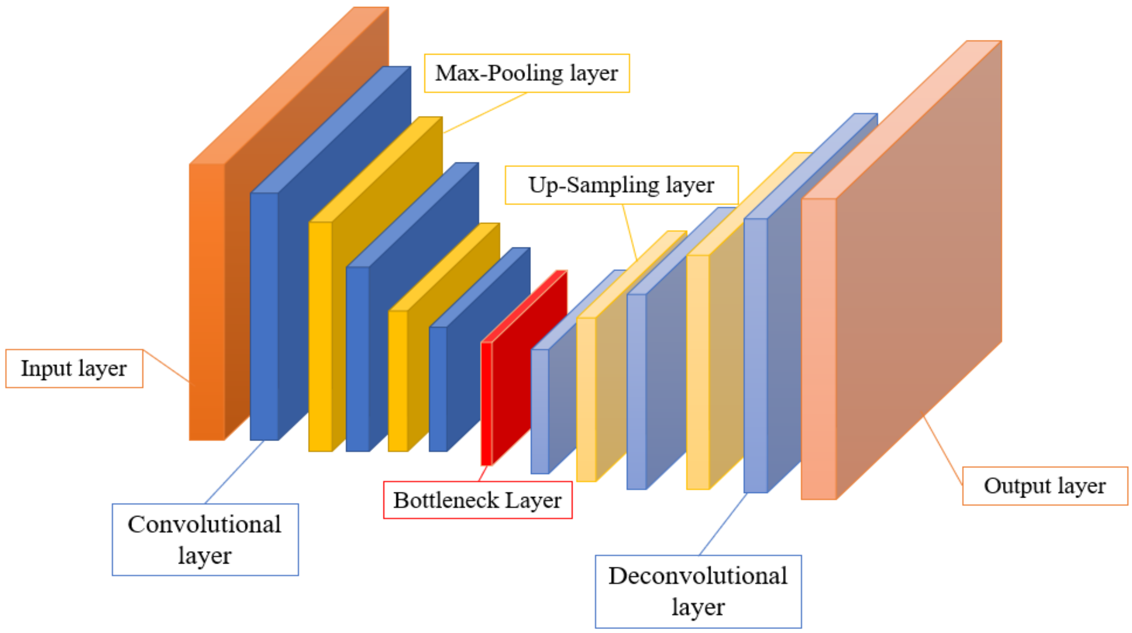

Figure 4: Representation of Convolutional Autoencoder

2.2.2 Electroencephalogram Signals (EEG)

The nervous system of the human controls the functioning of the brain by sending continues electrical pulses to different parts of the brain. The electroencephalogram (EEG) is a monitoring technique that records the electrical activity (electrical pulses) of the brain from the scalp through a group of sensors called electrodes, which collect and transmit the EEG signals to another device/station to be analyzed [24]. EEG signals are considered to be one of the richest sources of information about the human body. Hence, it can be used to diagnose different brain disorders such as epilepsy, loss of consciousness or dementia, check the status of the brain if dead for a person in a coma, sleep disorder and much more.

However, analyzing EEG is also very challenging since they contain several artifacts from different parts of the body. EEG signals usually have both intra-correlation and inter-correlation, which may opportunistically lead to efficient application-specific representations. Many representations were proposed to manage these correlations as mentioned above [23].

There are five main frequency bands in EEG signals; which are delta, theta, alpha, beta and gamma [21].

- Delta: it has a frequency range of 4 Hz or below. It tends to have high amplitude and the slow waves, where it's usual representation of dominant rhythm in infants up to one year and may occur in certain sleep stages.
- Theta: it has a frequency of 4 to 7.5 Hz and it tends to have low-medium amplitude and too slow activity. It is normal to have theta waves in children up to 13 years, while sleeping and an adult in unconsciousness state.
- Alpha: it has a frequency between 7.5 and 13 Hz. It appears in normal relaxed adults as it disappears while opening the eyes or doing any operation. Alpha waves tend to have considerably low amplitude.
- Beta: has a frequency of between 13 and 30 Hz and it tends to have low amplitude. It can be used for a patient in alert or anxious case. Gamma: it has a frequency of 30 Hz and above. Gamma appears normally while processing information such as working memory and attention.

2.2.3 Discrete Wavelet Transforms (DWT)

DWT is one of the most known algorithms that can be used for compression, quantization, and encoding on the raw EEG data, where the wavelets are discretely sampled, and it can capture both frequency and location information; which make it efficient in analyzing such non-stationary signals like EEG [63, 20].

Discrete wavelet transforms decompose the input signal into two frequency bands using high and low pass filters as part of the processing and compressing the signal. The high pass filter uses wavelet functions, while the lowpass filter uses scaling functions.

These filters will output two types of coefficients. The filtering process in DWT can be done in different levels, which give different resolution for the same input signal. The approximation coefficients correspond to low frequency and detail coefficients corresponded to high frequency. Inverse Discrete Wavelet Transform (IDWT) is the technique used to reconstruct the compressed signal with a certain amount of distortion.

2.2.4 Spatio-TEmporal Parametric Stepping Mobility Model (STEPS)

Spatio-TEmporal Parametric Stepping (STEPS) is a mobility model that can cover a large spectrum of human mobility patterns and from the name, it uses the power law to rule the movement of the node and make abstraction of Spatio-temporal preferences in human mobility. Implementing STEPS model is easy, flexible to configure and theoretically tractable [64]. Other mobility models ignore the purpose of the user movement during certain time; as most models focus on spatial features of human mobility, while ignoring the temporal features (when they visit the location) and most likely the users in STEPS spend more time in their favorite locations [65]. There are many mobility models that represent the movement of human/objects. However, we decided to use STEPS mobility model as it's the closest representations of movement in certain zones, which can be described as the movement of the patients within the hospital or at home for example. We have used the implementation of STEPS mobility model where we can specify the dimensions of the zones "it should be $N \times N$", the minimum and maximum speed of the users/patients, total number of users/patients and finally the simulation time. The output of running the model is an array of $(x\ ,\ y)$ location for all users at each time slot of the simulation. *Figure 5* represents a snapshot of the animation when running the model in MATLAB.

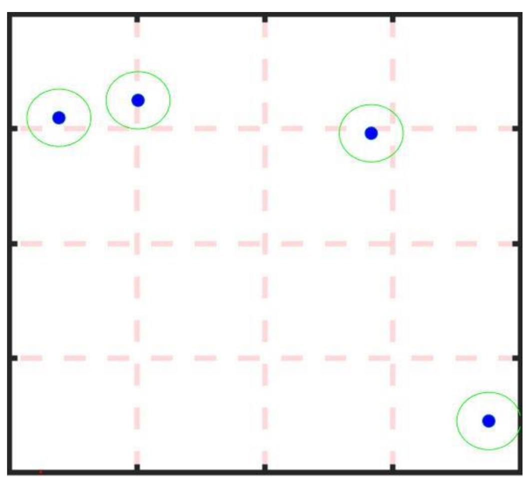

Figure 5: Snapshot of Running STEPS Mobility Model, having 4 Users, 16 Different Zones, Minimum Speed 2 and Maximum Speed 6. " Only for Demonstration"

CHAPTER 3: METHODOLOGY

In this Chapter, our proposed approach will be presented for implementing an adaptive EEG compression technique on the client side at the edge network and reconstruct the signals on the server side at the health center with the minimum distortion possible while considering the changes in the wireless network. In the following sections, we will explain (1) Preprocessing steps, (2) The adaptive compression technique using Deep Learning and (3) The optimization of resources in a wireless network. *Figure 6* summarizes the workflow of our proposed solution.

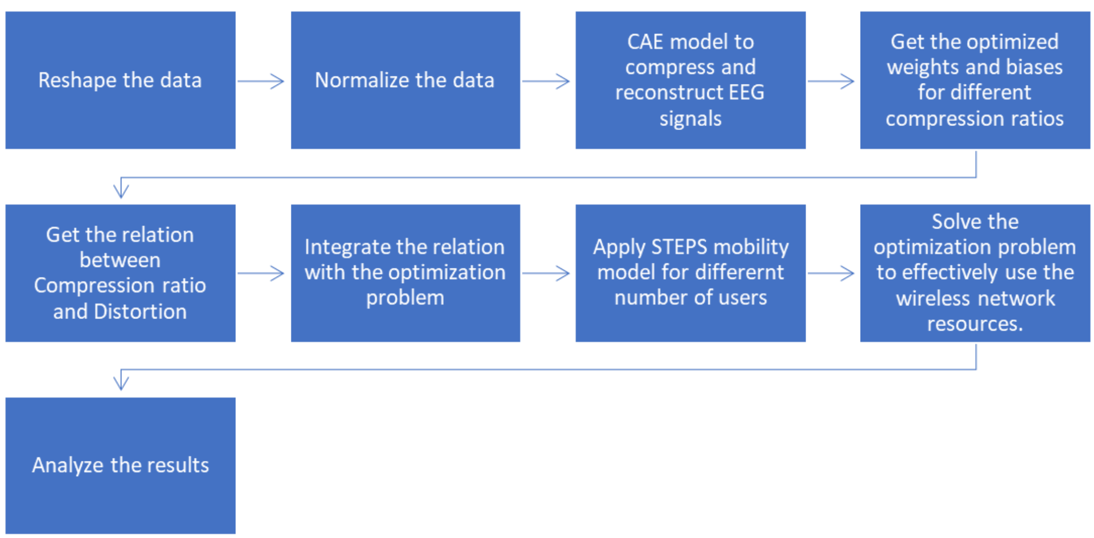

Figure 6: The Workflow of Our Proposed Solution

A representation of the system model in *Figure 7*. The scenario supposes that a different number of users changing their locations, where each patient has a wearable device or some sensors attached in/on his/her body to collect the EEG signals and send them to the Patient Data Aggregator (PDA).

The PDA should find the best compression ratio to compress the data based on the current network state and the application requirements by running the resource allocation algorithm. After that, it applies the proposed compression technique on the data before transmitting through the wireless network to the m-Health center.

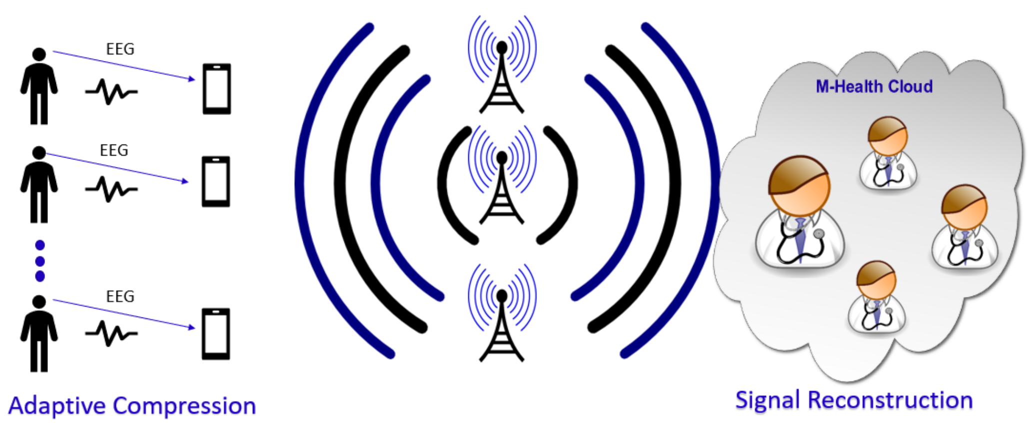

Figure 7: Scenario Where the Proposed Solution Can be Used

3.1 Preprocessing

In this subsection, a detailed description will be discussed for the pre-processing steps that aim to improve the quality of compressing the EEG signals, and minimizing the distortion of the reconstructed signal while applying the maximum possible compression ratio. For this purpose, reshaping and normalizing the data was used. The preprocessing step is done in the edge level of the network (at the PDA).

3.1.1 Reshaping the Data

Our datasets of Electroencephalogram (EEG) signals were represented in a 2D matrix; where each row represents an EEG sample. After that, ZigZag approach described in [66] was applied as only the even rows are flipped. The authors believe that this approach will exploit both spatial and temporal correlations of the EEG and hence enhance the performance of compressing EEG signals as they achieve the maximum correlations between the EEG signals. This approach was used in image compression algorithms as well. We have applied our compression/reconstruction technique without considering the ZigZag approach and it was found that the visualization and distortion rate occurred get worse compared with applying the ZigZag approach. Therefore, in the following, we fix the reshaping to use the ZigZag approach. *Figure* 8 represents an illusion of ZigZag approach applied to metrics.

X11	X12	X13	X14	X15	X16	X17	X18	X19
X29	X28	X27	X26	X25	X24	X23	X22	X21
X31	X32	X33	X34	X35	X36	X37	X38	X39
X49	X48	X47	X46	X45	X44	X43	X42	X41
X51	X52	X53	X54	X55	X56	X57	X58	X59
X69	X68	X67	X66	X65	X64	X63	X62	X61

Figure 8: A Simple Example Represents the Zigzag Approach Used in [66]

3.1.2 Normalizing the Data

In machine learning, normalizing the data is an efficient step during the training and testing. According to [62], normalizing the data to make the values between zero and one as a pre-processing step before usage will guarantee a stable convergence of weights and biases. Moreover, in deep neural networks, training is a very important step, and keeping the data without normalizing will make the training step more complicated and slow. This phenomenon called internal covariate shift and solving it can be achieved by normalizing the training data before being used in the input layer [67]. We have used equation (3.1) to normalize the data.

$$\text{Value after Normalization}= \frac{Value\ Before\ Normalization - minimum\ value}{maximum\ value - minimum\ value} \quad (3.1)$$

3.2 Data Compression and Reconstruction Using the Proposed Solution

As mentioned earlier we decided to explore the power of deep learning in compressing and reconstructing EEG signals. Two main models in deep learning were combined in order to create our proposed model. Autoencoder and convolutional neural networks were combined to end up with convolutional autoencoder. Convolutional Autoencoder (CAE) was used before in many image and motion processing applications [68, 69, 70]. However, our main aim here is to compress the EEG signal to the maximum while being able to reconstruct the signal with the minimum distortion. *Figure 9* denotes our model, which consists of the input layer, multiple hidden layers, and an output layer.

The output of each layer (feature map) is an input to the next layer, until reaching to the output layer. Since we are using Convolutional layers the EEG signal will be reshaped in 2D and the number of filters in each layer changes regularly until reaching to the bottleneck layer which gives certain compression ratio. We believe that the usage of convolutional layers instead of fully connected layers will utilize efficiently the significant correlations in EEG signals. CNN can also learn the internal features representation to expect the spatial relationship between the entries of the input, hence provides efficient representation of the EEG as a challenging non-stationary signal.

The general network architecture used for a certain dataset can be summarized as follows:

- Input layer: it represents the input (EEG signal) in 2D.
- Convolutional 2D layer: the input will be processed by a certain number of filters with certain kernel size.
- Max pooling layer: reduce the dimensions of the input "feature map" by a certain value.

- Convolutional 2D layer: its input is the reduced feature map from the previous layer and it will be processed by a certain number of filters which is less than the number of filters in the first convolutional layer.
- Max pooling layer: reduce the dimensions of the input "feature map" by a certain value.
- Convolutional 2D layer: its input is the reduced feature map from the previous layer and it will be processed by a certain number of filters which is less than the number of filters in the second convolutional layer.
- Max pooling layer: reduce the dimensions of the input "feature map" by a certain value and this layer considered as the bottleneck layer, where the number of filters represents the compression ratio that should be applied to the data.

The same number of layers will be repeated in the decoder side but instead of using max-pooling layers, up-sampling layers will be used. The first part of the model which is the encoder compresses the data to some limit while the second part is the decoder that works on reconstructing the data to get the original signal with the minimum distortion.

The training of the network is done offline and once a good performance is achieved, the model configuration (matrix) will be saved and used on the PDA of the client.

After running the optimization resource allocation problem to find the suitable compression (if needed), the configuration matrix corresponding to the compression ratio will be multiplied with the data to transmit the compressed data through the network.

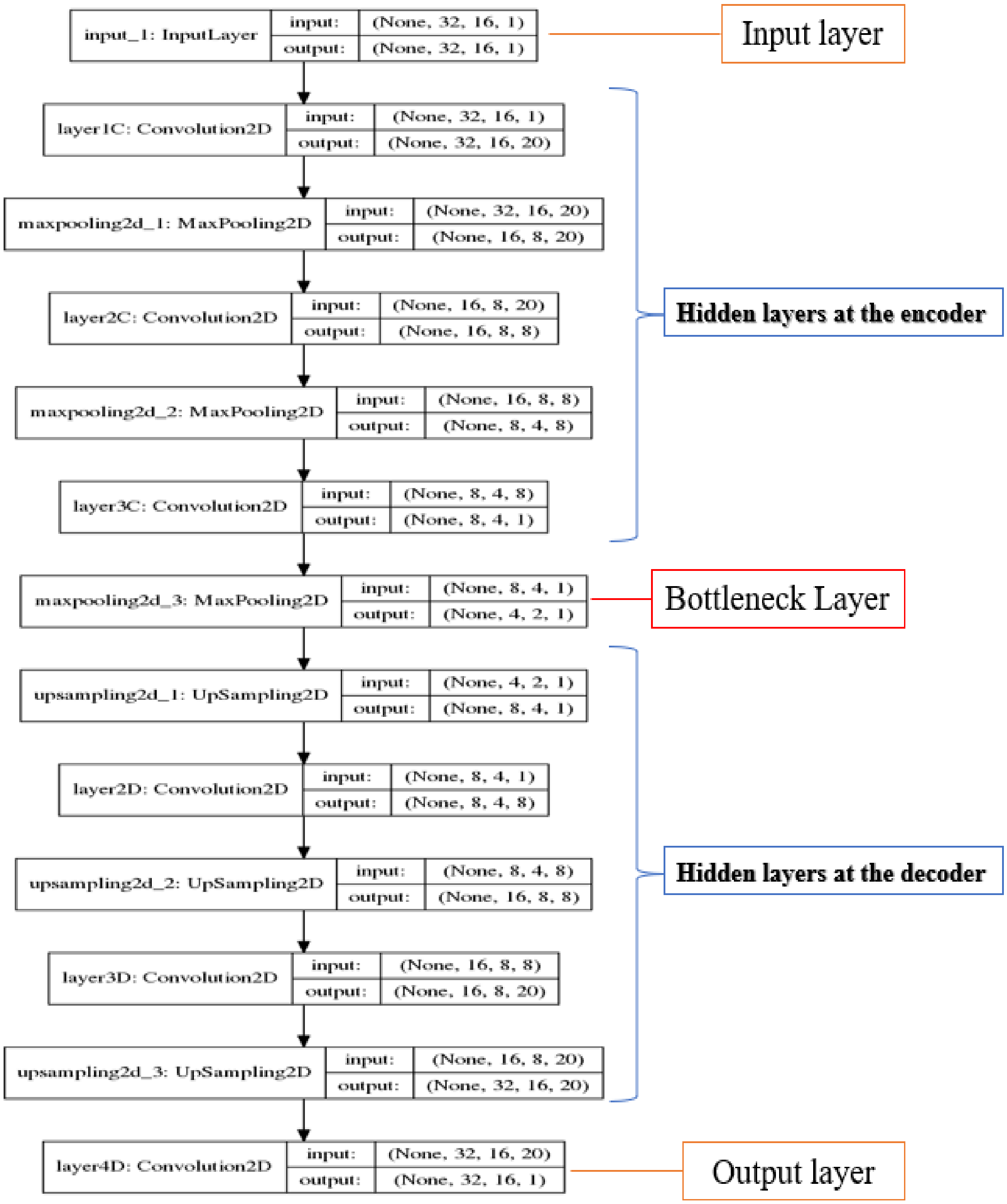

Figure 9: Deeper Representation of the Proposed Compression Approach on Certain Dataset

Here is an abstract explanation about each used layer in the proposed model, where fully connected layers are not used in the proposed model as the aim of the model to compress and reconstruct the data not to classify them into classes.

