Secure Steganography, Compression and Diagnoses of Electrocardiograms in Wireless Body Sensor Networks

A thesis submitted for the degree of

Doctor of Philosophy

Ayman Ibaida M.Sc,

School of Computer Science and Information Technology,

Science, Engineering, and Technology Portfolio,

RMIT University,

Melbourne, Victoria, Australia.

10th January, 2014

Declaration

I certify that except where due acknowledgement has been made, the work is that of the author alone; the work has not been submitted previously, in whole or in part, to qualify for any other academic award; the content of the thesis is the result of work which has been carried out since the official commencement date of the approved research program; and, any editorial work, paid or unpaid, carried out by a third party is acknowledged.

Ayman Ibaida

School of Computer Science and Information Technology

RMIT University

10th January, 2014

Acknowledgments

In offering my substantial gratitude to my supervisor, Ibrahim Khalil, I would like to thank him for his valuable guidance and advice during my Ph.D. journey. Without his guidance and motivation I couldnt have accomplished this. I would also like to thank my second supervisor, Xiaodong Li, for his advice and support. I wish also to thank the staff at the School of Computer Science for their support and the good research environment provided by the school.

I am very thankful to my colleague, Fahim Sufi, for his valuable help and guidance when I commenced my Ph.D.: he made my start smooth without any problems. Moreover, I would like to express my great gratitude to my close friend, Dhiah Al-Shammary, who helped me not only in my academic studies but also in my personal life. A real friend is more valuable than a treasure.

A special thanks to my father and my mother, they sacrificed a lot for me; without them I could not be at this level in my life. Their prayers always helped me a lot. I also wish to thank my brother and sister for their support during my PhD studies. Finally, I express my great love to my wife Manal for her patience and support for me. I used to take my strength to continue my Ph.D. from her kind words that encouraged me to reach this position. My family: without you, I could not have achieved what I have.

Credits

Portions of the material in this thesis have previously appeared in the following publications:

- Ibaida, A.; Khalil, I., "Wavelet-Based ECG Steganography for Protecting Patient Confidential Information in Point-of-Care Systems," *IEEE Transactions on Biomedical Engineering*, Vol.60, No.12, Pp.3322-3330, Dec. 2013. **ERA A* and Featured Article**

- Ayman Ibaida, Dhiah Al-Shammary, Ibrahim Khalil, Cloud enabled fractal based ECG compression in wireless body sensor networks, *Future Generation Computer Systems*, Available online 19 December 2013, ISSN 0167-739X. **ERA A**

- Vu Mai; Khalil, I.; Ibaida, A., "Steganography-based access control to medical data hidden in electrocardiogram,"),*2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society EMBC* , Pp.1302-1305, 3-7 July 2013

- Ibaida, A.; Khalil, I.; Sufi, F.; Cardiac abnormalities detection from compressed ECG in wireless telemonitoring using principal components analysis (PCA). *IEEE 5th International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*, 2009

- Ibaida, A.; Khalil, I.; Al-Shammary, D.; Embedding patients confidential data in ECG signal for healthcare information systems. *Annual International Conference of the IEEE engineering in Medicine and Biology Society (EMBC)*, 2010

- Ibaida, A.; Khalil, I.; van Schyndel, R.; A low complexity high capacity ECG signal watermark for wearable sensor-net health monitoring system . *IEEE Computing in*

Cardiology, 2011.

- Ibaida, A.; Khalil, I.; Distinguishing between ventricular Tachycardia and Ventricular Fibrillation from compressed ECG signal in wireless Body Sensor Networks. *Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2010.

Contents

List of Figures

List of Tables

Abstract

The usage of e-health applications is increasing in the modern era. Remote cardiac patients monitoring application is an important example of these e-health applications. Diagnosing cardiac disease in time is of crucial importance to save many patients lives. More than 3.5 million Australians suffer from long-term cardiac diseases. Therefore, in an ideal situation, a continuous cardiac monitoring system should be provided for this large number of patients. However, health-care providers lack the technology required to achieve this objective. Cloud services can be utilized to fill the technology gap for health-care providers. However, three main problems prevent health-care providers from using cloud services. Privacy, performance and accuracy of diagnoses. In this thesis we are addressing these three problems. To provide strong privacy protection services, two steganography techniques are proposed. Both techniques could achieve promising results in terms of security and distortion measurement. The differences between original and resultant watermarked ECG signals were less then 1%. Accordingly, the resultant ECG signal can be still used for diagnoses purposes, and only authorized persons who have the required security information, can extract the hidden secret data in the ECG signal. Consequently, to solve the performance problem of storing huge

amount of data concerning ECG into the cloud, two types of compression techniques are introduced: Fractal based lossy compression technique and Gaussian based lossless compression technique. This thesis proves that, fractal models can be efficiently used in ECG lossy compression. Moreover, the proposed fractal technique is a multi-processing ready technique that is suitable to be implemented inside a cloud to make use of its multi processing capability. A high compression ratio could be achieved with low distortion effects. The Gaussian lossless compression technique is proposed to provide a high compression ratio. Moreover, because the compressed files are stored in the cloud, its services should be able to provide automatic diagnosis capability. Therefore, cloud services should be able to diagnose compressed ECG files without undergoing a decompression stage to reduce additional processing overhead. Accordingly, the proposed Gaussian compression provides the ability to diagnose the resultant compressed file. Subsequently, to make use of this homomorphic feature of the proposed Gaussian compression algorithm, in this thesis we have introduced a new diagnoses technique that can be used to detect life-threatening cardiac diseases such as Ventricular Tachycardia and Ventricular Fibrillation. The proposed technique is applied directly to the compressed ECG files without going through the decompression stage. The proposed technique could achieve high accuracy results near to 100% for detecting Ventricular Arrhythmia and 96% for detecting Left Bundle Branch Block. Finally, we believe that in this thesis, the first steps towards encouraging health-care providers to use cloud services have been taken. However, this journey is still long.