- **Convolutional 2D layer:** it was used for the output and hidden layers.
- **Max-Pooling layer:** max-pooling layers were used in the model, where the max filter is used to get the maximum value of a certain region of the input, ending up with a matrix of the maximum values of all regions. *Figure 10* represents a simple example of how max-pooling layers work.

Max pooling layer affects only the width and the height of the input, but the depth stays the same. For example, if the dimension of the input is (x, y) after applying max pooling with non-overlapping kernel size (k, k), then the dimensions of the output will be $(\frac{x}{k}, \frac{y}{k})$.

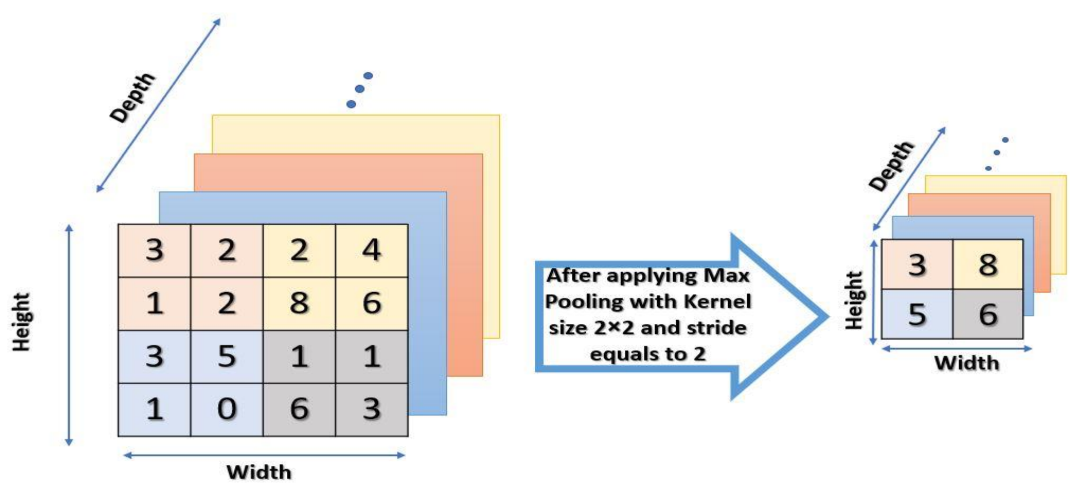

Figure 10: A Simple Example of How Max Pooling Layer Works

- **Upsampling layer:** it is the opposite operation of pooling, where it tries to reconstruct and approximate the original input, by repeating the rows and columns of the data.

An explanation of how to calculate the number of filters that represent certain compression ratio is illustrated by the example below with the following assumptions:

- Input size as a vector is *512*. Since we are using CNN, then we should reshape it in 2D where the size of the input will have the dimensions of *32×16*.
- If the bottleneck layer has an expected input size of *4×2* then having 1 filter at this layer will give around *98%* compression.

The compression ratio (CR) is defined as the data reduction in size relative to the uncompressed size of the data, represented by equation (3.2) below, where F is the number of filters, $M_i \times N_i$ is the dimensions of the input for the bottleneck layer and O_{Sample} is the size of the original data.

$$\text{Compression ratio (CR)} = \left|\frac{(\text{F} \times M_i \times N_i) - O_{Sample}}{O_{Sample}}\right| \times 100\% \tag{3.2}$$

$$\text{Compression ratio} = \left|\frac{(1\times4\times2)-512}{512}\right| \times 100\% = 98.4\ \%$$

So, if we are looking to compress 90% of the data then the number of filters needed should be calculated as the following:

$$Number\ of\ filters\ (F) = \frac{O_{Sample} - (CR \times O_{Sample})}{M_i \times N_i} \tag{3.3}$$

$$Number\ of\ filters = \frac{512 - (0.9\ \times 512)}{4 \times 2} = 6.4 = \lfloor 6.4 \rfloor = 6\ filters$$

Note that the value of the filter should be an integer, if not then floor function will be applied if the decimal value is less than *5*, else we apply ceil function. In this example, we applied floor function which gives *6* as the number of filters. However, we are not compressing exactly *90%*. To calculate the new compression ratio, we should apply the compression ratio equation to the following:

$$\text{Compression ratio (CR)} = \left|\frac{(6\times4\times2)-512}{512}\right| \times 100\% = 90.625\ \%$$

From this example, we can conclude that the size of the dataset and then the size of the input in the bottleneck layer decides the maximum compression ratio applied to the signal.

It is worth mentioning that the runtime complexity of using CAE approach can be represented by equation 3.4

$$O(\textstyle\sum_{i=1}^{d} L_i \times \frac{M_i \times N_i}{K^2}) \tag{3.4}$$

Where d is the number of convolutional layers in the neural network, L_i is the number of filters in layer i, K is the kernel size and $M_i \times N_i$ is the dimensions of the input for the i^{th} layer.

In our proposed CAE models, the size of the input is always divisible by the kernel size. The size of the input changes regularly, as a result of having max-pooling layers.

Summary:

Using Convolutional Autoencoder as a technique to compress the medical data is a new approach, which was not addressed before. Three main types of layers were used to build our model, convolutional, max-pooling and up-sampling layers. Unlike any other compression technique, deep learning is dataset and application specific. As we can't use the same model for different datasets, this outcome will be shown and discussed more in Chapter 4.

3.3 The Optimization of Resources in a Wireless Network

In practice, the state of a wireless network changes frequently, which denotes the importance of making the compression adaptive to the network state. In this subsection, the communication between patients' mobile devices and m-Health cloud, using wireless infrastructure will be managed as well as an explanation of the optimization problem of allocating the network resources having a huge amount of data transmission and limited resources. As mentioned before, our main focus is minimizing the transmission energy of the compressed data and the distortion of the reconstructed data of the patient while having different bandwidth and transmission rates.

The mobility of users was considered, where compressing and transmitting the data while the user is moving should be applicable; as it will affect some parameters of the optimization problem.

We used Spatio-TEmporal Parametric Stepping (STEPS) mobility model which was explained in Chapter 2, Section (2.2). The output of running the STEPS mobility model is an array of the x and y locations for all users at each time slot of the simulation. These locations were used to calculate the distance between the users and the base station (wireless access point), which assumed to be at $x_0 = 0$ and $y_0 = 0$.

By the end of this step, a two-dimensional array represents the distance between different users and the base station at different time slots is created and used in the optimization problem while solving for the optimum solution. It's important to note that changing the distance will affect the channel and the amount of compression applied to the data, causing changes in the distortion and transmission energy as stated in Chapter 4.

$$Distance = \sqrt{(x_i - x_0)^2 + (y_i - y_0)^2} \tag{3.5}$$

The optimization problem is solved with respect to five different decision parameters, which are:

- CR: The amount of compression ratio applied to the data of each user before transmitting.
- R: The data rate of each user should not exceed a certain threshold.
- D: The distortion of the reconstructed signal generated by each user should be less than the distortion threshold (D_{th}).
- The total end to end delay should be less than delay deadline.
- w: The bandwidth consumed by all users, which should be less than the maximum bandwidth of the network.
- The location of the user with respect to the base station.

To present a wireless communication environment, we use Rayleigh channel model [52], where the signal to the receiver is affected by both path loss and multi-path fading.

As stated earlier, we propose to optimize the transmission energy and the distortion in a wireless environment. Equation (3.6) represents the transmission energy for a user at certain time slot while having certain channel characteristic; which will be affected by the location (distance between the user and the base station) of the user at that time slot.

$$E_T = \frac{x_i \times l \times w_i \times k_i \times 2^{\frac{r_i}{w_i}}}{r_i} \tag{3.6}$$

Where E_T is the transmission energy for user i, x_i is the channel gain for user i, l is the packet length, k_i is the transmitted data, r_i and w_i are the data rate and bandwidth of user i respectively.

Equation (3.7) represents the relation between transmitted data and distortion for certain user i using our proposed model. The values of the parameters c_1 and c_2 change based on the application (dataset).

$$Distortion\ (D) = c_1 {k_i}^{c_2} \tag{3.7}$$

DWT is one of the most known algorithms, which was mentioned in Chapter 2, Section (2.2). We have used the DWT equation proposed by [52] for compression. However, we have used the regression model as depicted in [52], but we are looking to solve the optimization using CVX optimization tool instead of using transformations and solve using Lagrangian. Equation (3.8) represents the modified equation for distortion, where D is the distortion of the reconstructed signal, F is the wavelet filter length, k_i is the transmitted data for the i^{th}user and c_{11}, c_{22}, c_{33}, c_{44}, c_{55} and c_{66} are the model parameters which are estimated by the statistics of the typical EEG encoder.

$$D = c_{11}. F^{-c22} + c_{33}. e^{c_{44}k_i} + c_{55}. (k_i \times 100)^{-c_{66}} \tag{3.8}$$

Equation (3.9) is the general representation of the objective function where the optimization was applied to find a trade-off between the transmission energy and the distortion of the reconstructed signal for all users at the certain time considering the mentioned constraints in the beginning of Section (3.3). Both transmission energy and distortion were normalized by dividing the total value by the maximum value of both as presented in (3.9). The maximum value for the distortion rate is *100*. The w is the weighting factor, which has a value between 0 and 1, and can be set based on the desired transmission energy and distortion trade-off. When the value of the weight equals to 1, this represents minimizing the transmission energy only and neglecting the distortion. On the contrary, when the weight equals to 0, then the transmission energy will be neglected, and only the distortion will be considered. n is the number of users, E_{max} represents the maximum energy, E_T which was defined in (3.6) and D_i is the distortion represented in (3.7 and 3.8) for user i. w is the weighting factor, r_i is the data rate for user i, k_i is the transmitted data for user i, w_i is the bandwidth allocated for the i^{th} user, d_i is the end-to-end delay for user i; $d_{deadline}$ is the delay deadline, W_{max} is the maximum bandwidth and D_{th} is the distortion threshold which will change within a certain range (e.g. Between 8% and 12%).

$$P: min\left(\sum_{i=1}^{n}\left(\left(\frac{w}{E_{max}} \times E_T\right) + (\frac{1-w}{100} \times D_i)\right) \right)(3.9)$$

Subject to:

$$r_i > 0 \tag{3.10}$$

$$1 \geq k_i > 0 \tag{3.11}$$

$$w_i > 0 \tag{3.12}$$

$$\sum_{i=1}^{n} w_i < W_{max} \tag{3.13}$$

$$\sum_{i=1}^{n} d_i \leq d_{deadline} \tag{3.14}$$

$$D_i < D_{th} \tag{3.15}$$

Equation (3.16) represents the maximum data rate; where *l* is the packet length and $d_{deadline}$ is the delay deadline.

$$R_{max} = \frac{l}{d_{deadline}} \tag{3.16}$$

The maximum energy represented in equation (3.17) below:

$$E_{max} = \frac{x \times W_{max} \times l \times 2^{\frac{R_{max}}{W_{max}}}}{W_{max}} \tag{3.17}$$

x is the channel gain of the user, $W_{max}$is the maximum bandwidth available in the network, *l* is the packet length, $R_{max}$is the maximum data rate of a user.

In Chapter 4, we will go through the setup of the models and the analysis of the results.

CHAPTER 4: RESULTS AND DISCUSSION

In this chapter, we will present the results of both our proposed compression technique and solving the optimal resource allocation in a wireless environment.

4.1 Datasets

We conduct our experimental analysis on three datasets

- BCI-IV-2a from the BCI Competition IV [72]: These EEG signals recorded from 9 subjects, 22 electrodes, four different motor imagery tasks: the imagination of movement of the left and right hands, both feet and tongue. The EEG signals were sampled at 250 Hz and bandpass-filtered between 0.5 Hz and 100 Hz. The dataset contains a total of 7548 data samples. We divide it into 6416 for training and 1132 for testing.
- BCI-IV-2b from the BCI Competition IV [73]: These EEG signals recorded from three bipolar recordings: C3, Cz, and C4 with a sampling frequency of 250Hz from 9 participants. Like BCI-IV-2a, the signals were bandpass-filtered between 0.5 Hz and 100 Hz. The dataset contains a total of 5892 data samples. We divided it into 5202 for training and 690 for testing.
- DEAP dataset [74]: EEG recording of 32 participants where we have a total of 11520 data sample for both testing and training.

4.2 Setup environment

4.2.1 Compression

We have used Lenovo IdeaPad Z570 (Intel Core i5) with Ubuntu 14.4 operating system. Python programming language was used to build our compression CAE model as it has professional easy to use built-in libraries such as SciPy for scientific computation and scikit-learn which is a professional grade machine library.

TensorFlow and Theano are two main numerical libraries for building a deep neural network, in Python. We have used TensorFlow which was released by Google, and it can be used directly to build the model or using wrapper libraries built on top of TensorFlow to simplify the process.

Python has a simple library called Keras, which run on top of TensorFlow to hide the complexity behind the TensorFlow neural network models and facilitate the process of building deep learning for research and development. Keras can run on both GPUs and CPU which make it more powerful and effective; since using GPU will reduce the time required for training a model [62].

As mentioned in Chapter 3, CAE was built using three types of layers, convolutional 2D layer, pooling layer, and up-sampling layer. Each layer was used for a certain purpose and here is the detailed information about the parameters used in each layer on both BCI2-IV-2a and BCI-IV-2b dataset; since deep learning approach is dataset specific as we will explain later in the Chapter.

- Input layer: it will receive the input and reshape it in 2D, for BCI-IV-2a dataset the dimensions of the input is 32×16, and the dimension used for BCI-IV-2b 32×24.
- Convolutional 2D layer: 4×4 kernel size was used for the filters and Relu activation function in all convolutional hidden layers. The number of filters changes based on the amount of compression that needs to be applied to the data. The number of hidden Convolutional layers in BCI-IV-2a equals to three in both, the encoder and decoder and the one bottleneck layer. In BCI-IV-2b, two convolutional layers were used for most of the cases and one bottleneck layer.
- Pooling layer: we used a max-pooling layer to apply the reduction of the input size by *2*.
- Up-sampling layer: we used *2* as the up-sampling factor for rows and columns.

The model was compiled and configured for training using **Adam** optimizer**, mean square error** loss function and **mean absolute error**, **percentage root distortion rate** and **accuracy** as a compilation metrics. The model was trained for *70* epochs with a batch size of *5*.

At the end of the training and after the testing, the input signal and the predict signals which represent the reconstructed signal should be saved in mat representation to give us the chance of visual evaluation

The figure below shows encoder and decoder models when applying 61% compression on BCI-IV-2a dataset

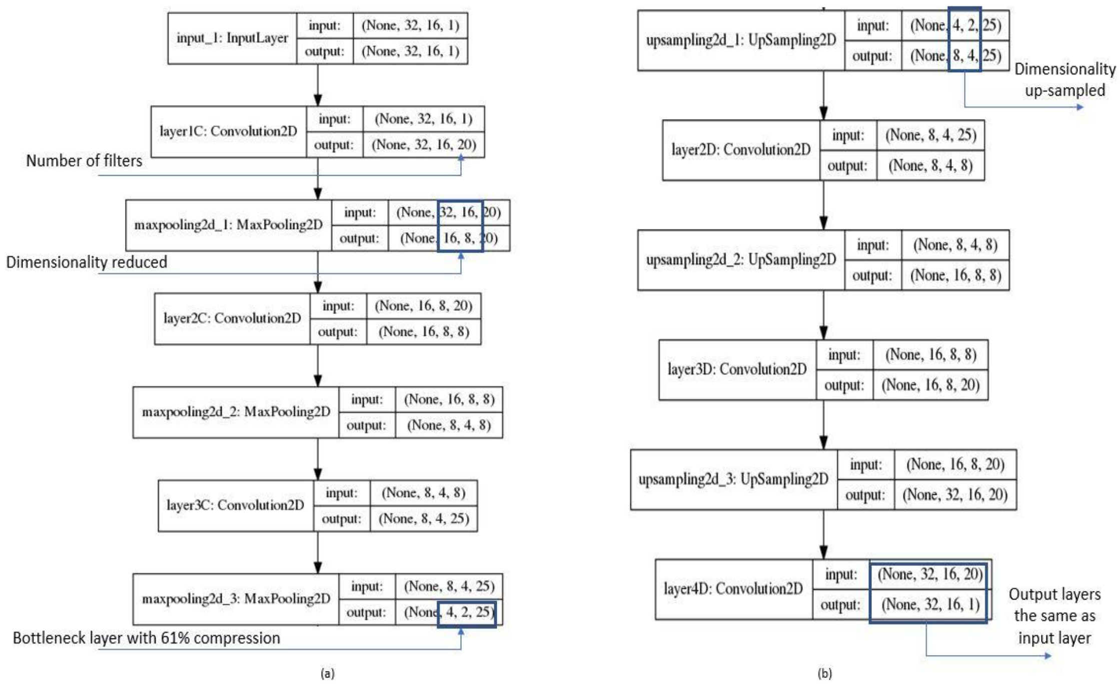

Figure 11: (a) Represents the Encoder Part of the Model Which Should be Implemented in the Edge Level of the Network (like PDA). (b) Represents the Decoder Part of the Model Which Should be Implemented on the Server Side of the Network (m-Health Server).

4.2.2 Optimization of the Network Resources

We have used Lenovo T460s (intel Corei7) with Microsoft Windows 10 Pro operating system to run the optimization problem. MATLAB was used to build the system model where the optimization problem should run. The CVX modeling system for convex optimization was used to build and solve the optimization problem by defining a set of constraints and objectives of the optimization problem [75].

In Chapter 3 we discussed the system model, where the patients change their locations while sending their EEG records through the network with respect to the network and application requirements. Table 4-1 represents the simulation parameter used when solving the optimization problem.

Table 4- 1

The Simulation Parameter Used When Solving the Optimization Problem

Parameter	Value	Parameter	Value
Maximum data rate (R_{max})	2000000 bit/s	**Packet length (l)**	50000 bits
Maximum bandwidth(W_{max})	1.5×10^{6}Hz	**Minimum bandwidth (W_{min})**	0 Hz
Time Steps	60 s	**Distortion threshold**	Between 8% and 16%
Noise spectral density (N_0)	-3.98×10^{-21} dBm/Hz	**Doppler frequency**	0.1 Hz
Delay Deadline	0.025 s	$\mathbf{c_1}$	4.9404
$\mathbf{c_2}$	-0.351	**F (filter length)**	2
$\mathbf{c_{11}}$	2.2	$\mathbf{c_{44}}$	1
$\mathbf{c_{22}}$	0.3	$\mathbf{c_{55}}$	3620
$\mathbf{c_{33}}$	1.4752	$\mathbf{c_{66}}$	1.465

4.3 Results and Discussion

4.3.1 Compression using CAE

As mentioned above, our compression approach is based on convolutional autoencoder neural network (CAE). The architecture of the model is highly dependent on the dataset, as each dataset should have its own model. Using the same (CAE) model for different datasets while changing only the bottleneck layer is applicable but it does not give good results.

Our performance metrics are the following:

- Compression ratio (CR): Each dataset has maximum compression ratio based on the size of each dataset. For example, BCI-IV-2a dataset can reach up to 98.4% compression which means we are sending only 1.6% of the data However, 93 % compression is the maximum compression ratio can be achieved on DEAP and BCI-IV-2b datasets. The compression ratio was calculated as mentioned in Chapter 3.
- Percent-root mean square distortion (PRD): O_{Sample} corresponds to the original signal, R_{Sample} corresponds to the reconstructed signal and N is the size of the data.

$$PRD\ (\%) = \sqrt{\frac{\sum_{i=1}^{N}(O_{Sample} - R_{Sample})^2}{\sum_{i=1}^{N} {O_{Sample}}^2}} \times 100\% \tag{4.1}$$

- Mean Absolute Error (MAE): it was used during the fitting of the model, where it was implemented and defined as a new evaluation metric in our model.

$$MAE(\%) = \left| \frac{\sum_{i=1}^{N} O_{Sample} - R_{Sample}}{N} \right| \times 100\% \tag{4.2}$$

4.3.1.1 BCI-IV-2a Dataset

4.3.1.1.1 Compression and Distortion Relation

Figure 12 shows the relation between compression ratio (defined as data saving as stated in equation(3.2) and the distortion of the reconstructed data calculated using percentage root distortion rate (PRD). The distortion of the data is extremely low with respect to the high compression ratios as we managed to send around *2%* (around 98% compression) of the data with distortion less than *1.5%*.