Chapter 1

Introduction

Cardiovascular diseases (CVD) are the Number One killer of the modern era [66]. Wireless cardiovascular monitoring facilities are widely used to provide continuous patient monitoring and to send urgent alerts to the specialists in case of any emergency or abnormal life-threatening cardiac behaviour. Accordingly, Electrocardiogram (ECG) signals are widely used in these monitoring systems. ECGs are electrical signals that reflect the heart electrical functionalities over time and they are collected using electrodes. Consequently, the development of portable wireless monitoring facilities such as body sensors, has increased significantly with the aim of utilizing ECG signals to diagnose most cardiac diseases.

The use of E-health applications is increasing to a great extent around the globe. Many health-care organizations such as insurance companies, hospitals, and government health sectors require access to patient information and records including their archived biomedical signals. Therefore, patient records need to be stored in a centralized repository which will allow other health-care organizations to access these records. Cloud can facilitate this service

[53; 8; 24; 23; 70].

In typical wireless tele-monitoring systems with body sensor networks, patients wear one or more body sensors to collect their ECG signals. Next, the collected biomedical signals are transmitted to the patient's smart-phone where any processing required is implemented. Finally, the collected signals as well as other patient information are transmitted to the medical cloud using the Internet [52]. Alternatively, patients at hospital can also send their biomedical signals with their information to the centralized medical cloud. However, in this scenario, many challenges arise.

1. Patient confidential information should be transmitted to the medical cloud along with patient physiological signals. In this case, confidential information requires a special mechanism for protection against intruders [66; 38; 67].

2. ECG signals are of enormous sizes. A typical electrocardiogram monitoring device generates massive volumes of digital data. Depending on the application for the data, the sampling rate varies from 125 to 1024 Hz. Each data sample may be represented using 8 to 16 bit binary number. Up to 12 different streams of data may be obtained from various sensors placed on the patient's body. Even if we presume the application will need the lower sampling rate and only one sensor will be needed that generates 8-bit data, we would accumulate ECG data at a rate of 7.5 KB per minute or 450 KB per hour. At the other extreme, (12 sensors generating 16-bit values at 1024 Hz), data is generated at the rate of 1500 KB per minute or more than 88 MB per hour [27]. If this large amount of data is transmitted wirelessly to the medical cloud then it will require a large bandwidth as well as very considerable power.

3. The body sensors and smart devices generally have limited power as well as limited processing capabilities. At the same time, they have few storage resources. As a result, complicated algorithms for ECG diagnoses cannot be implemented within these sensor nodes. Therefore, ECG diagnoses should be performed inside the medical cloud. However, if the ECG signals stored in the cloud are in a compressed format, and most of the proposed ECG diagnoses algorithms involve automatic diagnosis of cardiac diseases from ECG signal using the raw ECG signal, then the cloud must decompress the ECG signal, apply the diagnoses process and then compress the ECG again. In this case, a huge processing overhead is created.

The aim of this research, is to addresses these three major problems of the current wireless tele-monitoring systems that use a centralized medical cloud. New algorithms are introduced as cloud-enabled algorithms which are suitable to be implemented in the cloud. On the other hand, further algorithms are proposed to be implemented in the patient's smart device or the hospital server. Finally, the proposed algorithms are evaluated based on the improvements they provide in the system performance. The effect of applying these algorithms on the resultant ECG signal quality is also thoroughly investigated.

1.1 Scope and Goals

Improving the performance of wireless cardiovascular monitoring systems is of significant importance. In this research, we investigated and proposed new techniques that will add new knowledge to the theoretical and practical issues of wireless body sensor networks. The aim of this research is to find fast and reliable techniques for cardiac abnormalities diagnoses, combined with providing a reliable and secure method to transfer the ECG signal, as well as patient-sensitive medical information, to the responsible doctors in any emergency case. The outcome of the research will improve the performance and security of health-care systems and will save patients lives in many cases. Therefore, this research offers crucial benefits to the community.

Most research that has been done on automated diagnoses of cardiac diseases were based on raw ECG signal, where the original raw ECG signal is processed [37; 63; 91]. Part of this thesis, introduces a new research topic to the current ECG analyses area, providing a unique diagnosis for compressed-ECG signals using data mining approaches. This technique has never been used for ECG diagnoses research as a technique to solve the major problems in the Cardiovascular monitoring system. In fact, the success of this research will open many research opportunities and studies in the fields of direct compression data analysis. Accordingly, a new research area of analysing encrypted data without any decryption will be opened. Based on this research, others are invited to develop this area to make the whole communication faster. This research also investigates the topic of watermarking and steganography and proposes a new steganography model to hide sensitive patient information inside an ECG signal without increasing its size. Accordingly, this research will open another new interesting subject for researchers to use ECG signal as host rather then images to host secret data.

Finally, this research proposes a compression technique that utilizes fractal model which is used mostly in image processing and compression, to be used for ECG compression. This will lead and encourage other researchers to investigate more in this area as a new area in the ECG compression world.

The improvement in wireless cardiovascular monitoring systems will produce efficient health-care systems which will have increased productivity and usage. As a result, industries, manufacturers and health-care systems companies will play a crucial role in the future health-care revolution. In addition, since the ultimate aims of this work is saving patient lives, the massive potential benefits in realizing this work could, perhaps, encourage government support and increased revenues in related commercial companies. The results of this research will also motivate industrial companies to investigate new security techniques of ECG steganography. Further, this work will encourage those industries to scrutinize the new way of directly analysing the compressed data using the current available compression techniques. This research will also boost motivation in communication industries to further improve the proposed ECG signal compression technique to make it compatible with analogue signals in the communication area.

1.2 Research Questions

After explaining possible problems in the current remote patient health-care applications, in this research we seek to answer the following research questions.

- **How patients' confidential information can be protected and transmitted securely in wireless tele-cardiovascular monitoring systems?**

  Sending sensitive patient information through the Internet is a challenging problem. Researchers have proposed various techniques to protect patient health records while it is being transmitted on the communication channel or while being stored in the remote cloud. However, most previously-proposed techniques rely on the theory of cryptography and data ciphering. Therefore, they inherit the problems of the current data cyphering techniques, such as large data overhead and high complexity, especially with large key sizes. In this research, a new ECG-based steganography technique that enables sending patient's confidential information is proposed. This research will study the ability of hiding patient's sensitive information inside the sent/received ECG signal. It will also study the effect of hiding these data on the resultant ECG signal and whether the resultant ECG signal can be used for diagnostic purposes or not.