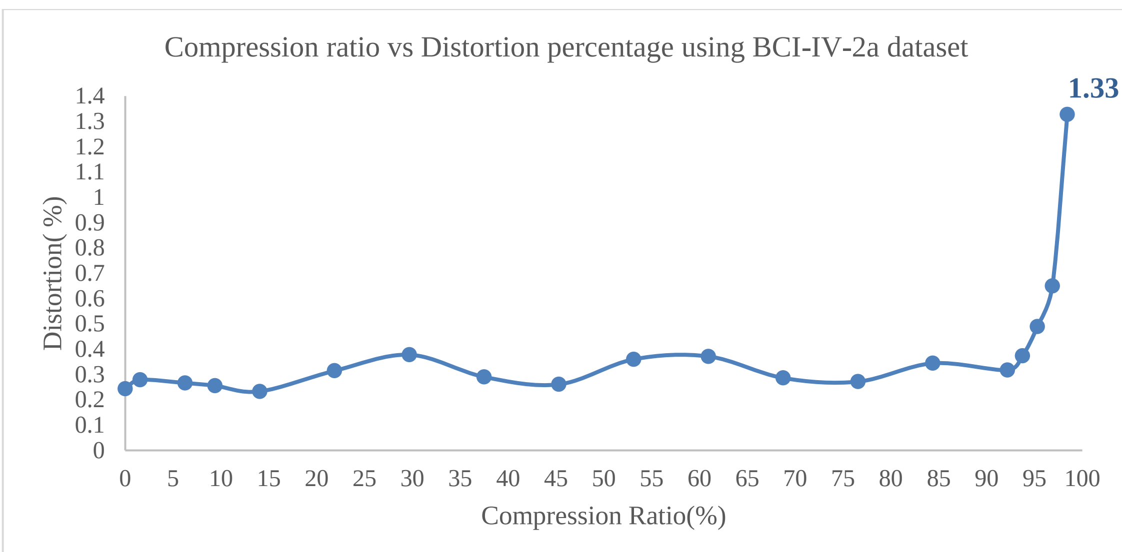

Figure 12: Relation between Compression Ratio and PRD (BCI-IV-2a Dataset)

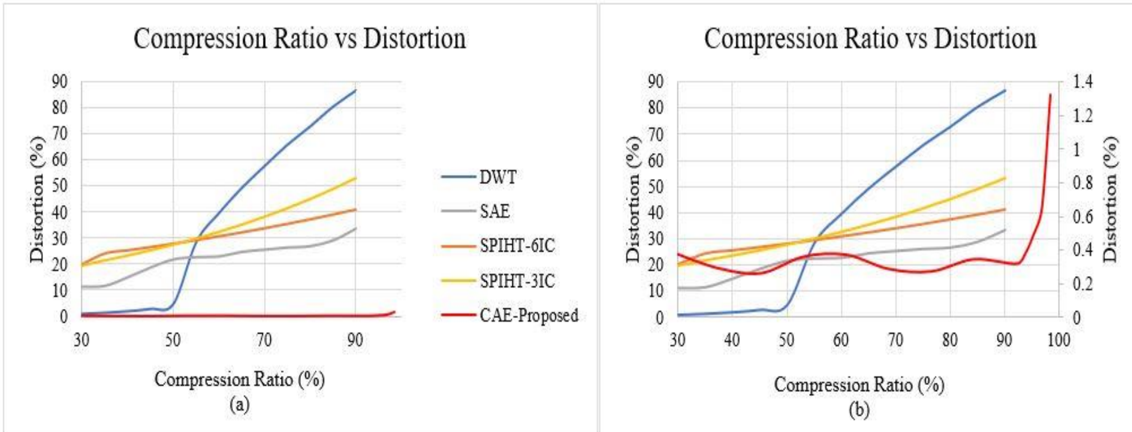

Figure 13: (a) Relation between Compression Ratio and Distortion (PRD) Applied to BCI-IV-2a Dataset. In (b) Another Distortion Scale was Used

Our approach was able to outperform all other state-of-the-art approaches, as a comparison between our proposed CAE model used in BCI-IV-2a dataset with another state of the art techniques used on the same dataset was done. *Figure 13* presents that the values of the distortion (PRD) start to increase at 90% compression using CAE, as it was almost in a steady state before that. The proposed Stacked Autoencoder (SAE) [15] was able to compress up to 90% with a distortion of 33.5%. DWT was able to outperform SAE, SPIHT-3, and SPIHT-6 at low compression ratios as it reaches to 1.5% distortion at 35% compression. However, using CAE was able to achieve 0.3% distortion at 35% compression. This shows the effectiveness of using convolutional layers instead of fully connected layers.

4.3.1.1.2 Compression Sampling Error Rate Relation

In addition to distortion, we also calculated the sample error rate as part of evaluating our model. As in the reconstructed signal, only the samples that have the same value as the original signal will be counted else it will be considered an error. In order to be reliable, we accept 1% more or less than the original sample. *Figure* 14 represents the sample error rate in different compression ratios for one reconstructed signal of the dataset. As it has the same inclination as the compression with distortion. In equation (4.3) N_s is the total number of samples and N_{cs} is the number of correct samples after the reconstruction of the original signal.

$$Sample\ error\ rate(SER) = \frac{N_s - N_{cs}}{N_s} \times 100\% \tag{4.3}$$

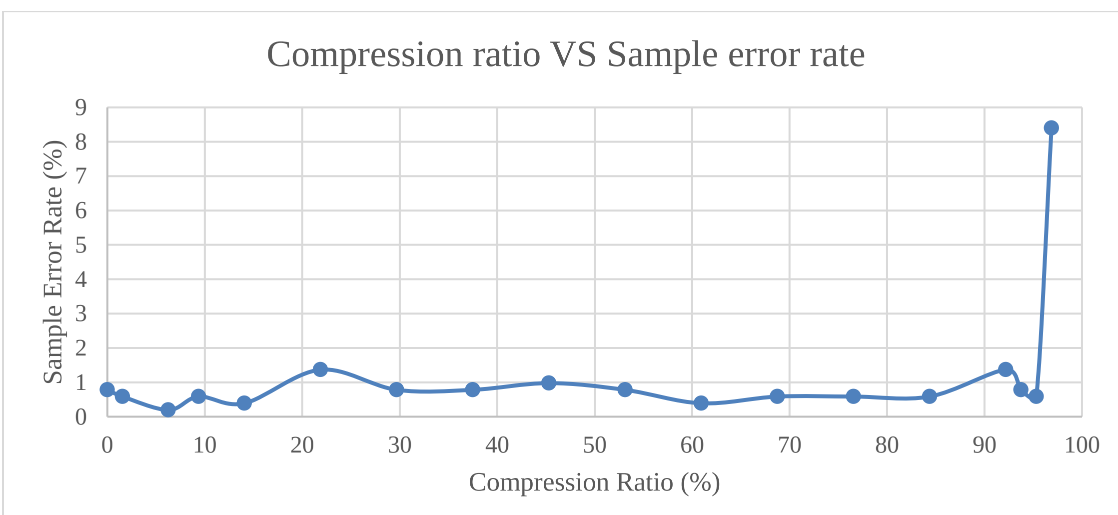

Figure 14: Relation between Compression Ratio and Sample Error Rate (BCI-IV-2a Dataset)

4.3.1.1.3 Number of Parameters

The number of parameters in any neural network should increase with the deeper network. However, our Convolutional autoencoder is used for compression, so the number of parameters decreases until reaching to the bottleneck layer where the number of filters will have a certain value with respect the amount of compression needed. The figure below represents the relation between the number of parameter and compression as well as the number of filters.

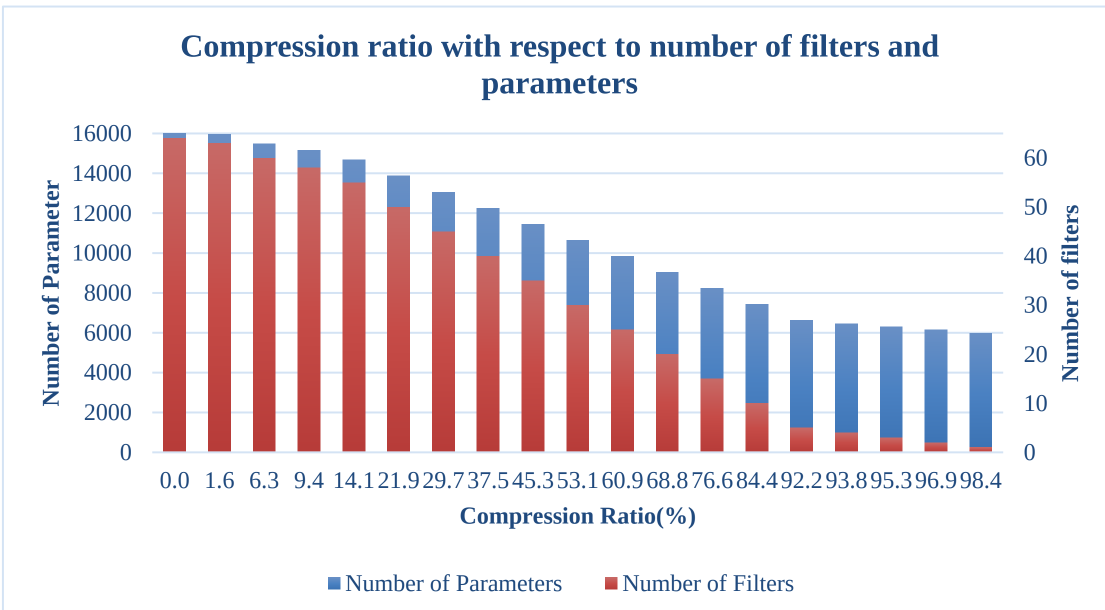

Figure 15: The Relation between Number of Parameter and Compression Ratio as well as the Number of Filters (BCI-IV-2a Dataset)

4.3.1.1.4 Visualization of the Reconstructed Signal

In addition to the Percent-root mean square distortion (PDR) as a performance metric, we tried to visualize any random signal from the dataset and compare it before and after being reconstructed. *Figure* 16 is an example of original and reconstructed signal at *98%* compression. Appendix A includes examples of different compression ratios.

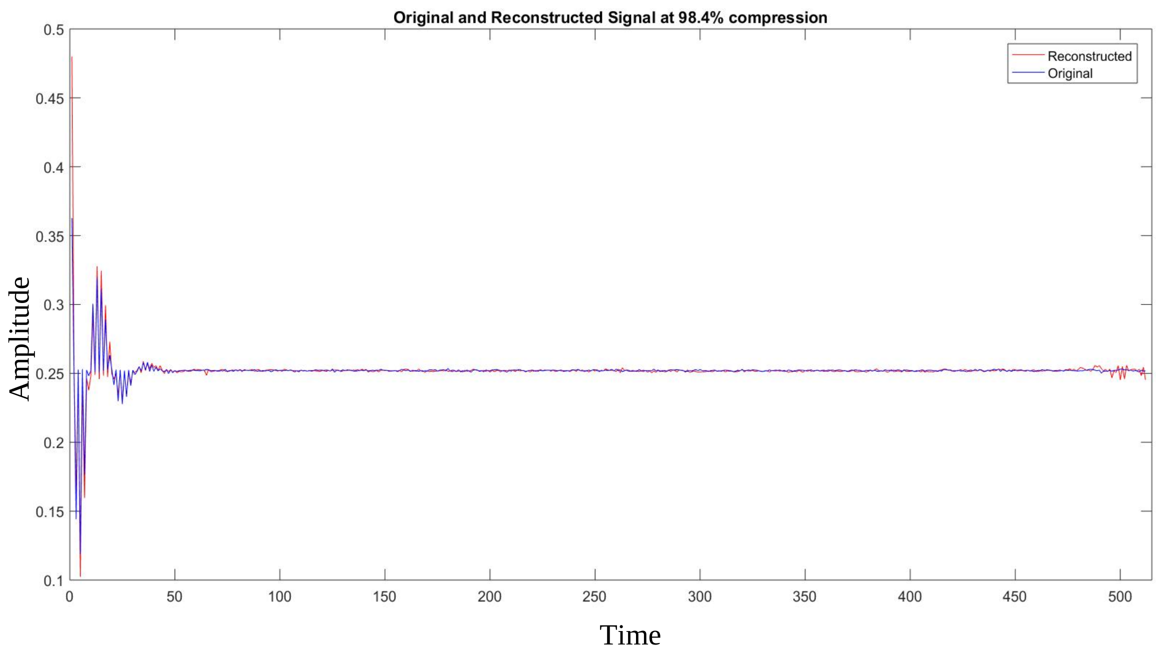

Figure 16: Original and Reconstructed Signal at 98.4% Compression (BCI-IV-2a Dataset)

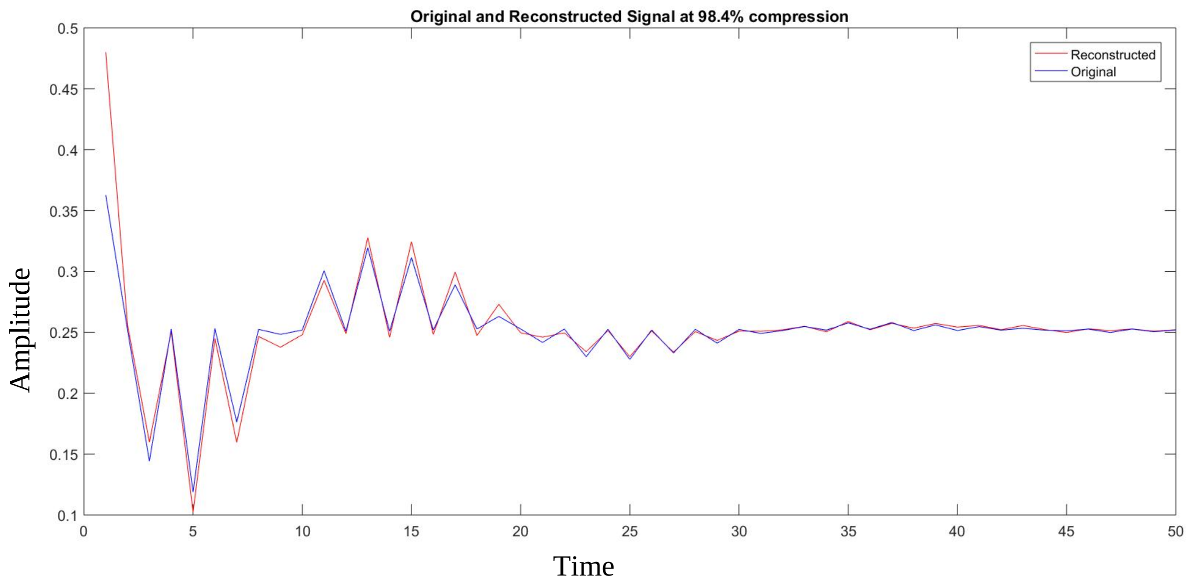

Figure 17: Original and Reconstructed Signal at 98.4% Compression (Zoom-In - BCI-IV-2a Dataset)

4.3.1.2 BCI-IV-2b Dataset

4.3.1.2.1 Compression and Distortion Relation

Figure 18 shows the relation between compression ratio and the distortion of the reconstructed data calculated using PRD. The distortion of the data is low with respect to the high compression ratios but not lower than the distortion achieved on BCI-IV-2a dataset. The proposed CAE model for this dataset managed to achieve around *90%* compression with distortion less than *12%*.

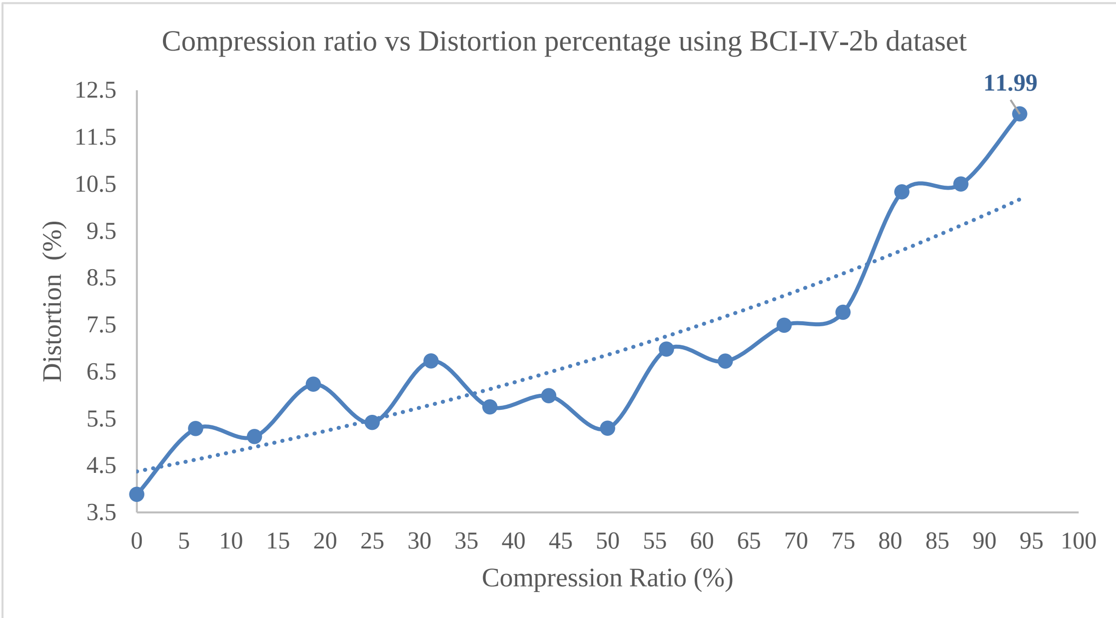

Figure 18: Relation between Compression ratio and Distortion (BCI-IV-2b dataset)

The CAE model used to manage BCI-IV-2b dataset is not the same as the model used for BCI-IV-2a as we explained in section 4.2. Again the power of using convolutional layers instead of fully connected layers can guarantee low distortion at high compression ratio compared with another state of the art approaches. However, experimentally it was found that the efficiency of CAE model at low compression ratio can't always outperform other techniques. As in BCI-IV-2b dataset, DWT was able to compress 30% of the data with 1.6% distortion[15], meanwhile, our CAE model approach achieves 6.7% distortion with 30% compression

4.3.1.2.2 Compression Sampling Error Rate Relation

We calculated the sample error rate using equation (4.3). The behavior represented by *Figure 19* gives the same trend as the relation between the compression and distortion, where the sample error rate increase while increasing the compression, and this gives a good prediction about the visualization results.