- **How can ECG signal self-similarity characteristics be utilized to relize a powerful lossy compression algorithm that can be implemented in either the client side or the cloud side?**

  This question addresses lossy compression techniques which can provide high compression ratio with low information loss. What will be the impact of using the proposed method on the resultant ECG signal? How can we design an algorithm which can be implemented inside the cloud to use its parallel processing capabilities without adding to the overhead. Moreover, can the same algorithm be modified and implemented inside the client device rather than the cloud?

- **How to utilize ECG morphological characteristics to implement a lossless compression algorithm that can be used for compressing ECG signal while keeping the resultant compressed file usable for diagnoses without decompression?**

In this research question, a new lossless compression technique is proposed that can provide a high compression ratio with zero data loss. Moreover, the characteristics of the compressed file should allow the diagnoses process to be applied directly to the compressed file without decompression. Therefore, the compressed file should preserve the important features of the ECG signal as well as the sequence of the data.

- **How can cardiac diseases be diagnosed directly from compressed ECG using the proposed lossless compression algorithm?**

  In this research we will focus especially on tachyarrhythmia diseases such as ventricular tachycardia and ventricular fibrillation. Another cardiac disease is also investigated; Left Bundle Branch Block (LBBB). We intend to focus on ventricular diseases because they are the most life-threatening [55]. Using the proposed compression technique, what will be the accuracy of detecting ventricular abnormality? What will be the accuracy of detecting LBBB? What will be the suitable data mining technique to achieve this goal?

1.3 Contributions

In answering the research questions introduced in Section 1.2, this thesis provides several contributions in the area of health-care and particularly remote cardiac monitoring systems. The thesis contributes in three major areas: security, communication performance and cardiovascular diagnoses.

- **Proposing new ECG steganography techniques for protecting patient confidential information**

In this research, two steganography techniques are introduced. Firstly, time domain ECG steganography is implemented. It is based on special range transform which provides a simple and secure method to hide information in different bit positions. However, the amount of data that can be hidden is small. To hide larger amounts of data, a frequency domain wavelet based steganography technique has been introduced. This combines encryption and a scrambling technique to protect patient confidential data. The proposed method allows the ECG signal to hide its corresponding patient confidential data and other physiological information thus guaranteeing the integration between ECG and the rest. To evaluate the effectiveness of the proposed techniques on the ECG signal, two distortion measurement metrics have been used: the Percentage Residual Difference (PRD) and the Wavelet Weighted PRD (WWPRD). Both the proposed techniques provide high security protection for patients' data with low distortion and ECG data remains diagnosable after watermarking (i.e. hiding patient confidential data) as well as when watermarks (i.e. hidden data) are removed from the watermarked ECG. The proposed techniques will provide security without adding any overhead to the transmitted ECG signal.

- **Proposing a new cloud enabled fractal based ECG lossy compression**

  In this research, a new fractal-based ECG lossy compression technique is proposed. It is found that the ECG signal self similarity characteristic can be used efficiently to achieve high compression ratios. The proposed technique is based on modifying the popular fractal model to be used in compression in conjunction with Iterated Function System. An ECG signal is divided into equal blocks called range blocks. Subsequently, another down-sampled copy of the ECG signal is created which is called domain. For each range block the most similar block in the domain is found. As a result, fractal coefficients (i.e. parameters defining the fractal compression model) are calculated and stored in the compressed file for each ECG signal range block. In order to make our technique cloud- friendly, the decompression operation is designed in such a way that allows the user to retrieve part of the file (i.e ECG segment) without decompressing the whole file. Therefore, clients do not need to download the full compressed file before they can view the result. The proposed algorithm has been implemented and compared with other existing lossy ECG compression techniques. It has been found that the proposed technique can achieve a higher compression ratio and a lower Percentage Residual Difference (PRD) Value. Furthermore, the proposed technique is designed to allow it to be implemented within a parallel processing environment such as cloud without adding more overheads in the communication between the processing units.

To use fractal compression inside the client device and not the cloud, we modified the current fractal model to increase its performance and proposed another fast fractal ECG lossy compression technique. The effect of using both methods on the ECG signal is calculated using PRD measurement.

- **Proposing new ECG lossless compression using Gaussian-based approximation and Burrow-Wheeler Transform**

  In this research, a new Gaussian-based ECG lossless compression technique is proposed. It is found that the Gaussian function can be used to model and approximate ECG signals. Therefore, optimization theory is used to determine the best Gaussian function parameters for ECG approximation. Next, residuals are calculated and differentiated. Subsequently, the residuals are encoded using the Burrow-Wheeler Transform (BWT), followed by move-to-front (MTF) and run-length encoding. Finally, the resultant encoded signal is further encoded using Huffman coding. BWT and MTF encoding are modified to deal with numbers instead of dealing with strings. The proposed algorithm has been implemented and compared with other existing lossless ECG compression techniques. It has been found that the proposed technique can achieve a higher compression ratio with a guaranteed exact reconstruction of the ECG signal. Processing the signal in a time domain ensures that the features and sequence of the ECG signal points will stay at the same sequence. As a result, the compressed file will be highly correlated with the original ECG signal. This feature will be used later for diagnoses purposes directly from the ECG compressed file.

- **Proposing new cardiac diseases detection algorithms from the compressed ECG signal using Principle Components Analyses and neural networks**

  In this research a new diagnoses technique is introduced to detect ventricular arrhythmia and LBBB directly from compressed ECG signal without applying the decompression stage. The proposed algorithm uses PCA technique for feature extraction and k-mean for clustering of normal and abnormal ECG signals. The k-mean algorithm is replaced with a Perceptron-Neural Network to improve the accuracy of the system. The diagnosis process is implemented in the cloud to enable it to diagnose the compressed ECG files already stored in the cloud.

1.4 Thesis Structure

In this introductory chapter we discussed the current problems in cardiac remote monitoring systems. Moreover, we explained the research questions and the thesis contributions. The rest of this thesis is organized as follows:

1. Chapter 2 introduces the concept of steganography and explains two new proposed ECG steganography techniques. This chapter then evaluates and shows the effect of using the proposed steganography techniques on the resultant ECG quality. Security analysis is then introduced and discussed for the proposed techniques.