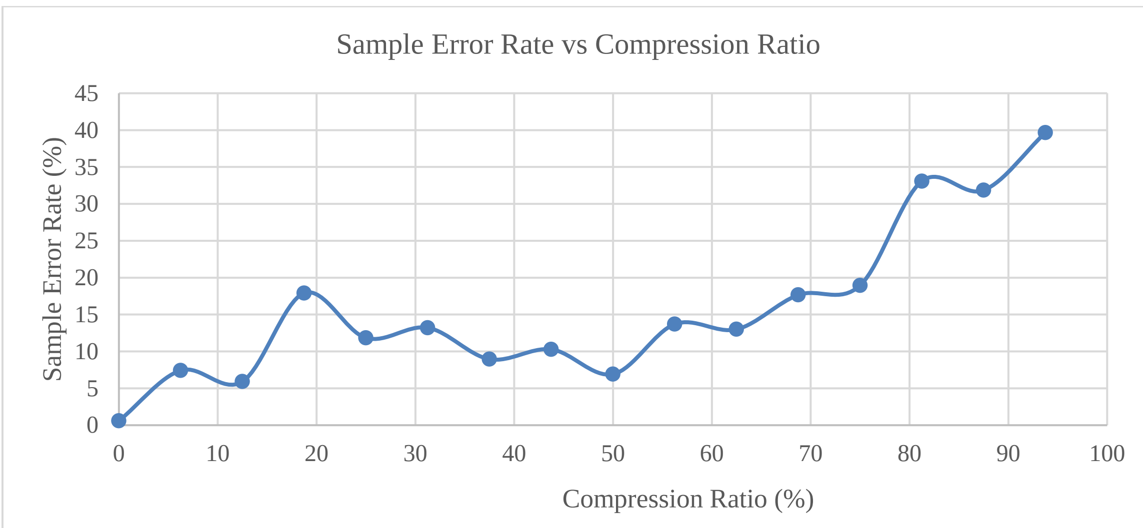

Figure 19: Relation between Compression ratio and Sample Error Rate (BCI-IV-2b dataset)

4.3.1.2.3 Number of Parameters

The number of parameters affected by the number of filters, as the more the filters the more the parameters. *Figure* 20 represents the relation between the number of parameter and compression as well as the number of filters.

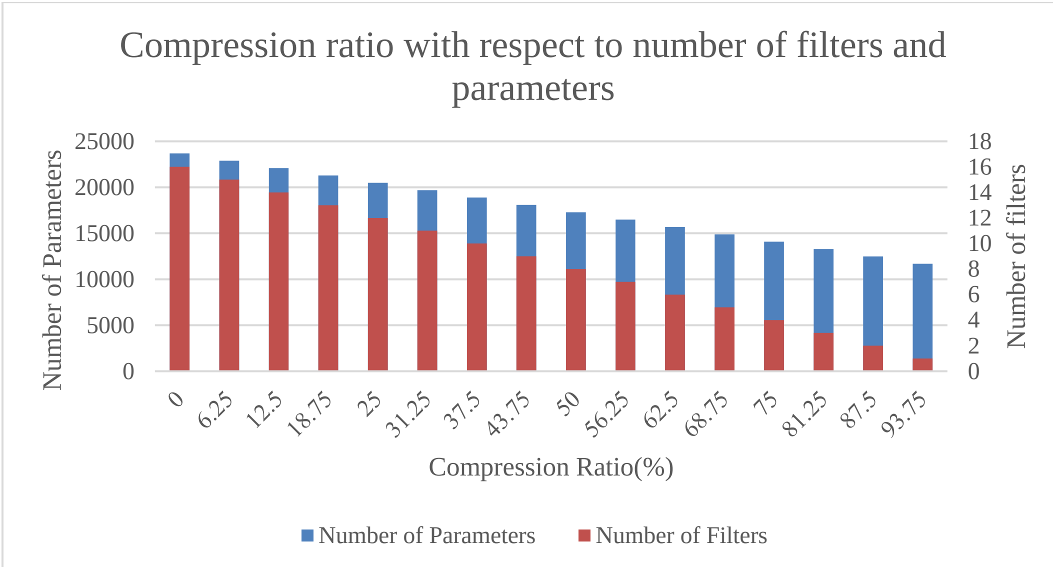

Figure 20: The relation between number of parameter and compression as well as the number of filters (BCI-IV-2b dataset)

4.3.1.2.4 Visualization of the Reconstructed Signal

Figure 21 represents a visualization of a reconstructed signal along with the original signal at compression ratio ~*90%*. The reconstructed signal very close to the original signal with some shifting.

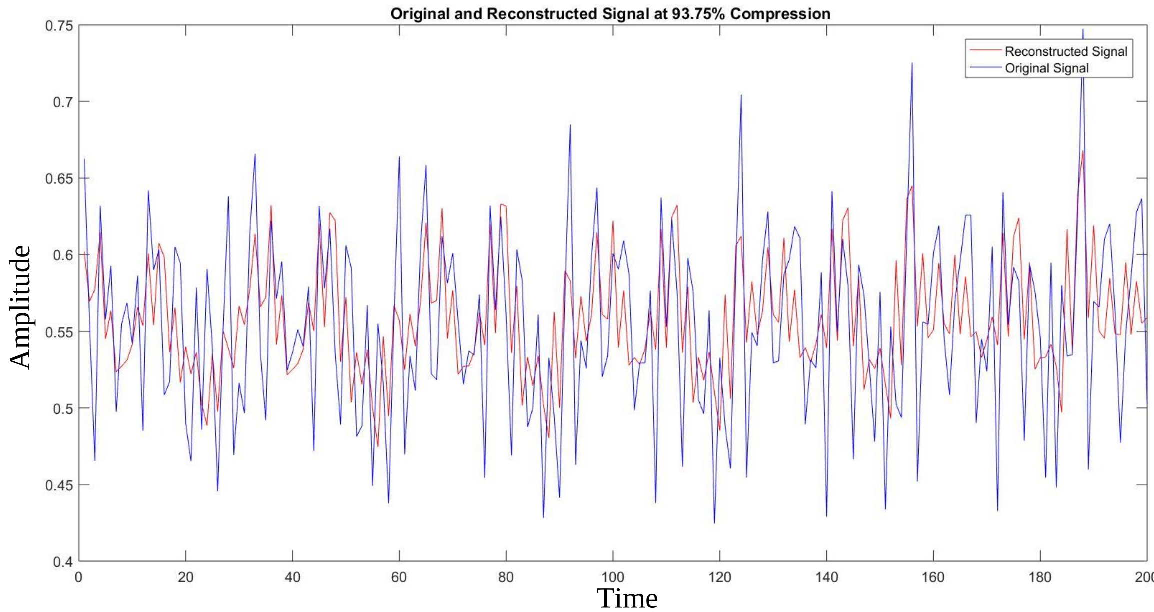

Figure 21: Original and Reconstructed Signal at high compression (Zoom-in BCI-IV-2b)

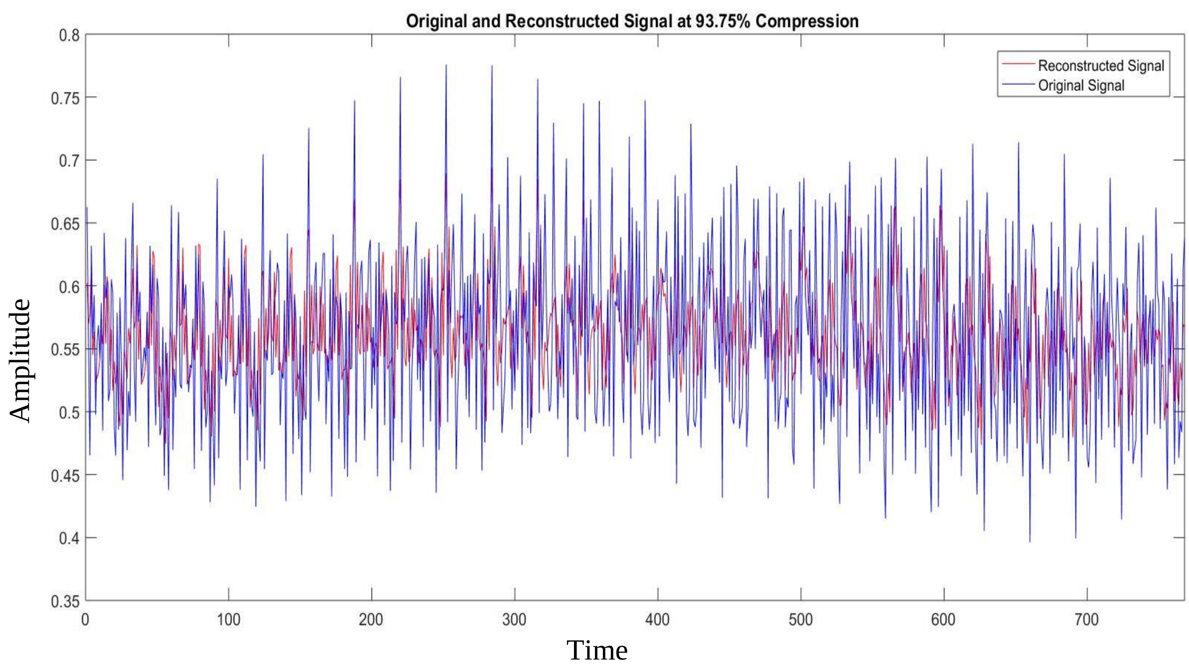

Figure 22: Original and Reconstructed Signal at high compression (BCI-IV-2b)

4.3.1.3 DEAP dataset

This subsection was added to show that using the same CAE model of a certain dataset on another dataset will not give an acceptable result. In the following the CAE model used to compress and reconstruct BCI-IV-2a dataset was applied on DEAP dataset, where only changes to the bottleneck layer were applied to manage a certain amount of compression ratio.

4.3.1.3.1 Compression and Distortion Relation

Figure 23 shows the relation between compression ratio and the distortion of the reconstructed signal. The distortion of the data is low compared to the state of the art methods. Nevertheless, it has higher distortion than the distortion in BCI datasets. The maximum compression ratio here is *93.75%* where we managed to send around *6%* of the data with distortion *31 %*. Even though the relation looks consistent and gives an acceptable distortion rate. However, evaluating the performance through the sample error rate and visualization of the signal gives bad inconsistent results, the coming subsection will show it.

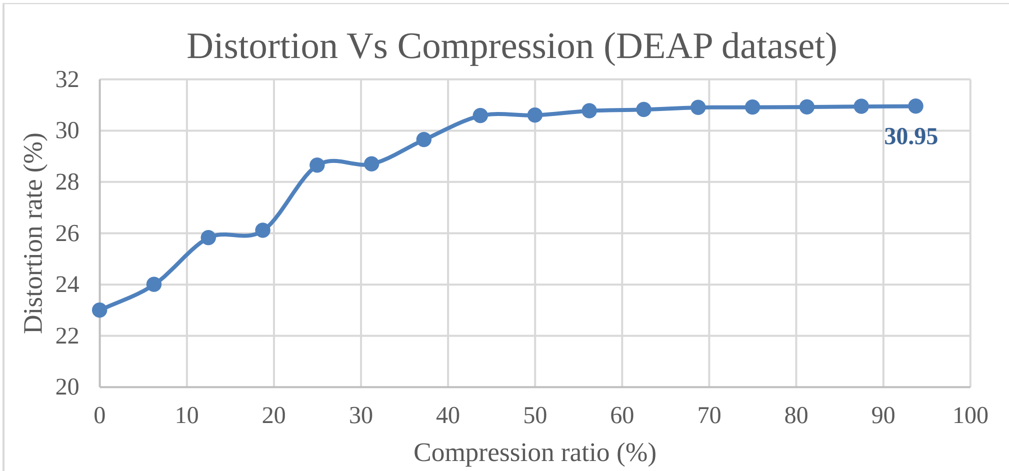

Figure 23: Relation between Compression Ratio and Distortion (DEAP Dataset)

4.3.1.3.2 Compression Sampling Error Rate Relation

The sample error rate has an inconsistent trend. We have used equation (4.3) to calculate the sample error rate. However, in this relation, we consider an error of $\pm$ 10%, as we will not be able to draw the relation with less error percentage. The behavior here can give a prediction about the bad visualization results of the reconstructed signal. *Figure 24* shows the relation between the compression ratio and the sample error rate.

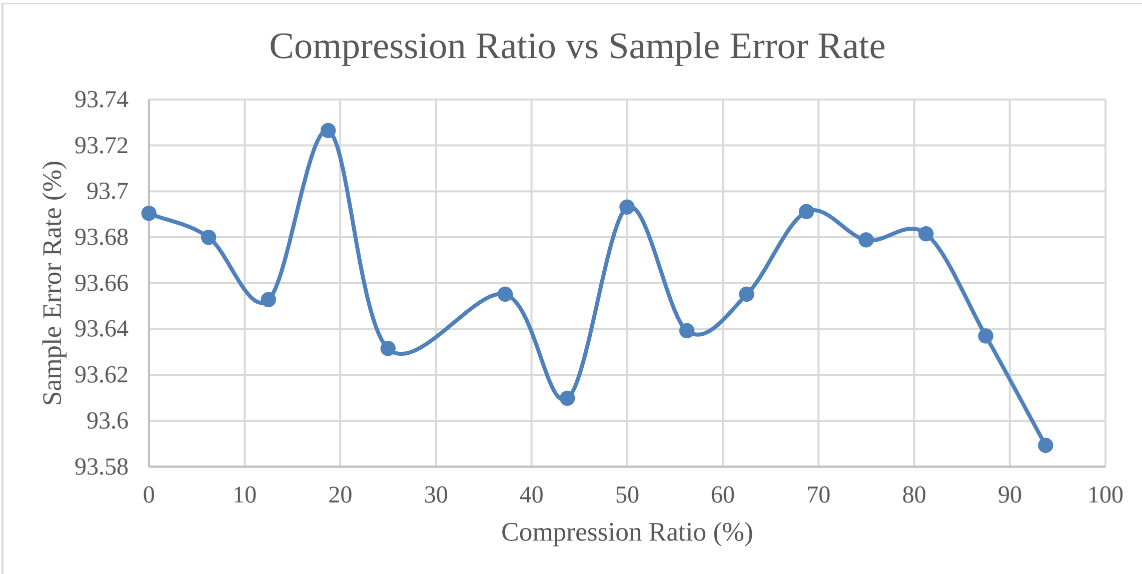

Figure 24: Relation between Compression Ratio and Sample Error Rate (DEAP Dataset)

4.3.1.3.3 Number of Parameters

As mentioned before the number of parameters decrease as the number of filters decrease and vice versa. *Figure* 25 denotes the relation between the number of parameter and compression as well as the number of filters.

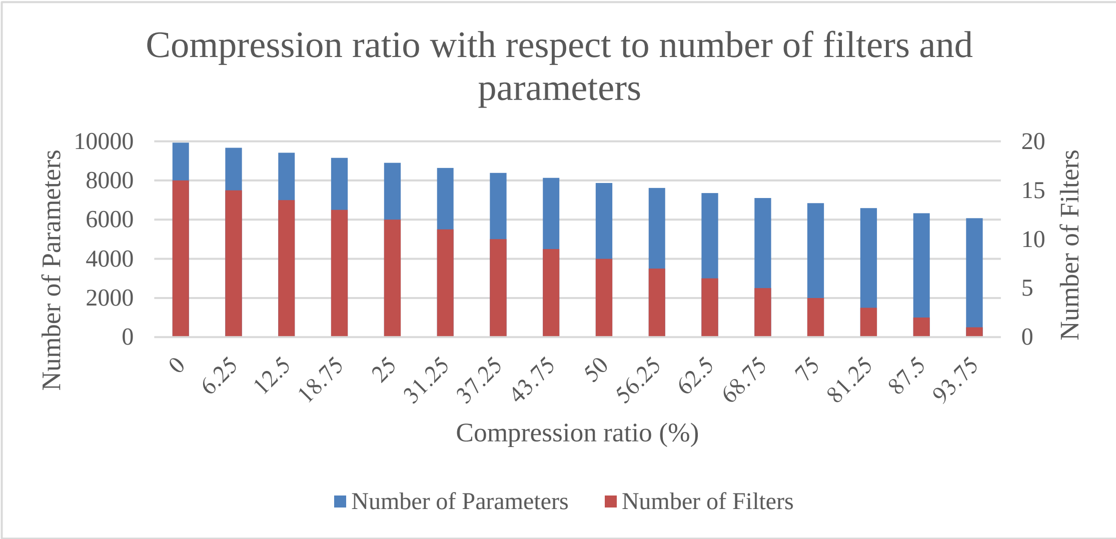

Figure 25: The Relation between Number of Parameter and Compression as well as the Number of Filters (DEAP Dataset)

4.3.1.3.4 Visualization of the Reconstructed Signal

Figure 26 represents a visualization of a reconstructed signal along with the original signal at compression ratio *43.75%*. The signal was destroyed even if the *PRD* was giving an acceptable value compared with other technique.

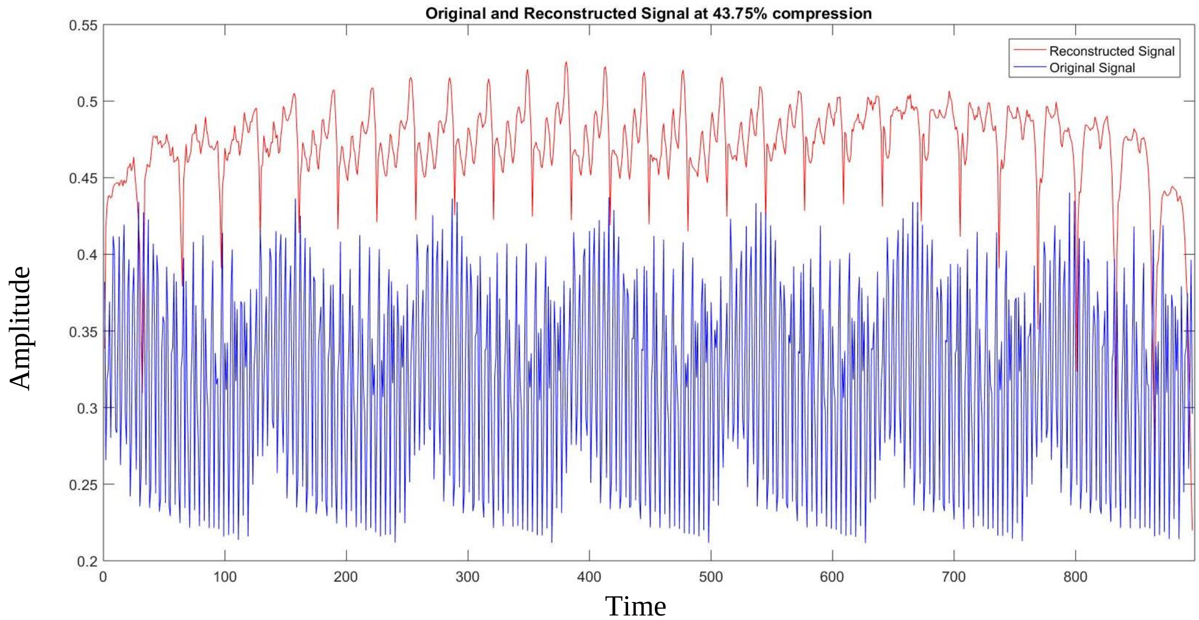

Figure 26: Original and Reconstructed Signal at 43.75% Compression (DEAP Dataset)

4.3.1.4 Summary

Deep learning is a very effective approach that can be used in medical signal processing. As discussed in the results Chapter, deep learning and particularly Convolutional Autoencoder, unlike any other signal compression approaches as a certain model should be used for specific dataset and application. Using the same model for different datasets will not achieve the same performance.

Our models used for BCI datasets were able to outperform the proposed method in [15] (DLDC) and though all other techniques such as DWT, 2D-SPIHT-3-ICs, and 2D-SPIHT-6-ICs.

CAE model has low distortion rate in both high and low compression ratios, unlike DLDC. Here are some observations about deep learning we found during the experimental work for building the model:

- Deep learning is different from all other signal processing and compression techniques, as it applies compression based on previous learning phase.
- The more training applied to the neural network with more data, the less the distortion, and this is the known rule in deep learning, which was proved through the experimental phase of building the model.
- Deep learning is application specific, i.e. not only the type of the data affects its architecture, parameters, and performance; but also, the dataset itself, and hence the application where the dataset is used for. This point was proved by using the model of BCI-IV-2a dataset with DEAP dataset, where we end up with very bad results, especially in visualization.
- Using deep learning may lead to maximum compression ratio with extremely low distortion for some applications, using one or two model configurations. For example, in the case of BCI-IV-2a dataset, the model configurations can be used into the network without adaptive compression and regardless of the network state; where we apply the maximum compression ratio at any time, *Figure 12*. However, for other datasets like BCI-IV-2b, *Figure 18*, where we have good results but not optimized. An optimization with respect to different network states should be applied to adjust the compression ratio with respect to the network state and the application requirements.

4.3.2 Optimization of Resource Allocation in Wireless Environment

The simulation results were generated using the following network topology: 4, 6 and 10 patients (user) moving into 4 main zones based on STEPS mobility model; where the minimum speed of a patient equals to 2 m/s, the maximum speed 6 m/s.

The simulation duration equals 60-time slots. Each patient has some sensor nodes that collect the data and send them to the PDA. Flat Rayleigh fading with Doppler frequency of 0.1 Hz is used to model small-scale channel variations. Each user will have a different distance from the base station during the 60-time slots. However, all users have the same data length and the same delay deadline. The weighting factor used in these runs will vary between 0 and 1, where 1 means that transmission energy is only considered while 0 means that full privilege is given to distortion.

The distortion relation with respect to the transmitted data which is inferred from BCI-IV-2b dataset was used in the optimization problem.

For the distortion threshold, the simulation will solve the optimization problem for 5 different distortion thresholds starting from 8 up to16 by incrementing the threshold by 2 at each iteration; which means that the optimization problem will be solved for 300 times and it consumes different time based on the number of users. Running the optimization problem for 4 users takes around 10 minutes running while having 10 users takes around 80 minutes.

4.3.2.1 The Effect of Changing the Weighting Factor on the Optimization Problem

Changing the weighting factor will affect the priority of minimizing the transmission energy and distortion of the reconstructed signal. In this subsection, the effect of changing the weighting factor on the distortion, transmission energy, and compression ratio at different distortion thresholds will be explained.

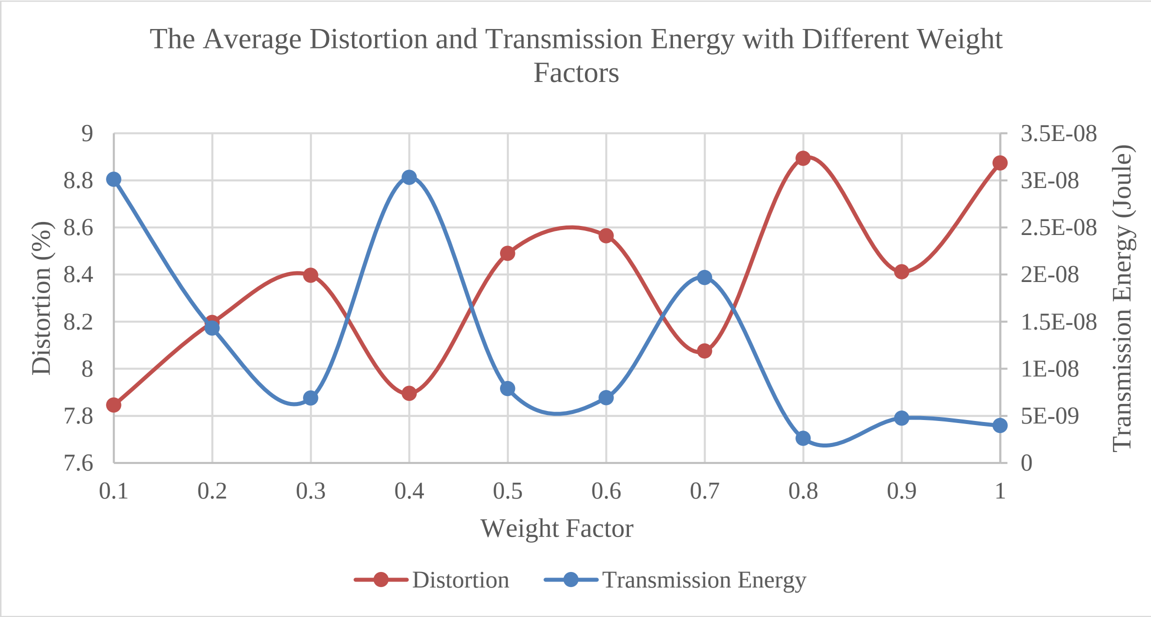

Figure 27: Relation between the Average Transmission Energy and the Average Distortion at Different Weighting Factors

Figure 27 gives a clear conclusion that the transmission energy and the distortion are two conflicting objectives; hence our proposed optimization will try to address the trade-off between the two objectives. The value of the transmission energy at 1 weighting factor was minimized to the maximum while the distortion was neglected, and it reached the highest possible value. Meanwhile, when the weighting factor was 0.1 the average distortion reached to the minimum value and the average transmission energy reached to the maximum possible value. *Figure 28* another representation to the relation with respect to compression ratio instead of distortion where the compression decreases while the transmission energy increase. The relation between the compression and distortion is presented by *Figure 29*, where the more we compress the higher the distortion occurred on the reconstructed signals.

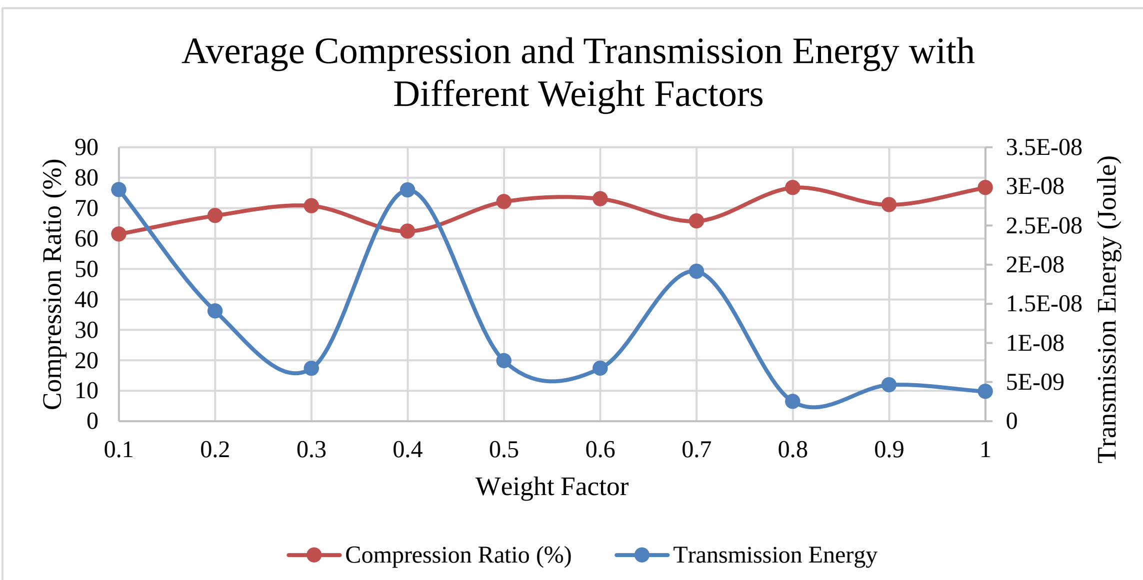

Figure 28: The Relation Between the Average Transmission Energy and the Average Compression for 4 Users at Five Different Distortion Thresholds and at Different Weighting Factors

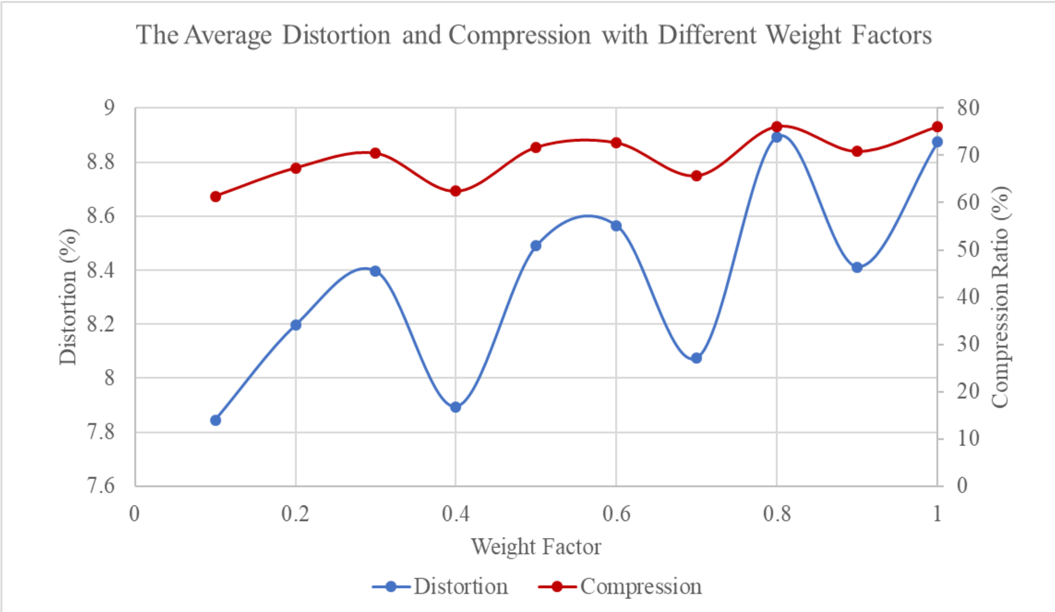

Figure 29: The Relation between the Average Distortion and the Average Compression for 4 Users at Five Different Distortion Thresholds and at Different Weighting Factors

4.3.2.2 Minimizing the Transmission Energy and Distortion with and without Considering Network Bandwidth.

In this study, the bandwidth was considered in the optimization problem as a decision parameter where the users need to use a certain amount of the total bandwidth based on the channel given to the users and the aggregate bandwidth of all users should not exceed the bandwidth threshold. Considering the bandwidth for all users should have a great effect, not only on the transmission energy but also the distortion of the reconstructed signal. *Figure 30* shows the average transmission energy and the average distortion for 3 scenarios, where the number of users equals to 4, 6 and 10 users. The effectiveness of minimizing the distortion appears clearly for few users and when the number of users increases the distortion follow the same trend as having a constant bandwidth.

However, this is not the case with transmission energy as considering the bandwidth during the optimization problem; gives a good implication to the system compared with having an equal amount of bandwidth among the users; which decreases when the number of users increases.

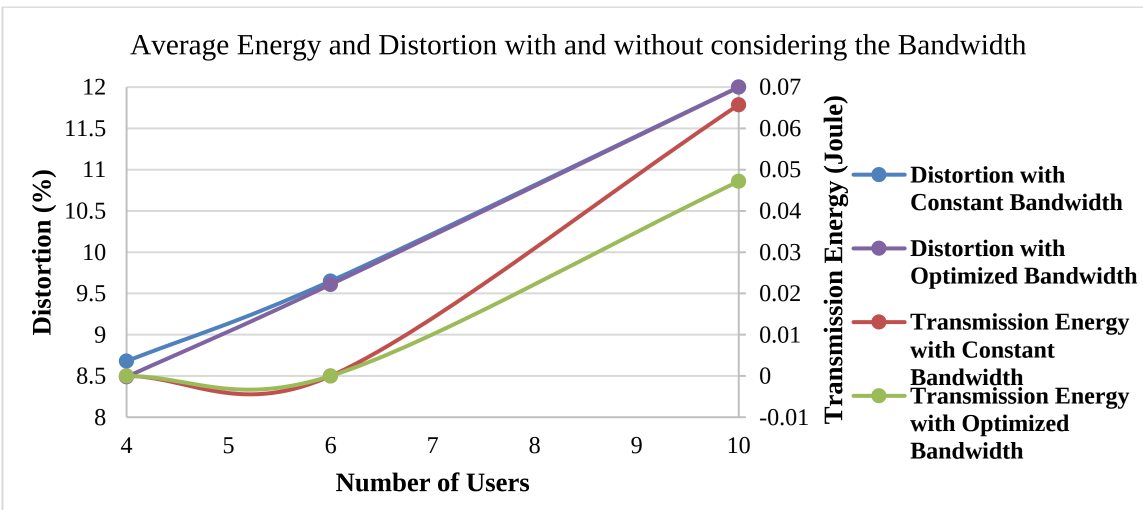

Figure 30: The Average Transmission Energy and Average Distortion for Different Number of Users While Considering and Ignoring the Bandwidth in the Optimization Problem

Figure 31 represents the average transmission energy for three main scenarios: 4, 6 and 10 users while considering and ignoring the bandwidth as a decision parameter of the optimization problem. Considering the bandwidth means providing a certain amount of bandwidth based on the channel of the user, which is mainly affected by the movement of the users with respect to the base station (wireless access point). In most of the cases, the value of the average transmission energy while considering the bandwidth as a decision parameter is less than the value of the average transmission energy with constant bandwidth.

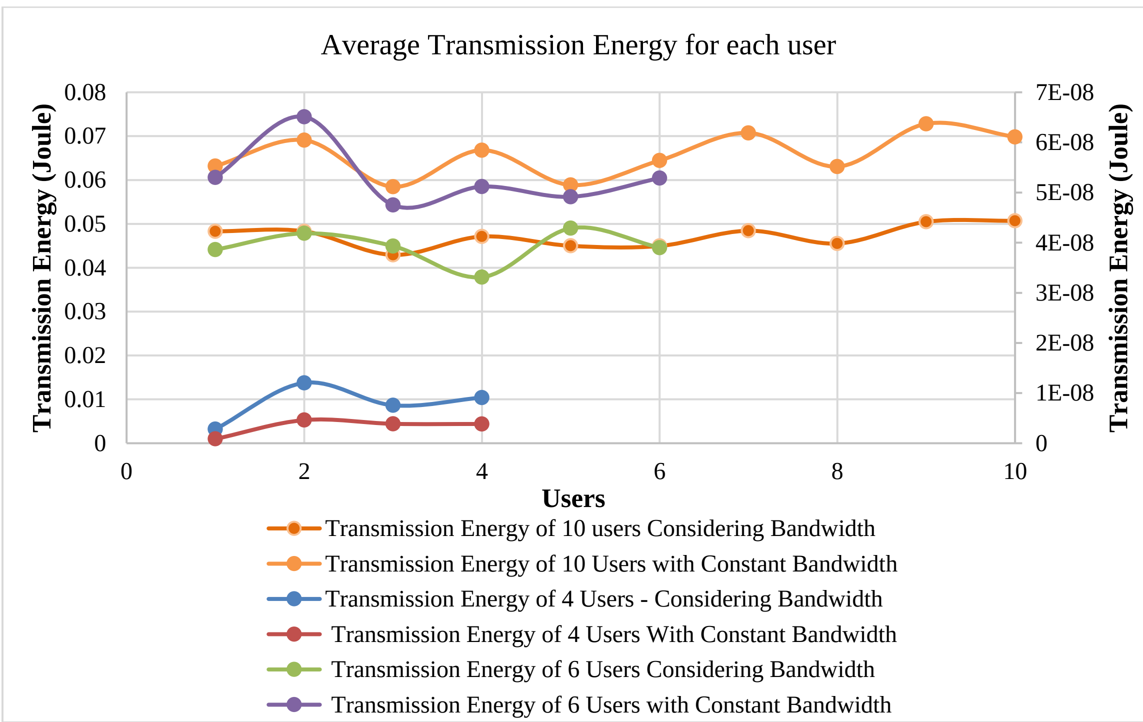

Figure 31: Average Transmission Energy With and Without Considering Bandwidth as a Decision Parameter

Figure 32 shows that using CAE while considering the bandwidth will lead to minimization in the bandwidth consumption for most of the cases. Bandwidth is one of the decision parameters of the optimization problem as mentioned earlier and there are other decision parameters that affect the resource allocation problem such as the data rate, which would cause allocating more bandwidth for certain users.

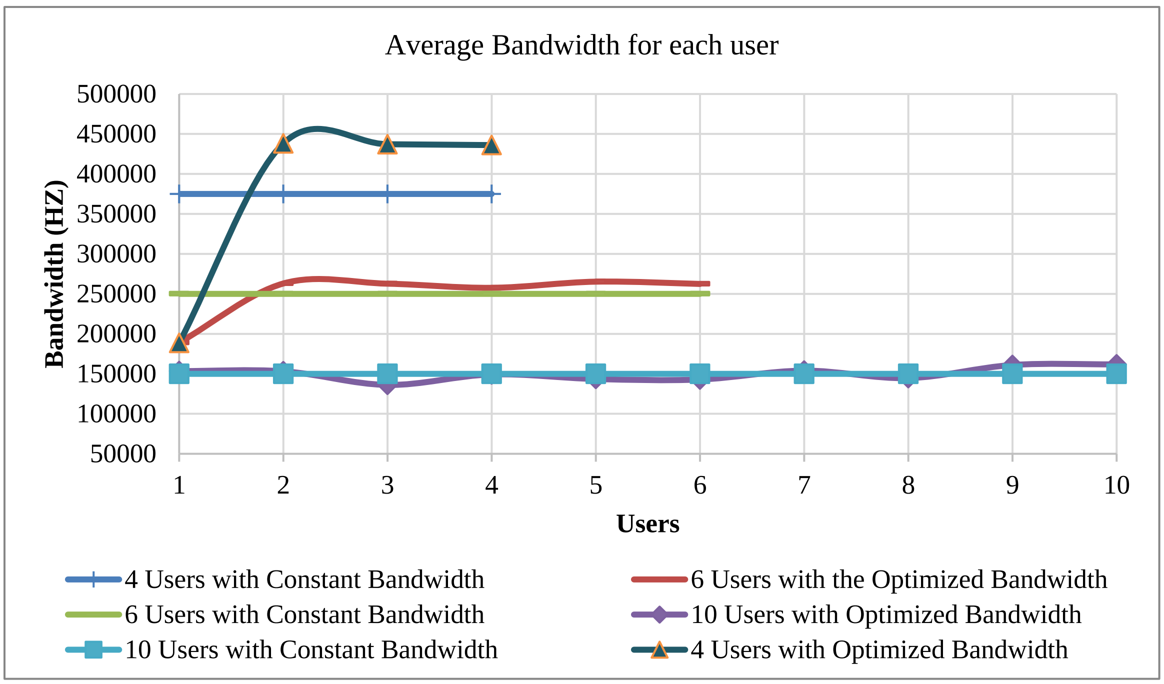

Figure 32: Average Bandwidth for Each User

Figure 33 represents the relation between the transmission energy for 6 users with considering and ignoring the bandwidth. The transmission energy decrease while increasing the distortion threshold, since increasing the distortion threshold will constrain the amount of compression applied as equation (3.15) states. The dashed lines represent users having constant bandwidth, their transmission energy always greater than the case where bandwidth is considered in the resources allocation problem; which increase the efficiency of the system by almost 54.3%

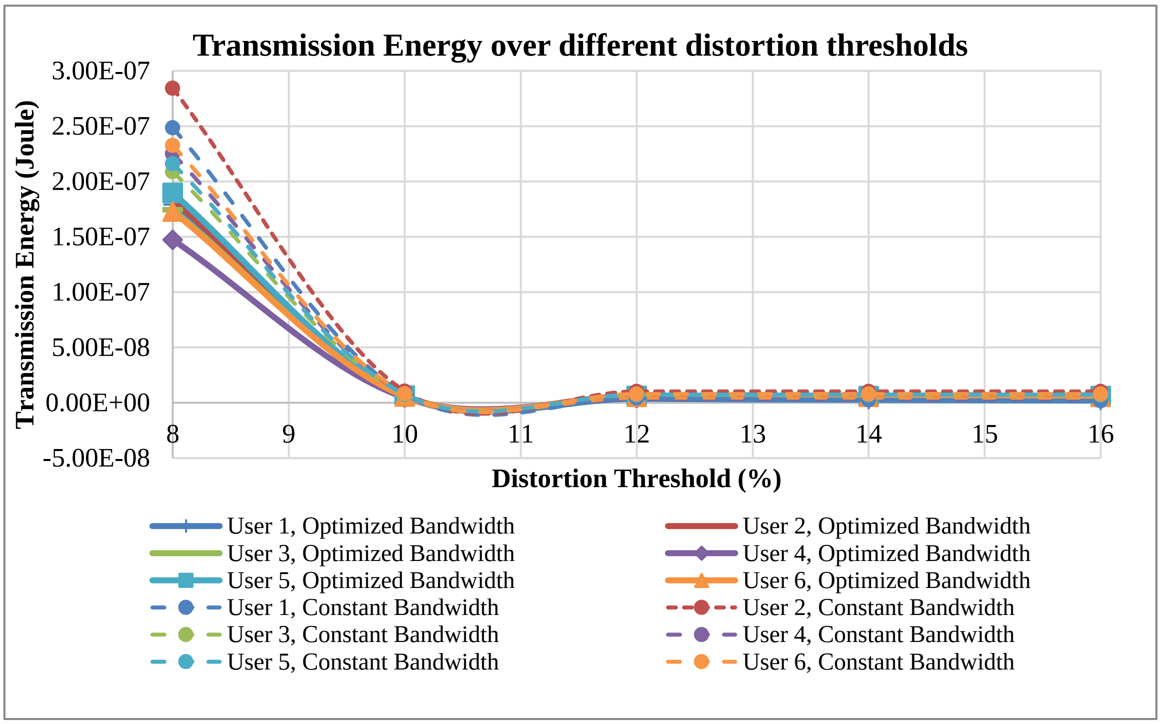

Figure 33: Average Transmission Energy of 6 Users Through 5 Different Distortion Thresholds

4.3.2.3 Optimization study at certain distortion threshold

In this subsection, a study of transmission energy and distortion at 12% distortion threshold (D_{th}) for a different number of users will be presented and discussed. *Figure*

34 presents the trade-off between the transmission energy and distortion at different weighting factors. It's clear that whenever setting the priority to the transmission energy it decreases, and the distortion increases.