2. Chapter 3 proposes a new lossy compression technique using a fractal model. It also explains how the proposed technique can be utilized within a parallel processing environment such as cloud. Finally, a new mathematical model is explained to improve the performance of the current fractal compression model.

3. Chapter 4 analyses the current available lossless compression techniques. It introduces a new Gaussian-based lossless technique and compares it with the other lossless technique.

4. Chapter 5 introduces a new diagnostic technique directly from the compressed file introduced in Chapter 4, Also in this chapter, ventricular abnormalities will be diagnosed as well as the Left Bound Branch Block.

5. Chapter 6 summarizes the thesis contributions proposed in this research and discusses future improvements and research areas that could benefit from this thesis.

Chapter 2

ECG Steganography to Protect Patient Confidential Information

This chapter answers the first research question raised in Section 1.2. The security problem of transmitting patient confidential information is discussed with examples of current available solutions and their limitations. This chapter also shows how the steganography theory combined with encryption can be a feasible and reliable solution for this security problem. Section 2.1 discusses the problem statement and its negative implication on the current e-health system. Section 2.2 briefly discusses the related works and what other researchers have done to solve this problem. Section 2.3 explains the first proposed time domain steganography technique. In Section 2.4 the second wavelet based steganography technique is introduced. In this section we discuss the basic system design, the embedding process (i.e patient sensitive data into ECG signal), the extraction process and security analysis. Then Section 2.5 explains diagnosability measurement. Section 2.6.2 shows the results of PRD calculated before and after secret data extraction for both the proposed algorithms. Finally, Section 2.7 summarizes this chapter.

2.1 Introduction

The number of elderly patients is increasing dramatically due to recent medical advances. Accordingly, to reduce medical labor costs, the use of remote healthcare monitoring systems and Point-of-Care (PoC) technologies have become popular [51; 28]. Monitoring patients at home can drastically reduce the increasing traffic at hospitals and medical centres. Moreover, Point-of-Care solutions can provide greater reliability in emergency services as patient medical information (ex. for diagnosis) can be sent immediately to doctors and response or appropriate action can be taken without delay. However, Remote health care systems are used in large geographical areas essentially for monitoring purposes, and, the Internet represents the main communication channel used to exchange information. Typically, patient biological signals and other physiological readings are collected using body sensors. Next, the collected signals are sent to the patient PDA device for further processing or diagnosis. Finally, the signals and patient confidential information as well as diagnosis report or any urgent alerts are sent to the central hospital servers or medical cloud via the Internet. Doctors can check those biomedical signals and possibly make a decision in the case of an emergency from anywhere using any device[33].

Using Internet as a main communication channel introduces new security and privacy threats as well as data integration issues. According to the Health Insurance Portability and Accountability Act (HIPAA), information sent through the Internet should be protected and secured. HIPAA mandates that while transmitting information through the Internet a patient's privacy and confidentiality be protected as follows: [48]:

1. Patient privacy: It is of crucial importance that a patient can control who will use his/her confidential health information, such as name, address, telephone number, and Medicare number. As a result, the security protocol should provide further control on who can access patient's data and who cannot.

2. Security: The methods of computer software should guarantee the security of the information within the communication channels as well as the information stored on the hospital server or on the cloud.

Accordingly, it is of crucial importance to implement a security protocol which will have powerful communication and storage security [59].

Several researchers have proposed various security protocols to secure patient confidential information. Techniques used can be categorized into two subcategories. Firstly, there are techniques that are based on encryption and cryptographic algorithms. These techniques are used to secure data during the communication and storage. As a result, the final data will be stored in encrypted format [48; 57; 28; 82]. The disadvantage of using encryption based techniques is its large computational overhead. Therefore, encryption based methods are not suitable in resource-constrained mobile environment. Alternatively, many security techniques are based on hiding its sensitive information inside another insensitive host data without incurring any increase in the host data size and huge computational overhead. These techniques are called steganography techniques. Steganography is the art of hiding secret information inside another type of data called host data [61]. However, steganography techniques alone will not solve the authentication problem and cannot give the patients the required ability to control who can access their personal information as stated by HIPAA.

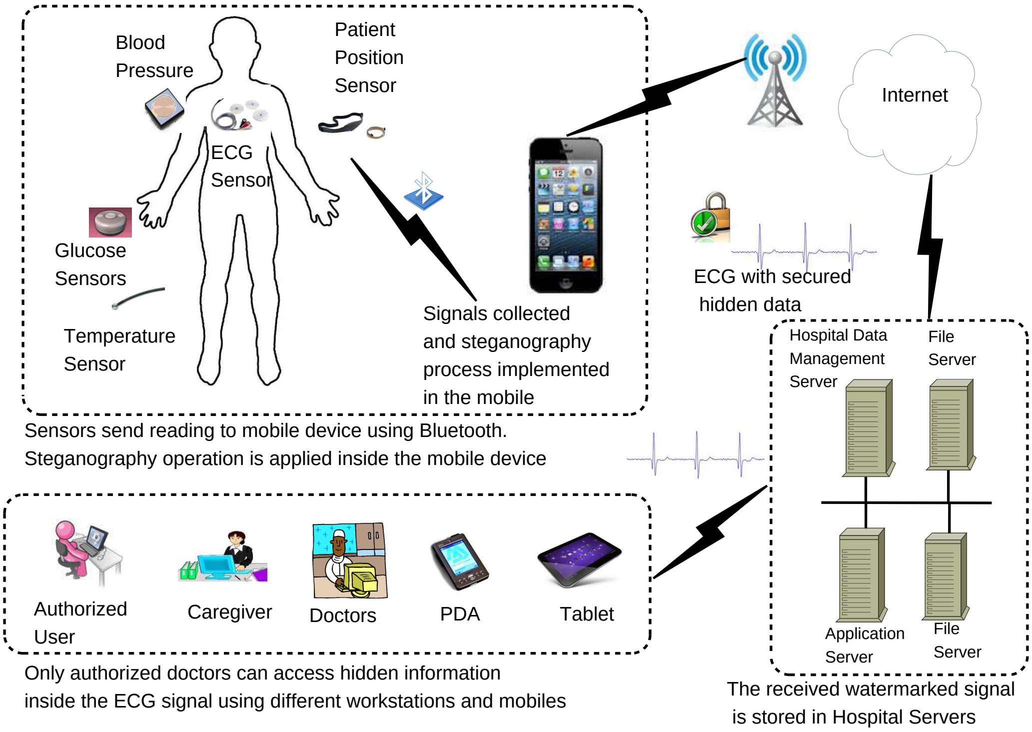


Figure 2.1: ECG steganography scenario in Point-of-Care (PoC) systems where body sensors collect different readings as well as ECG signal and watermarking process implemented inside the patient's mobile device