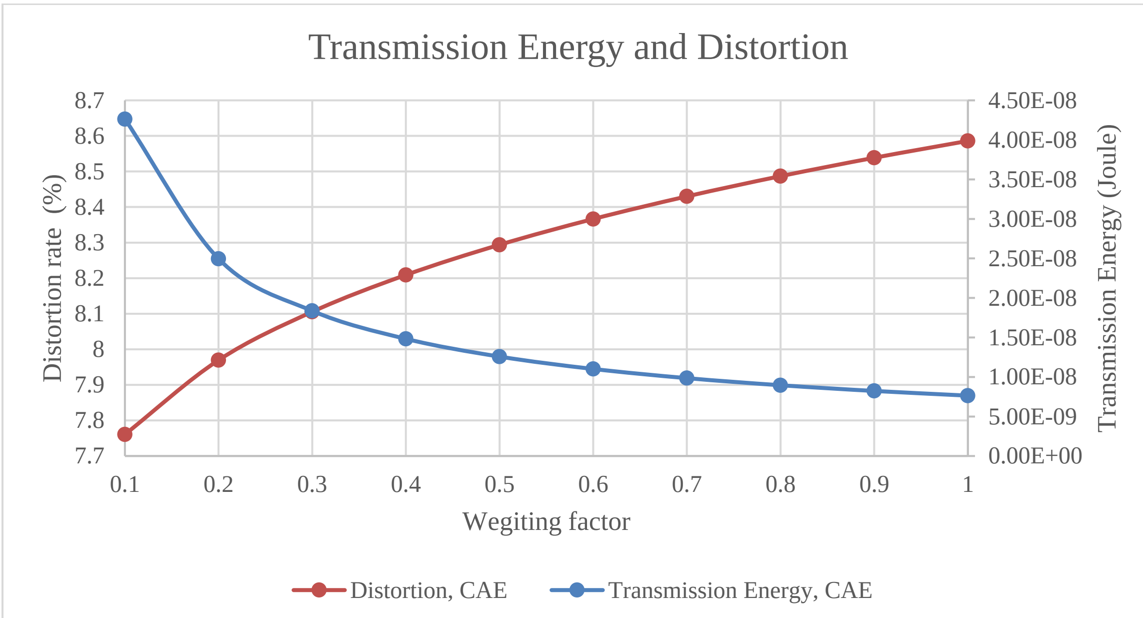

Figure 34: Trade-off between Transmission Energy and Distortion for 4 Users at Distortion Threshold Equals to 12%

Figure 35 and *Figure 36* summarize the relationship between the transmission energy and distortion, where the energy increases while the number of users increases and the distortion has the same behavior as the energy. In *Figure 36*, both the transmission energy and distortion reach to values greater than the ones represented by *Figure 35* which consider the bandwidth as a decision parameter of the optimal resource allocation problem.

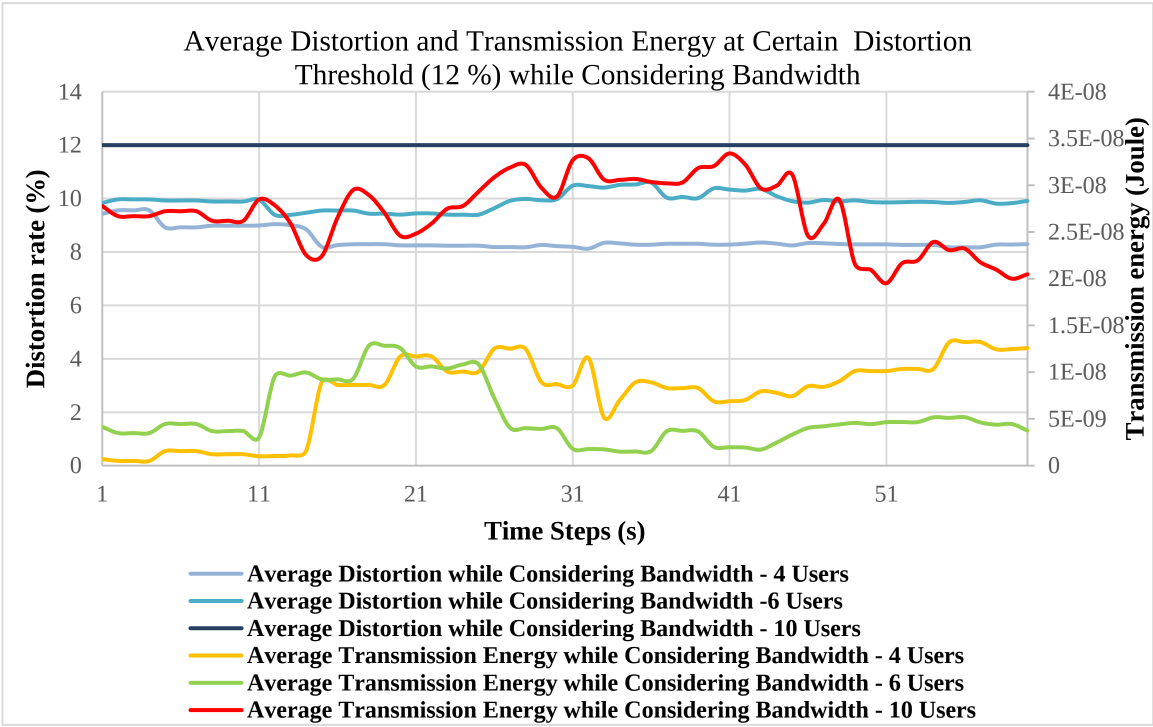

Figure 35: Average Distortion and Transmission Energy at Certain Distortion Threshold (12 %) While Considering the Bandwidth as a Decision Parameter of the Optimization Resource Allocation Problem

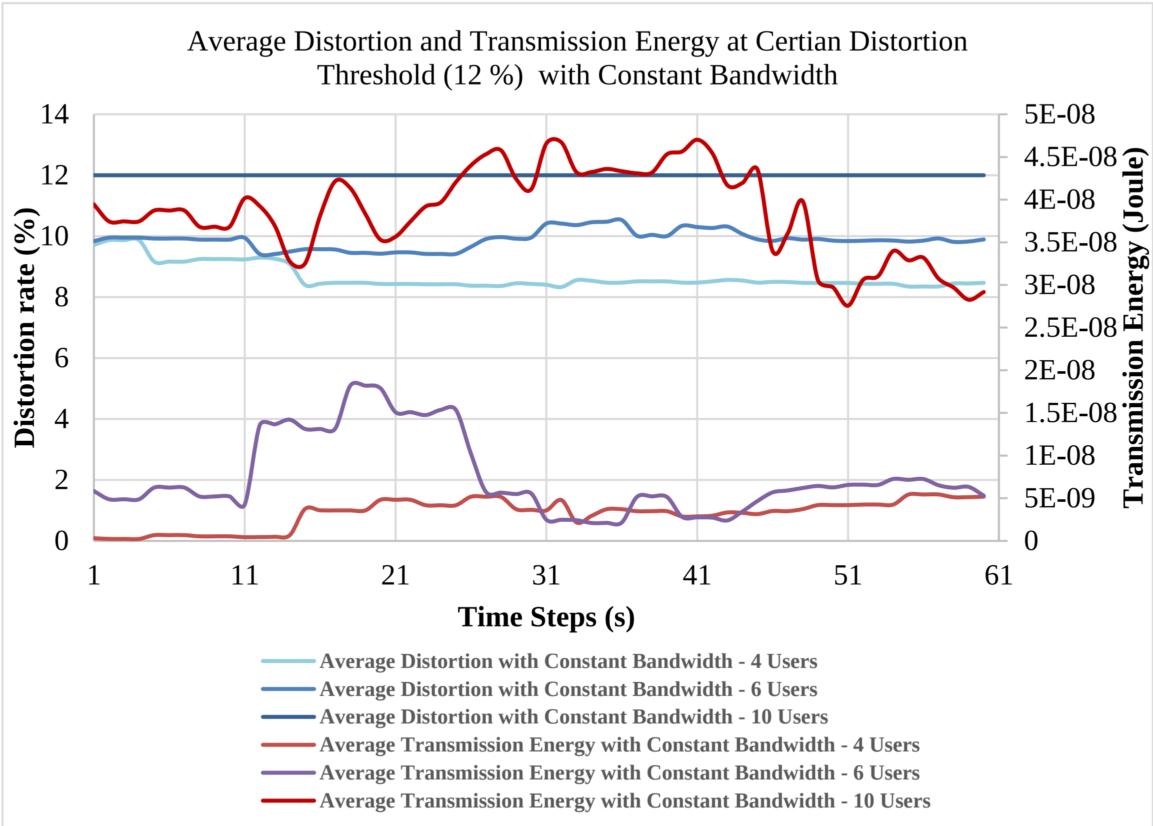

Figure 36: Average Distortion and Transmission Energy at Certain Distortion Threshold (12 %) with Constant Bandwidth for all Users

4.3.2.4 Comparing CAE approach against DWT

In the following subsections, we will discuss the comparison between using CAE and DWT as a compression/reconstruction approach at distortion threshold (D_{th}) equals to 12%. The results show that CAE outperforms DWT as it consumes less transmission energy, more compression, and less distortion compared to DWT.

4.3.2.4.1 Transmission Energy

Figure 37 presents the average transmission energy for both CAE and DWT and it shows that increasing the number of users will increase the transmission energy rapidly using DWT, compared to CAE. (note: - for *Figure 37*, logarithmic scale is used for the y-axis as the difference between energies is really high).

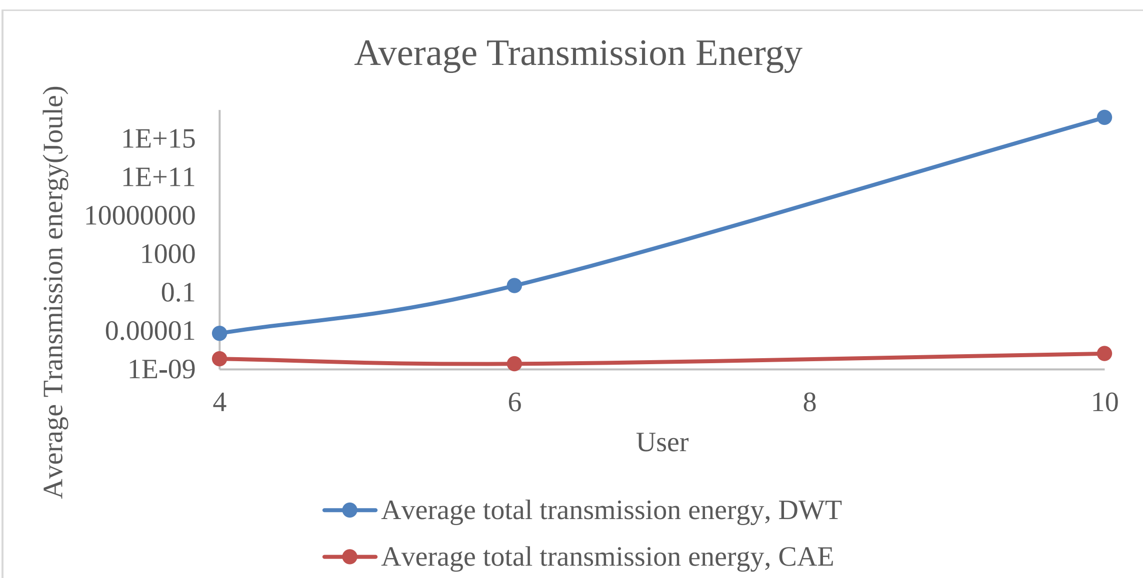

Figure 37: Average Transmission Energy using CAE and DWT Approaches for 4, 6 and 10 Users

Figure 38 shows the transmission energy of 4 users at different weighting factors (w), where it ensures the high performance of CAE against DWT. We are using logarithmic scale for the y-axis in order to represent both in the chart. The energy is minimized significantly using CAE as we increase the weighting factor.

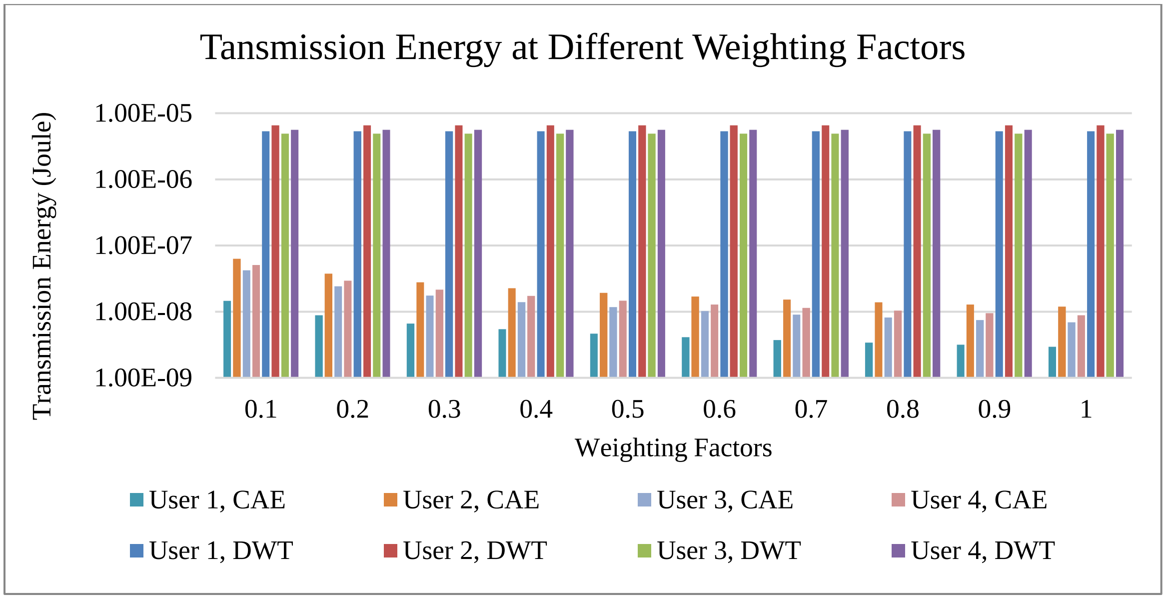

Figure 38: Transmission Energy for DWT and CAE at Different Weighting Factors

4.3.2.4.2 Distortion

Figure 39 shows that setting the distortion threshold to 12% limits the performance of DWT as it reaches the threshold with small compression ratio comparing to CAE, with a different number of users. However, CAE was able to reach distortion less than the threshold while applying high compression ratio like 90%, while the maximum compression ratio for DWT was around 30%.

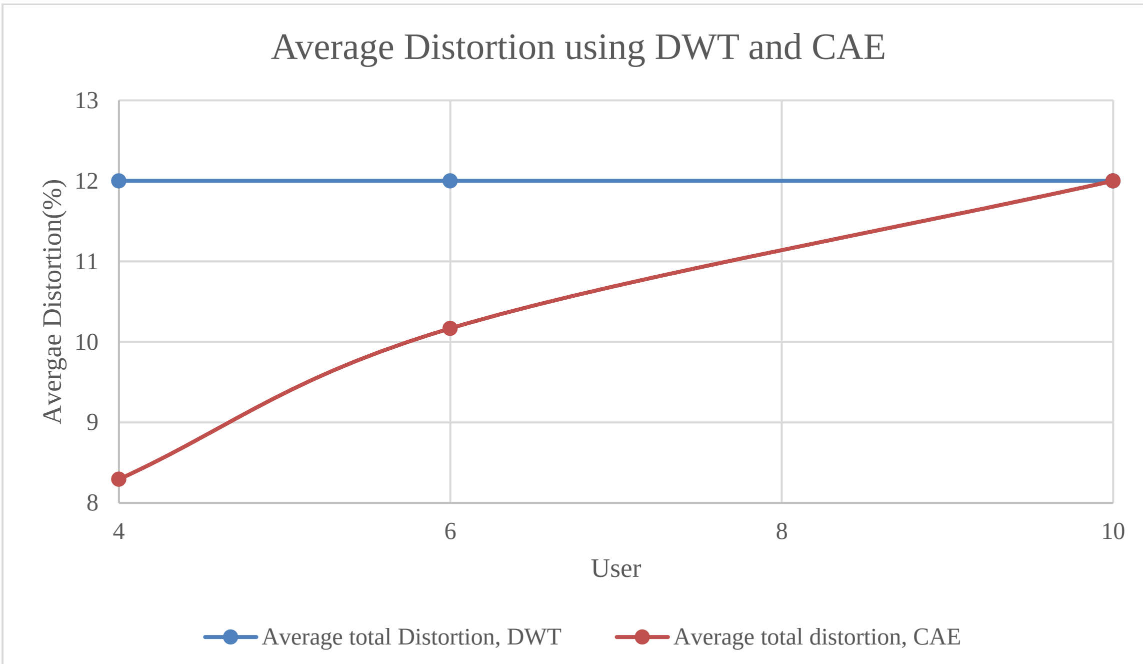

Figure 39: Average Distortion for 4,6 and 10 Users Using CAE and DWT

4.3.2.5 Summary

An optimal network resource allocation with respect to both the network state and the application requirement is an essential target, especially when having a huge amount of medical data that needs to be transmitted and received by the caregiver with the minimum distortion. In this subsection, the results of the allocation problem were presented and explained. The main outcome of this section can be summarized as follows:

- Solving the optimization resource allocation problem with respect to the bandwidth as a decision parameter increases the efficiency of the system by more than 50%. Both the distortion and the transmission will be minimized effectively compared with a scenario that gives the same bandwidth to all user's despite their network state such as the allocated channel state.
- The worse the users' channel, the more allocated bandwidth, and then more transmission energy will be consumed.
- Increasing the compression ratio means increasing the distortion of the reconstructed signal, which implies a decrease in the transmission energy as well as the allocated bandwidth.
- Changing the weighting factor to manage the trade-off between transmission energy and distortion of the reconstructed signal effect, gives a clear conclusion that when giving the highest priority to the transmission energy the distortion will reach to the maximum possible value and vice versa.