To apply steganography technique using ECG signal as a host, there are two approaches to achieve this goal. Firstly, perform all the steganography stages in time domain which will result in better performance. However, the amount of information that can be hidden using this approach will be small. Secondly, performing the steganography technique on frequency domain which will result in lower performance but the amount of information that can be hidden will be larger. Therefore, in this chapter, two new security techniques are proposed to guarantee secure transmission of patient confidential information combined with patient physiological readings from body sensors. The first proposed technique is based on time domain to provide better performance. This technique is based on transforming the ECG signal to a special time domain called (Shift special range transform) to increase the algorithm security. Then it uses LSB embedding to hide the secret data in the transformed special domain ECG. Finally, it returns the resultant ECG signal to its original range. The second proposed technique is based on frequency domain and is a hybrid between cryptography technique and steganography techniques. Firstly, it is based on using steganography techniques to hide patient confidential information inside patient biomedical signal. Moreover, the proposed technique uses encryption based model to allow only the authorized persons to extract the hidden data. In our proposed technique, the patient ECG signal is used as the host signal that will carry the patient secret information as well as other readings from other sensors such as temperature, glucose, position, and blood pressure. The ECG signal is used here due to the fact that most of the health-care systems will collect ECG information. Moreover, the size of the ECG signal is large compared to the size of other information. Therefore, it will be suitable to be a host for other small size secret information. As a result, both the proposed techniques will follow HIPAA guidelines in providing open access for patients biomedical signal but prevents unauthorized access to patient confidential information.

In this scenario body sensor nodes will be used to collect ECG signal, glucose reading, temperature, position and blood pressure, the sensors will send their readings to patient's PDA device via Bluetooth. Then , inside the patient's PDA device the steganography technique will be applied and patient secret information and physiological readings will be embedded inside the ECG host signal. Finally, the watermarked ECG signal is sent to the hospital server or cloud via the Internet. As a result, the real size of the transmitted data is the size of the ECG signal only without adding any overhead, because the other information are hidden inside the ECG signal without increasing its size. At hospital server or cloud, the ECG signal and its hidden information will be stored. Any doctor can see the watermarked ECG signal and only authorized doctors and certain administrative personnel can extract the secret information and have access to the confidential patient information as well as other readings stored in the host ECG signal. This system is shown in Fig. 2.1.

Both the proposed steganography techniques have been designed in such a way that guarantees minimum acceptable distortion in the ECG signal, Furthermore, it will provide the highest security that can be achieved. The use of these techniques will slightly affect the quality of ECG signal. However, watermarked ECG signal can still be used for diagnoses purposes as it is proven in this chapter. In this work the following research questions are answered:

- Can the proposed techniques protect patient confidential data as explained in the HIPAA security and privacy guidelines?
- What will be the effect on the original ECG signal after applying the proposed steganography techniques in terms of size and quality?

2.2 Related Work

There are many approaches to secure patient sensitive data [17; 82; 28; 49]. However, one approach [40; 26; 92] proposed to secure data is based on using steganography techniques to hide secret information inside medical images. The challenging factors of these techniques are how much information can be stored, and to what extent the method is secure. Finally, what will be the resultant distortion on the original medical image or signal.

Kai-mei Zheng and Xu Qian [92] proposed a new reversible data hiding technique based on wavelet transform. Their method is based on applying B-spline wavelet transform on the original ECG signal to detect QRS complex. After detecting R waves, Haar lifting wavelet transform is applied again on the original ECG signal. Next, the non QRS high frequency wavelet coefficients are selected by comparing and applying index subscript mapping. Then, the selected coefficients are shifted one bit to the left and the watermark is embedded. Finally, the ECG signal is reconstructed by applying reverse haar lifting wavelet transform. Moreover, before they embed the watermark, Arnold transform is applied for watermark scrambling. This method has low capacity since it is shifting one bit. As a result only one bit can be stored for each ECG sample value. Furthermore, the security in this algorithm relies on the algorithm itself, it does not use a user defined key. Finally, this algorithm is based on normal ECG signal in which QRS complex can be detected. However, for abnormal signal in which QRS complex cannot be detected, the algorithm will not perform well.

H. Golpira and H. Danyali [26] proposed a reversible blind watermarking for medical images based on wavelet histogram shifting. In this work medical images such as MRI is used as host signal. A two dimensional wavelet transform is applied to the image. Then, the histogram of the high frequency sub-bands is determined. Next, two thresholds are selected, the first is in the beginning and the other is in the last portion of the histogram. For each threshold a zero point is created by shifting the left histogram part of the first threshold to the left, and shifting the right histogram part of the second threshold to the right. The locations of the thresholds and the zero points are used for inserting the binary watermark data. This algorithm performs well for MRI images but not for ECG host signals. Moreover, the capacity of this algorithms is low and no encryption key is involved in its watermarking process.

Finally, S.Kauf and O.Farooq [40] proposed new digital watermarking of ECG data for secure wireless communication. In their work, each ECG sample is quantized using 10 bits, and is divided into segments. The segment size is equal to the chirp signal that they use. Therefore, for each ECG segment a modulated chirp signal is added. Patient ID is used in the modulation process of the chirp signal. Next, the modulated chirp signal is multiplied by a window dependent factor, and then added to the ECG signal. The resulting watermarked signal is 11 bits per sample. The final signal consists of 16 bits per sample, with 11 bits for watermarked ECG and 5 bits for the factor and patient ID. However, in this algorithm the size of ECG signal is increased from 12 bits/sample to 16 bits/sample. This behavior overrides completely the concept of using steganography and the main purpose of steganography that does not increase the original size of the host signal.