CHAPTER 5: CONCLUSION AND FUTURE RESEARCH

In this chapter, we conclude the thesis and provide some ideas for future work that can be done based on this work.

5.1 Conclusion

In this thesis, an adaptive compression technique using convolutional autoencoder model was developed and integrated into a wireless environment, where a huge amount of data needs to be compressed and transmitted through the network based on the network state as well as the application requirements. Reconstructing the compressed data should be within an acceptable range.

An optimization problem to find the trade-off between the transmission energy and distortion was implemented in MATLAB. We found that Deep neural network in general and particularly Convolutional Autoencoder (CAE) can be used efficiently to compress EEG signals, with an excellent performance that outperforms all the state of the art approaches. Using BCI-IV-2a dataset, CAE could send 1.6% of the data and still have a distortion less than 1.5%. CAE, unlike any other signal processing approaches, the same model cannot fit all datasets even though they have the same type of data. CAE considered as dataset and application specific since each dataset needs a certain model.

It is worth mentioning that using deep learning may in some applications lead to maximum compression ratio with extremely low distortion; where minimum data will be transmitted into the network, hence the problem of “adaptive” compression becomes simple to use one maximum compression regardless of the network state. Therefore, resource optimization problem can be solved independently from the adaptive compression, leading to the simpler optimization problem. Nevertheless, this not the case for all models and all datasets as solving the optimization resource allocation problem becomes essential in order to have an adaptive system.

5.2 Future Work

In this thesis, we have considered one type of vital signs which is electroencephalogram (EEG), it would be nice to investigate the abilities of CAE on different types of data e.g. electrocardiogram (ECG), Magnetic resonance (MRI) or X-ray images, etc. This will require redesigning the CNN architecture, and come up with new model configurations, and regression equations similar to equation (3.6) for each data type.

Another important point that would enhance the overall system by adding a classifier using deep learning on the edge level where data with higher priority will be sent first. A different variation of this problem is to design a data reduction technique based on historical conditions, where the classifier may be used to suppress any redundant EEG samples from being transmitted based on the patient condition.

REFERENCES

[1] DeVol, R., Bedroussian, A., Charuworn, A., Chatterjee, A., Kim, I. K., Kim, S., & Klowden, K. (2008). An unhealthy America: the economic burden of chronic disease—charting a new course to save lives and increase productivity and economic growth. Milken Institute. October 2007. URL: http://www. milkeninstitute. org/publications/publications. taf.

[2] Li, K., Shao, M. W., & Wu, W. Z. (2017). A data reduction method in formal fuzzy contexts. International Journal of Machine Learning and Cybernetics, 8(4), 1145-1155.

[3] Quick, D., & Choo, K. K. R. (2016). Big forensic data reduction: digital forensic images and electronic evidence. Cluster Computing, 19(2), 723-740.

[4] Wang, J., Yue, S., Yu, X., & Wang, Y. (2017). An efficient data reduction method and its application to cluster analysis. Neurocomputing, 238, 234-244.

[5] ur Rehman, M. H., Liew, C. S., Abbas, A., Jayaraman, P. P., Wah, T. Y., & Khan, S. U. (2016). Big data reduction methods: a survey. Data Science and Engineering, 1(4), 265-284.

[6] Awad, Alaa, Amal Saad, Ali Jaoua, Amr Mohamed, and Carla-Fabiana Chiasserini. "In-Network Data Reduction Approach Based on Smart Sensing." In Global Communications Conference (GLOBECOM), 2016 IEEE, pp. 1-7. December.2016.

[7] Mahrous, H., & Ward, R. (2016). Block Sparse Compressed Sensing of Electroencephalogram (EEG) Signals by Exploiting Linear and Non-Linear Dependencies. Sensors, 16(2), 201.

[8] Capurro, I., Lecumberry, F., Martín, Á., Ramírez, I., Rovira, E., & Seroussi, G. (2014, September). Low-complexity, multi-channel, lossless and near-lossless EEG compression. In Signal Processing Conference (EUSIPCO), 2014 Proceedings of the 22nd European (pp. 2040-2044). IEEE.

[9] Dufort, G., Favaro, F., Lecumberry, F., Martín, Á., Oliver, J. P., Oreggioni, J., ... & Steinfeld, L. (2016, August). Wearable EEG via lossless compression. In Engineering in Medicine and Biology Society (EMBC), 2016 IEEE 38th Annual International Conference of the (pp. 1995-1998). IEEE.

[10] Hejrati, B., Fathi, A., & Abdali-Mohammadi, F. (2017). A new near-lossless EEG compression method using ANN-based reconstruction technique. Computers in Biology and Medicine.

[11] Sriraam, N. (2009). Context-based near-lossless compression of EEG signals using neural network predictors. AEU-International Journal of Electronics and Communications, 63(4), 311-320.

[12] Sriraam, N. (2012). Correlation dimension based lossless compression of EEG signals. Biomedical Signal Processing and Control, 7(4), 379-388.

[13] Sriraam, N. (2012). A high-performance lossless compression scheme for EEG signals using wavelet transform and neural network predictors. International journal of telemedicine and applications, 2012, 5.

[14] Sriraam, N., & Eswaran, C. (2008). Performance evaluation of neural network and linear predictors for near-lossless compression of EEG signals. IEEE Transactions on Information Technology in Biomedicine, 12(1), 87-93.

[15] Said, A. B., Mohamed, A., & Elfouly, T. (2017, June). Deep learning approach for EEG compression in mHealth system. In IEEE Wireless Communications and Mobile Computing Conference (IWCMC), 2017 13th International (pp. 1508-1512).

[16] Said, A. B., Mohamed, A., Elfouly, T., Harras, K., & Wang, Z. J. (2017, March). Multimodal deep learning approach for joint EEG-EMG data compression and classification. In IEEE Wireless Communications and Networking Conference (WCNC), 2017 (pp. 1-6).

[17] Xu, G., Han, J., Zou, Y., & Zeng, X. (2015). A 1.5-D multi-channel EEG compression algorithm based on NLSPIHT. IEEE Signal Processing Letters, 22(8), 1118-1122.

[18] R. Hussein, A. Mohamed, and M. Alghoniemy, “Scalable real-time energy-efficient EEG compression scheme for a wireless body area sensor network,” Biomedical Signal Processing and Control, 2015.

[19] Daou, H., & Labeau, F. (2014). Dynamic dictionary for combined EEG compression and seizure detection. IEEE Journal of biomedical and health informatics, 18(1), 247-256.

[20] Nguyen, B., Nguyen, D., Ma, W., & Tran, D. (2017). Investigating the possibility of applying EEG lossy compression to EEG-based user authentication. 2017 International Joint Conference on Neural Networks (IJCNN). doi:10.1109/ijcnn.2017.7965839.

[21] Raj, C. M., & Harsha, A. (2016, August). Study on wavelet spectral band based EEG compression. In IEEE Data Science and Engineering (ICDSE), 2016 International Conference on (pp. 1-5).

[22] Karimu, R. Y., & Azadi, S. (2016). Lossless EEG Compression Using the DCT and the Huffman Coding.

[23] Srinivasan, K., Dauwels, J., & Reddy, M. R. (2013). Multichannel EEG compression: Wavelet-based image and volumetric coding approach. IEEE Journal of biomedical and health informatics, 17(1), 113-120.

[24] Alsenwi, M., Ismail, T., & Mostafa, H. (2016, December). Performance analysis of hybrid lossy/lossless compression techniques for EEG data. In Microelectronics (ICM), 2016 28th International Conference on (pp. 1-4).

[25] Capurro, I., Lecumberry, F., Martín, A., Ramírez, I., Rovira, E., & Seroussi, G. (2017). Efficient sequential compression of multi-channel biomedical signals. IEEE journal of biomedical and health informatics.

[26] Awad, A., Mohamed, A., & Chiasserini, C. F. (2017). Dynamic Network Selection in Heterogeneous Wireless Networks: A user-centric scheme for improved delivery. IEEE Consumer Electronics Magazine, 6(1), 53-60.

[27] Doan, T. T., & Beck, C. L. (2017). Distributed Lagrangian methods for network resource allocation. 2017 IEEE Conference on Control Technology and Applications (CCTA). doi:10.1109/ccta.2017.8062536

[28] Trestian, R., Ormond, O., & Muntean, G. M. (2014). Enhanced power-friendly access network selection strategy for multimedia delivery over heterogeneous wireless networks. IEEE Transactions on Broadcasting, 60(1), 85-101.

[29] Naghavi, P., Rastegar, S. H., Shah-Mansouri, V., & Kebriaei, H. (2016). Learning RAT selection game in 5G heterogeneous networks. IEEE Wireless Communications Letters, 5(1), 52-55.

[30] Niyato, D., & Hossain, E. (2009). Dynamics of network selection in heterogeneous wireless networks: An evolutionary game approach. IEEE transactions on vehicular technology, 58(4), 2008-2017.

[31] Andrews, J. G., Singh, S., Ye, Q., Lin, X., & Dhillon, H. S. (2014). An overview of load balancing in HetNets: Old myths and open problems. IEEE Wireless Communications, 21(2), 18-25.

[32] Stevens-Navarro, E., Lin, Y., & Wong, V. W. (2008). An MDP-based vertical handoff decision algorithm for heterogeneous wireless networks. IEEE Transactions on Vehicular Technology, 57(2), 1243-1254.

[33] El Helou, M., Ibrahim, M., Lahoud, S., Khawam, K., Mezher, D., & Cousin, B. (2015). A network-assisted approach for RAT selection in heterogeneous cellular networks. IEEE Journal on Selected Areas in Communications, 33(6), 1055-1067.

[34] Sun, C., Stevens-Navarro, E., & Wong, V. W. (2008). A Constrained MDP-Based Vertical Handoff Decision Algorithm for 4G Wireless Networks. 2008 IEEE International Conference on Communications. doi:10.1109/icc.2008.415

[35] Wu, Q., Du, Z., Yang, P., Yao, Y. D., & Wang, J. (2016). Traffic-aware online network selection in heterogeneous wireless networks. IEEE Transactions on Vehicular Technology, 65(1), 381-397.

[36] Awad, A., Mohamed, A., & Chiasserini, C. F. (2016, April). User-centric network selection in multi-RAT systems. In Wireless Communications and Networking Conference Workshops (WCNCW), 2016 IEEE (pp. 97-102). IEEE.

[37] Chang, Z., Wang, Z., Guo, X., Han, Z., & Ristaniemi, T. (2017, May). Energy efficient and distributed resource allocation for wireless powered OFDMA multi-cell networks. In Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt), 2017 15th International Symposium on (pp. 1-6). IEEE.

[38] Kashef, M., Ismail, M., Abdallah, M., Qaraqe, K. A., & Serpedin, E. (2016). Energy efficient resource allocation for mixed RF/VLC heterogeneous wireless networks. IEEE Journal on Selected Areas in Communications, 34(4), 883-893.

[39] Huberman, B. A., & Sharma, P. (2016, November). COMPARE: COMParative Advantage driven REsource allocation for Virtual Network Functions. In Network Function Virtualization and Software Defined Networks (NFV-SDN), IEEE Conference on (pp. 212-218). IEEE.

[40] Morvari, F., & Ghasemi, A. (2016). Two-stage resource allocation for random access M2M communications in LTE network. IEEE Communications Letters, 20(5), 982-985.

[41] Kuo, F. C., Ting, K. C., Wang, H. C., & Tseng, C. C. (2017). On Demand Resource Allocation for LTE Uplink Transmission Based on Logical Channel Groups. Mobile Networks and Applications, 1-12.

[42] Kuo, F., Wang, H., Ting, K., Tseng, C., & Liu, P. (2012). Robust LTE uplink scheduling based on call admission control. 2012 International Conference on ICT Convergence (ICTC). doi:10.1109/ictc.2012.6387195

[43] Kuo, F. C., Ting, K. C., Wang, H. C., Tseng, C. C., & Chen, M. W. (2017). Differentiating and scheduling LTE uplink traffic based on exponentially weighted moving average of data rate. Mobile Networks and Applications, 22(1), 113-124.

[44] Beck, A., Nedic, A., Ozdaglar, A., & Teboulle, M. (2014). An $ O (1/k) $ Gradient Method for Network Resource Allocation Problems. IEEE Transactions on Control of Network Systems, 1(1), 64-73.

[45] Ng, D. W. K., Lo, E. S., & Schober, R. (2016). Multi-objective resource allocation for secure communication in cognitive radio networks with wireless information and power transfer. IEEE transactions on vehicular technology, 65(5), 3166-3184.

[46] Abdulghani, A. M., Casson, A. J., & Rodriguez-Villegas, E. (2012). Compressive sensing scalp EEG signals: implementations and practical performance. Medical & biological engineering & computing, 50(11), 1137-1145.

[47] Majumdar, A., & Ward, R. K. (2015). Energy efficient EEG sensing and transmission for wireless body area networks: A blind compressed sensing approach. Biomedical Signal Processing and Control, 20, 1-9. doi:10.1016/j.bspc.2015.03.002

[48] Kim, H., Kim, Y., & Yoo, H. J. (2008, August). A low-cost quadratic level ECG compression algorithm and its hardware optimization for body sensor network system. In Engineering in Medicine and Biology Society, 2008. EMBS 2008. 30th Annual International Conference of the IEEE (pp. 5490-5493). IEEE.

[49] Awad, A., Elsayed, M. H., & Mohamed, A. (2016). Encoding Distortion Modeling For DWT-Based Wireless EEG Monitoring System. arXiv preprint arXiv:1602.04974.

[50] Awad, A., Mohamed, A., Chiasserini, C. F., & Elfouly, T. (2017, March). Network Association With Dynamic Pricing Over D2D-Enabled Heterogeneous Networks. In Wireless Communications and Networking Conference (WCNC), 2017 IEEE

[51] Psaras, I., Chai, W. K., & Pavlou, G. (2014). In-network cache management and resource allocation for information-centric networks. IEEE Transactions on Parallel and Distributed Systems, 25(11), 2920-2931.

[52] Awad, A., Mohamed, A., Chiasserini, C. F., & Elfouly, T. (2017). Distributed in-network processing and resource optimization over mobile-health systems. Journal of Network and Computer Applications, 82, 65-76.

[53] Awad, A., Mohamed, A., & El-Sherif, A. A. (2013). Energy efficient cross-layer design for wireless body area monitoring networks in healthcare applications. 2013 IEEE 24th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC). doi:10.1109/pimrc.2013.6666376

[54] Awad, A., Hussein, R., Mohamed, A., & El-Sherif, A. A. (2013). Energy-aware cross-layer optimization for EEG-based wireless monitoring applications. 38th Annual IEEE Conference on Local Computer Networks. doi:10.1109/lcn.2013.6761267

[55] Awad, A., Mohamed, A., El-Sherif, A. A., & Nasr, O. A. (2014). Interference-aware energy-efficient cross-layer design for healthcare monitoring applications. Computer Networks, 74, 64-77. doi:10.1016/j.comnet.2014.09.003

[56] Awad, A., Hamdy, M., & Mohamed, A. (2014). Transmission Delay Minimization for Energy Constrained Communication in Wireless Body Area Sensor Networks. 2014 6th International Conference on New Technologies, Mobility and Security (NTMS). doi:10.1109/ntms.2014.6814056

[57] How Deep Learning Will Change Our World. (n.d.). Retrieved from http://www.circuitinsight.com/programs/54708.html

[58] Ravì, D., Wong, C., Deligianni, F., Berthelot, M., Andreu-Perez, J., Lo, B., & Yang, G. Z. (2017). Deep learning for health informatics. IEEE Journal of biomedical and health informatics, 21(1), 4-21.

[59] Gibson, A., & Patterson, J. (n.d.). Deep Learning. Retrieved from https://www.safaribooksonline.com/library/view/deep-learning/9781491924570/ch04.html

[60] Baldi, P. (2012, June). Autoencoders, unsupervised learning, and deep architectures. In Proceedings of ICML Workshop on Unsupervised and Transfer Learning (pp. 37-49).

[61] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. In Advances in neural information processing systems (pp. 1097-1105).

[62] Brownlee, J. (2016). Deep learning with Python: develop deep learning models on Theano and TensorFlow using Keras. Place of publication not identified: Machine Learning Mastery

[63] Discrete wavelet transform. (2017, September 27). Retrieved from https://en.wikipedia.org/wiki/Discrete_wavelet_transform

[64] Nguyen, A. D., Sénac, P., Ramiro, V., & Diaz, M. (2011, May). STEPS-an approach for human mobility modeling. In International Conference on Research in Networking (pp. 254-265). Springer, Berlin, Heidelberg.

[65] Hong, S., Lee, K., & Rhee, I. (2010, November). STEP: A spatio-temporal mobility model for humans walks. In Mobile Adhoc and Sensor Systems (MASS), 2010 IEEE 7th International Conference on (pp. 630-635). IEEE.

[66] K. Srinivasan, J. Dauwels, and M. R. Reddy, "A two-dimensional approach for lossless EEG compression," Biomedical Signal Processing and Control, 2011.

[67] Ioffe, S., & Szegedy, C. (2015, June). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In International Conference on Machine Learning (pp. 448-456).

[68] Leng, B., Guo, S., Zhang, X., & Xiong, Z. (2015). 3D object retrieval with stacked local convolutional autoencoder. Signal Processing, 112, 119-128.

[69] Makhzani, A., & Frey, B. (2014, September). A winner-take-all method for training sparse convolutional autoencoders. In NIPS Deep Learning Workshop.

[70] Holden, D., Saito, J., Komura, T., & Joyce, T. (2015, November). Learning motion manifolds with convolutional autoencoders. In SIGGRAPH Asia 2015 Technical Briefs (p. 18). ACM.

[72] Brunner, C., Leeb, R., Müller-Putz, G., Schlögl, A., & Pfurtscheller, G. (2008). BCI Competition 2008–Graz data set A. Institute for Knowledge Discovery (Laboratory of Brain-Computer Interfaces), Graz University of Technology, 136-142.

[73] Leeb, R., Brunner, C., Müller-Putz, G., Schlögl, A., & Pfurtscheller, G. (2008). BCI Competition 2008–Graz data set B. Graz University of Technology, Austria.

[74] "DEAP: A Database for Emotion Analysis Using Physiological Signals (PDF)", S. Koelstra, C. Muehl, M. Soleymani, J.-S. Lee, A. Yazdani, T. Ebrahimi, T. Pun, A. Nijholt, I. Patras, IEEE Transaction on Affective Computing, Special Issue on Naturalistic Affect Resources for System Building and Evaluation, in press

[75] CVX: Matlab Software for Disciplined Convex Programming. (n.d.). Retrieved from http://cvxr.com/cvx/

APPENDIX A: EXTRA BCI-IV-2A DATASET VISUALIZATION RESULTS

The proposed Convolutional Autoencoder (CAE) on BCI-IV-2a dataset has an excellent performance as discussed in chapter 4. This appendix contains an example of the original and reconstructed signal at different compression ratios to ensure the accuracy and effectiveness of the proposed solution.