2.3 Time Domain Special Range ECG Steganography

As mentioned previously, in recent e-health systems the usage of ECG signal has increased significantly to provide highly qualified remote medical services. In this section, a new steganography technique is proposed that is able to hide the secret message in any position in the host signal without distorting the original signal. This technique provides high security for the secret message by selecting more secured positions (such as MSB) in the host ECG signal that are unexpected to the intruders. The proposed model consists of four sequential steps as shown in Fig. 2.2.

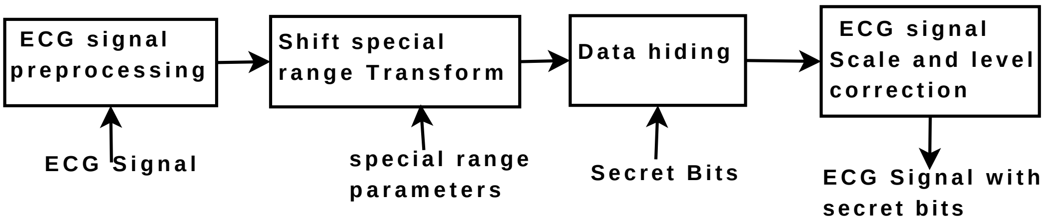


Figure 2.2: Block Diagram for the Proposed ECG Steganography System

2.3.1 ECG Signal Preprocessing

The *first step* is responsible for shifting up and scaling the ECG signal to avoid the negative values and converting the signal floating point numbers into integers. At the same time, the function of this step represents the fist level of security of the steganography technique by hiding the values of both shifting and scaling factors that are mandatory parameters for extracting the secret information. Equation 2.1 is required for shifting up the ECG signal samples.

$$\bar{X} = s + X \tag{2.1}$$

Where $\bar{X}$ is the shifted ECG signal, s is the shift value and X is the original ECG signal. Then, the resultant shifted signal is scaled to compute the integer version of the ECG signal. Equation 2.2 is required to perform the scaling function.

$$\widehat{X} = p * \bar{X} \tag{2.2}$$

Where $\widehat{X}$ is the scaled ECG vector, p is the scale factor and $\bar{X}$ is the shifted ECG vector. Technically , the value of p is based on the ECG sample precisions of the ECG acquisition device and the ADC converter (analogue to digital).

2.3.2 Shift Special Range Transform

In our proposed system, a number of special values in the ECG signal samples are found to be relevant hosts that can hide the secret bits in the most significant positions with the condition of inverting the values of the right hand bits to the secret bit position. At the same time, these special host values do not produce large errors as a result of the hiding operation. Let $M = 128$ be a special value that allows us to hide the secret bit B in the 8^{th} position of binary value of M . By computing the binary representation of M, we get $(10000000)_2$. If $B = 1$, then the special value of M will not be changed as the 8^{th} bit of M is already equal to 1. On the other hand, if $B = 0$, conversion of the 8^{th} bit of M to zero will cause a dramatic change to the host value as the new value of M will be equal to 0 $(00000000)_2$. However, by applying the condition of inverting all the right hand side bits of the secret position (in this case to the right of 8^{th} bit), the new value of M will be 127 which is equivalent to $(01111111)_2$. As a result, this process will reduce the resultant error to 1. For our ECG host signal, we use 32 bits to form each ECG sample. In this binary format, there are many special ranges that are relevant to hide the secret bits in all host sample positions. Equation 2.3 calculates the total number of special ranges that can be used.

Table 2.1: Special ranges for different positions

R_{min}	R_{max}	n	**Position**
127	129	3	8
32767	32769	3	16
8388607	8388607	3	24
2147483647	2147483649	3	32

$$T = \sum_{i=2}^{i=r} 2^{r-i} \tag{2.3}$$

Where T is the total number of special ranges, i is the position of the secret bit, and r is the total number of bits per sample. By applying Eq. 2.3, and setting $r = 32$, the resultant T is 2.1475×10^9.

With the aim of utilizing all the ECG signal samples as host for the secret bits, in the *second step* of the proposed technique (Shift range transform) a new shifting transform is proposed to shift any number in the host signal to any required special range number. Equation 2.4 is required to perform the shifting operation to the host signal sample.

$$S = R_{min} + (M \mod n) \tag{2.4}$$

Where S is the new shifted value, R_{min} is the starting value of the target special range, M is the value of the host signal sample and n is the length of the special range. Table 2.1 shows some examples of special ranges that we used to hide bits in positions 8,16,24 and 32.

```
011001110110100111
101111101010000110
100011000111011000
101010111110001010
```

Figure 2.3: The Secret Binary Data

2.3.3 Data Hiding

The *third step* of the proposed technique is the actual data hiding process. The basic idea of this process is to hide the secret bits using the shifted value as a host, then the resultant value would be shifted back to its original level. Equation 2.5 is required to perform the hiding processes.

$$M_n = \begin{cases} M_o + (R_{max} - S) & \text{if } B\text{=}1 \\ \\ M_o - (S - R_{min}) & \text{if } B\text{=}0 \end{cases} \tag{2.5}$$

Where M_n is the new resultant value of the host data, M_o is the original host signal sample, R_{min} is the minimum value of the selected special range, R_{max} is the maximum value of the selected special range and B is the secret bit. Figure 2.3 shows a block of the secret bits that were inserted into the host signal samples of ECGs.

2.3.4 ECG Signal Scaling and Level Correction

The *final step* of the proposed technique is to de-scale the signal and shift it back to its original values. To extract the hidden data from the host signal the receiver needs to know two parameters, the used range and signal pre-processing parameters. The receiver should apply the same transformation using Eq. 2.1 in addition to performing bitwise operation to extract the secret bits located in special range values of ECG host samples. Consequently, only authorized persons who know this information can extract the hidden data. The strength of this approach is based on the very large possibilities of the available special ranges that is up to 2.1475×10^9 and pre-processing parameters such as 2^{32} possible shifting values and few possibilities of scaling values 200 for MIT-BIH data. Therefore, even if we ignore scaling factor, the intruder needs to try this very large number of possibilities (i.e. $2.1475 \times 10^9 \times 2^{32}$) to enable him to extract the secret message from host ECG samples.

2.4 Frequency Domain Wavelet based ECG Steganography

The sender side of the proposed steganography technique consists of four integrated stages as shown in Fig. 2.6. The proposed technique is designed to ensure secure information hiding with minimal distortion of the host signal. Moreover, this technique contains an authentication stage to prevent unauthorized users from extracting the hidden information.

2.4.1 Stage 1: Encryption

The aim of this stage is to encrypt the patient confidential information in such a way that prevents unauthorized persons - who do not have the shared key- from accessing patient confidential data. In this stage XOR ciphering technique is used with an ASCII coded shared key, which will play the role of the security key. XOR ciphering is selected because of its simplicity. As a result, XOR ciphering can be easily implemented inside a mobile device. Figure 2.4 shows an example of what information could be stored inside the ECG signal [75].

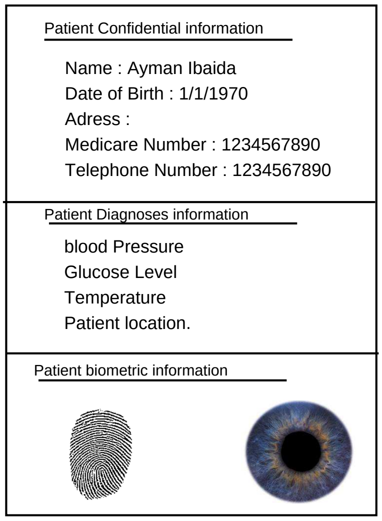


Figure 2.4: Original data consisting of patient information and sensor readings as well as patient biometric information.

2.4.2 Stage 2: Wavelet Decomposition

Wavelet transform is a process that can decompose the given signal into coefficients representing frequency components of the signal at a given time. Wavelet transform can be defined as shown in Eq. 2.6.

$$C(S, P) = \int_{-\infty}^{\infty} f(t)\psi(S, P)\, dt \tag{2.6}$$

where ψ represents wavelet function. S and P are positive integers representing transform parameters. C represents the coefficients which is a function of scale and position parameters [72]. Wavelet transform is a powerful tool to combine time domain with frequency domain in one transform. In most applications discrete signals are used. Therefore, Discrete Wavelet Transform (DWT) must be used instead of continuous wavelet transform. DWT decomposition can be performed by applying wavelet transform to the signal using band filters. The result of the band filtering operation will be two different signals, one will be related to the high frequency components and the other related to the low frequency components of the original signal. If this process is repeated multiple times, then it is called multi-level packet wavelet decomposition. Discrete Wavelet transform can be defined as in Eq. 2.7

$$W(i,j) = \sum_i \sum_j X(i)\Psi_{ij}(n) \tag{2.7}$$

where $W(i,j)$ represents the DWT coefficients. i and j are the scale and shift transform parameters, and $\Psi_{ij}(n)$ is the wavelet basis time function with finite energy and fast decay. The wavelet function can be defined as in Eq. 2.8

$$\Psi_{ij}(n) = 2^{-i/2}\Psi(2^{-i}n - j) \tag{2.8}$$

In our proposed technique, a 5-level wavelet packet decomposition has been applied to the host signal. Accordingly, 32 sub-bands resulted from this decomposition process as shown in Fig. 2.9. In each decomposition iteration the original signal is divided into two signals. Moreover, the frequency spectrum is distributed on these two signals. Therefore, one of the resulting signals will represent the high frequency component and the other one represents the low frequency component. Most of the important features of the ECG signal are related to the low frequency signal. Therefore, this signal is called the approximation signal (A). On the other hand, the high frequency signal represents mostly the noise part of the ECG signal and is called detail signal (D). As a result, a small number of the 32 sub-bands will be highly correlated with the important ECG features while the other sub bands will be correlated with

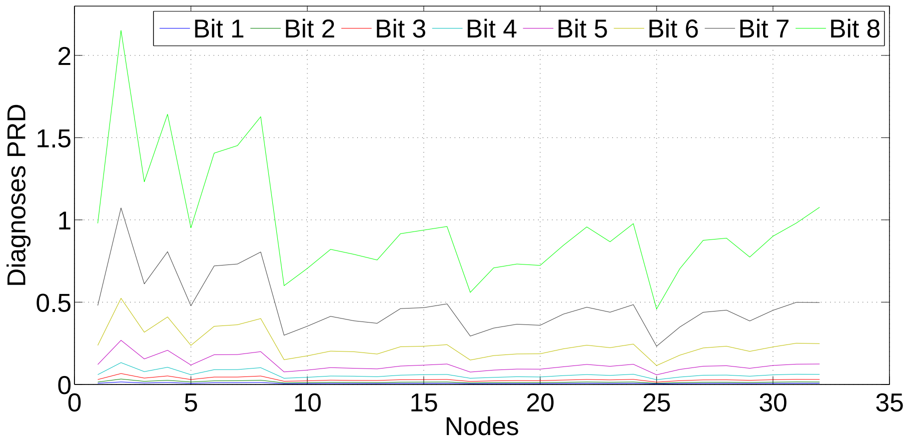


Figure 2.5: Effect of applying steganography using different levels on the resulting PRD for each of the 32 sub-bands.

the noise components in the original ECG signal [5]. Therefore, in our proposed technique different number of bits will be changed in each wavelet coefficient (usually called steganography level) based on its sub-band. As a result, a different steganography level will be selected for each band in such a way that guarantees the minimal distortion of the important features for the host ECG signal. The process of steganography levels selection was performed by applying lot of experimentation as shown in Fig. 2.5. It is clear that, hiding data in some sub-bands will highly affect the original signal, while hiding in other sub-bands would result in small distortion effect. Accordingly, the selected steganography levels for bands from 1 to 17 is 5 bits and 6 bits for the other bands.