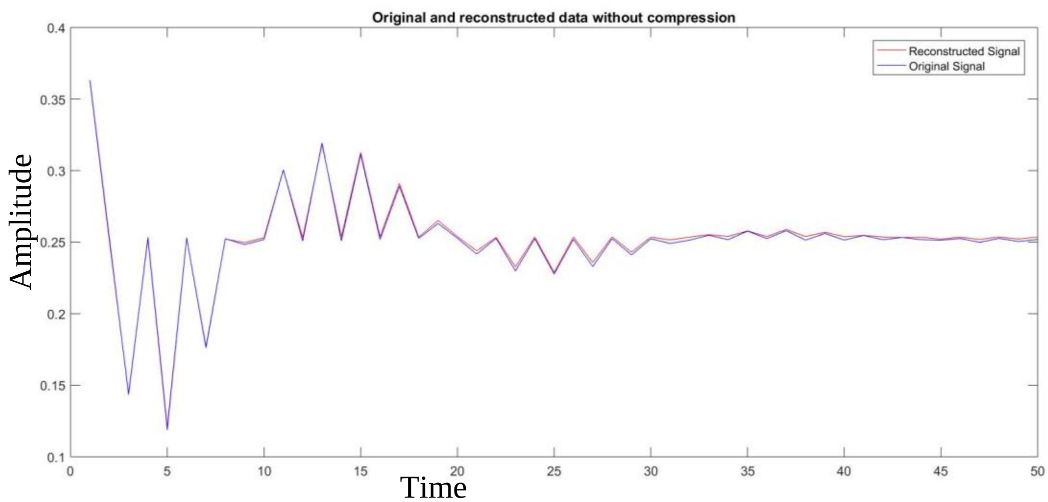

Figure 40: Original and Reconstructed Signal at 0 % Compression

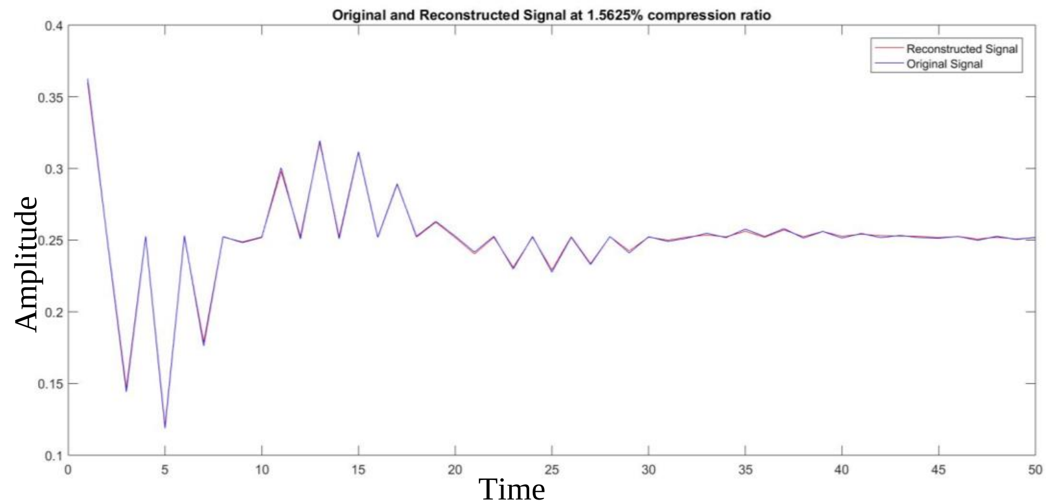

Figure 41: Original and Reconstructed Signal at 1.56 % Compression

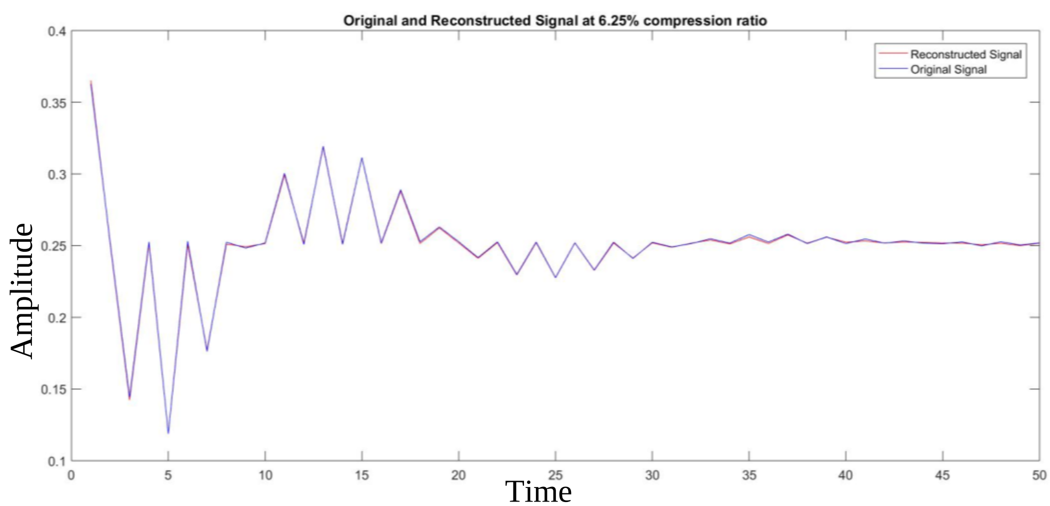

Figure 42: Original and Reconstructed Signal at 6.25 % Compression

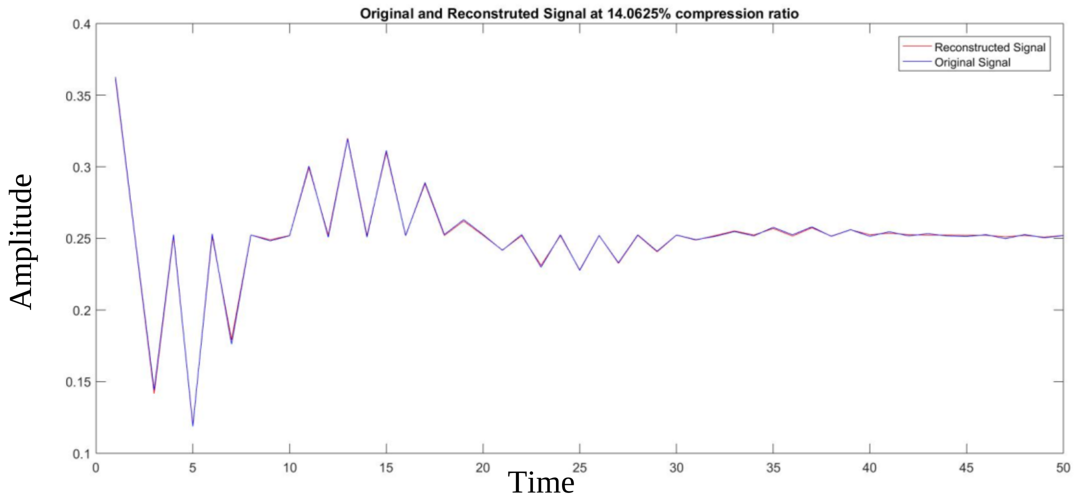

Figure 43: Original and Reconstructed Signal at 14 % Compression

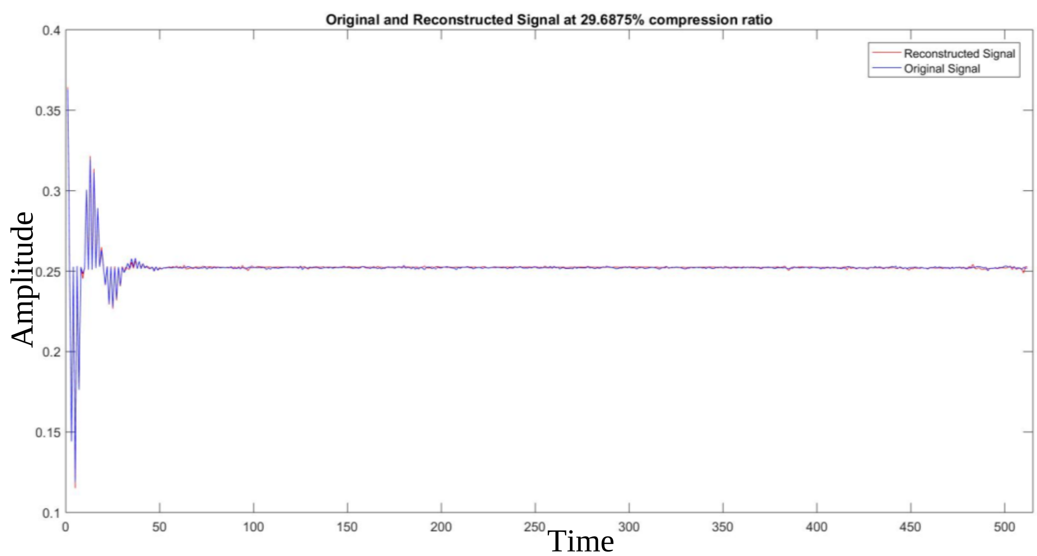

Figure 44: Original and Reconstructed Signal at 30 % Compression

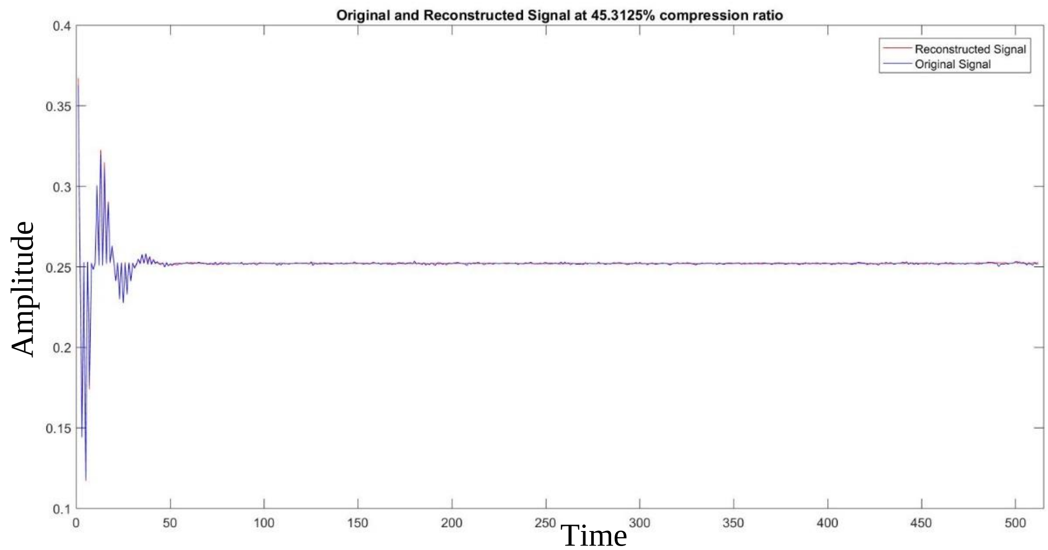

Figure 45: Original and Reconstructed Signal at 45 % Compression

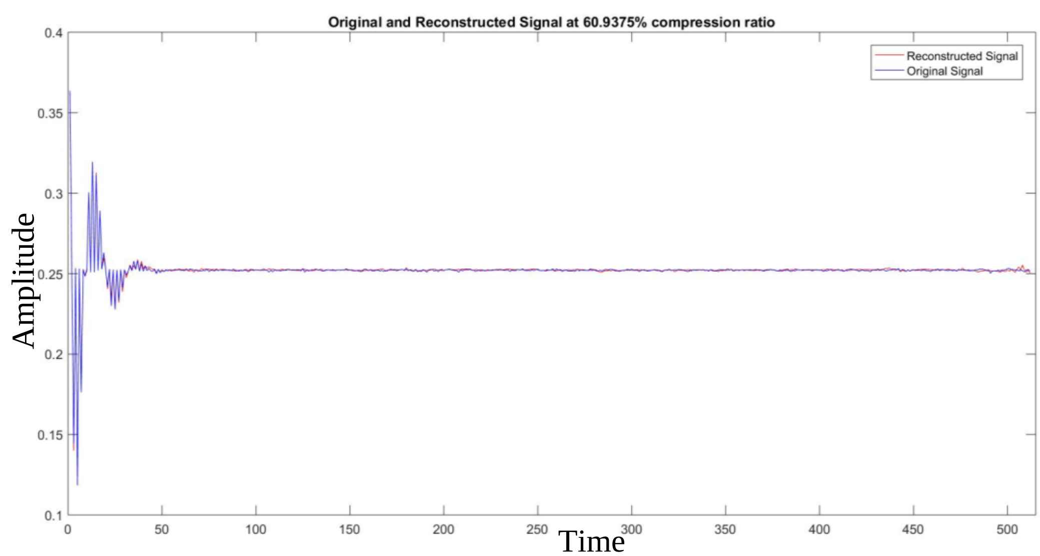

Figure 46: Original and Reconstructed Signal at 61 % Compression

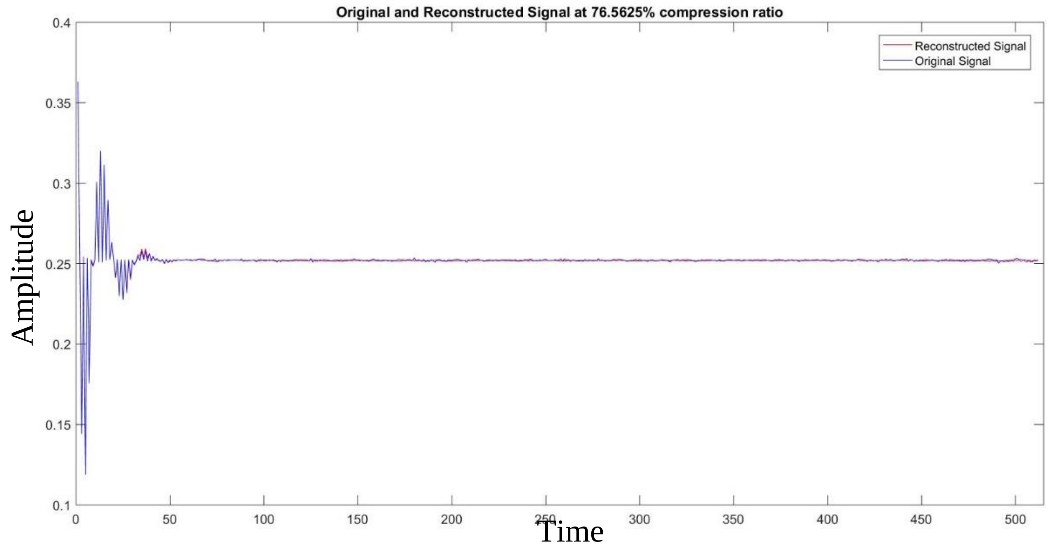

Figure 47: Original and Reconstructed Signal at 77 % Compression

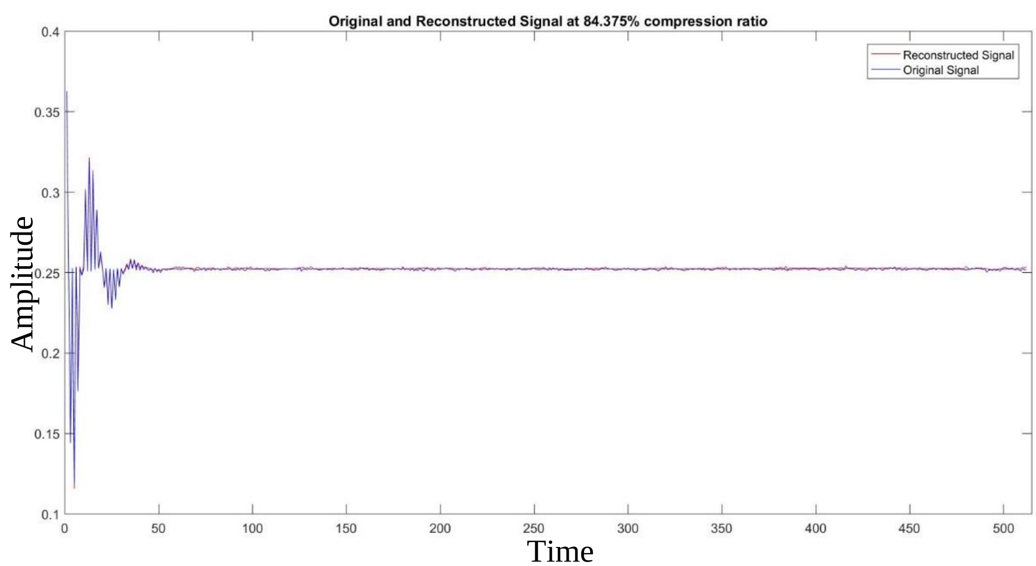

Figure 48: Original and Reconstructed Signal at 85 % Compression

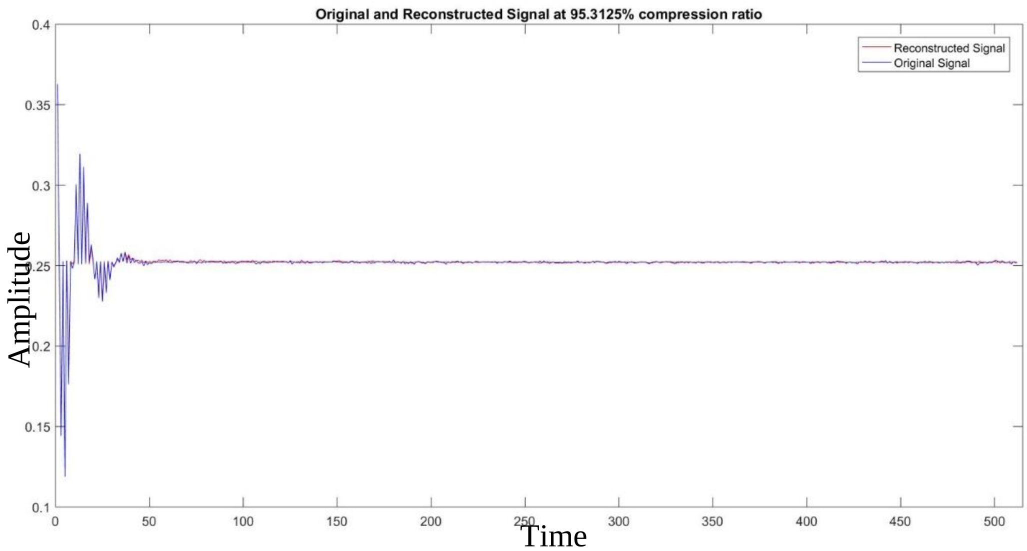

Figure 49: Original and Reconstructed Signal at 95 % Compression

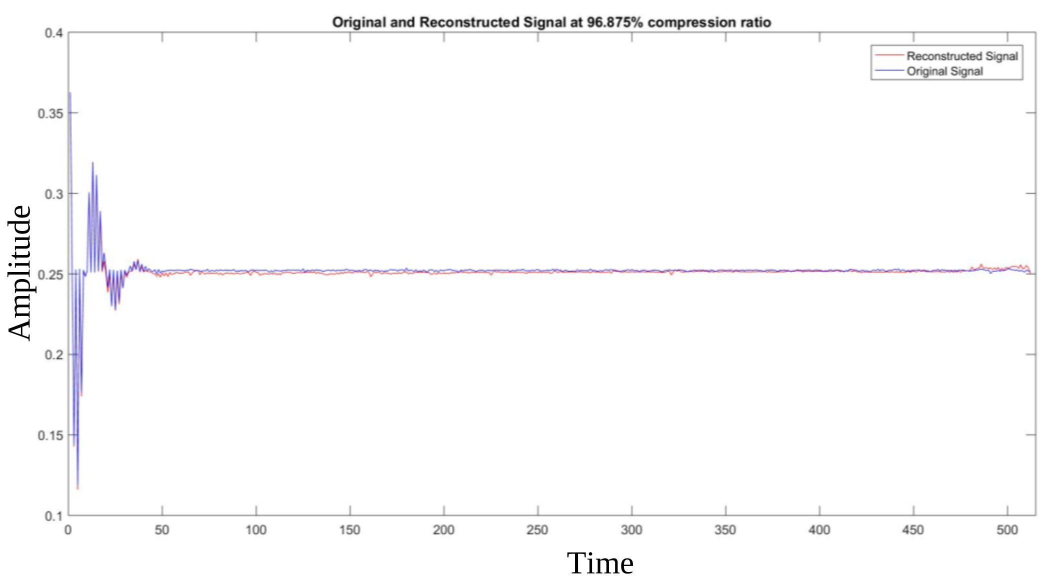

Figure 50: Original and Reconstructed Signal at 97 % Compression