2.4.3 Stage 3: The embedding operation

At this stage the proposed technique will use a special security implementation to ensure high data security. In this technique a scrambling operation is performed using two parameters. First is the shared key known to both the sender and the receiver. Second is

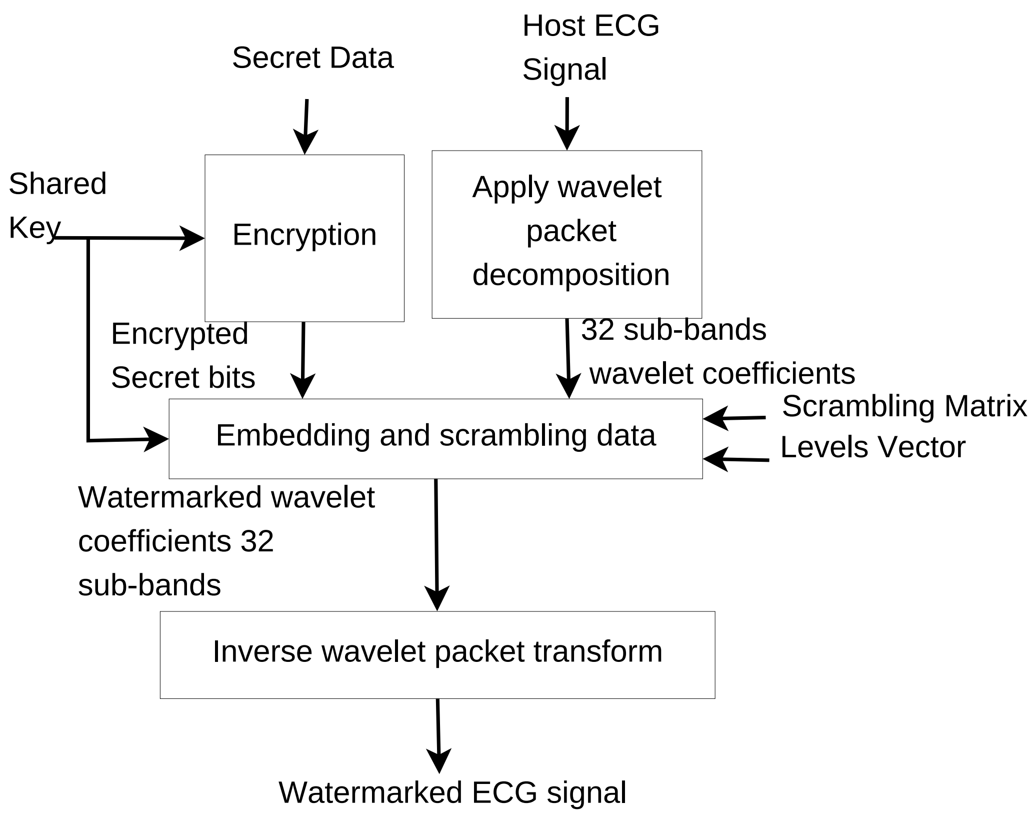


Figure 2.6: Block diagram of the sender steganography which includes encryption, wavelet decomposition and secret data embedding.

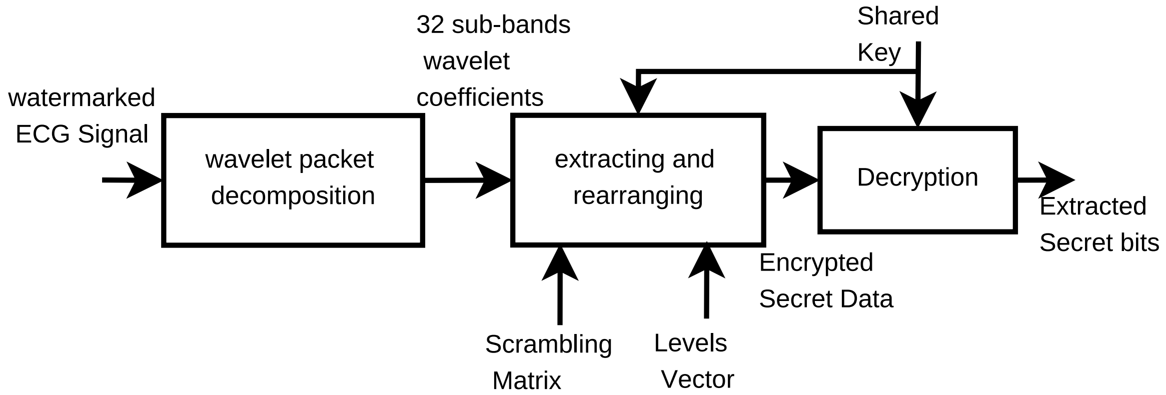


Figure 2.7: Block diagram of the receiver steganography which includes wavelet decomposition, extraction and decryption

the scrambling matrix, which is stored inside both the transmitter and the receiver. Each transmitter/receiver pair has a unique scrambling matrix defined by Eq. 2.9

$$S = \begin{bmatrix} s_{1,1} & s_{1,2} & \cdots & s_{1,32} \\ s_{2,1} & s_{2,2} & \cdots & s_{2,32} \\ \vdots & \vdots & \ddots & \vdots \\ s_{128,1} & s_{128,2} & \cdots & s_{128,32} \end{bmatrix} \tag{2.9}$$

where S is a 128×32 scrambling matrix. s is a number between 1 and 32. While building the matrix we make sure that the following conditions are met:

- The same row must not contain duplicate elements
- Rows must not be duplicates.

The detailed block diagram for the data embedding process is shown in Fig. 2.8. The embedding stage starts with converting the shared key into ASCII codes, therefore each character is represented by a number from 1 to 128. For each character code the scrambling sequence fetcher will read the corresponding row from the scrambling matrix. An example of a fetched row can be shown in Eq. 2.10

$$S_r = \begin{matrix} 32 & 22 & 6 & 3 & 16 & 11 & 30 & 7 \\ 28 & 17 & 14 & 8 & 5 & 29 & 21 & 25 \\ 31 & 27 & 26 & 19 & 15 & 1 & 23 & 2 \\ 4 & 18 & 24 & 13 & 9 & 20 & 10 & 12 \end{matrix} \tag{2.10}$$

The embedding operation performs the data hiding process in the wavelet coefficients according to the sub-band sequence from the fetched row. For example, if the fetched row is as in Eq. 2.10, the embedding process will start by reading the current wavelet coefficient in sub-band 32 and changing its LSB bits. Then, it will read the current wavelet coefficient in sub-band 22 and changing its LSB bits, and so on. On the other hand, the steganography level is determined according to the level vector which contains the information about how many LSB bits will be changed for each sub-band. For example if the data is embedded in sub-band 32 then 6 bits will be changed per sample, while if it is embedded into wavelet coefficient in sub-band 1 then 5 LSB bits will be changed.

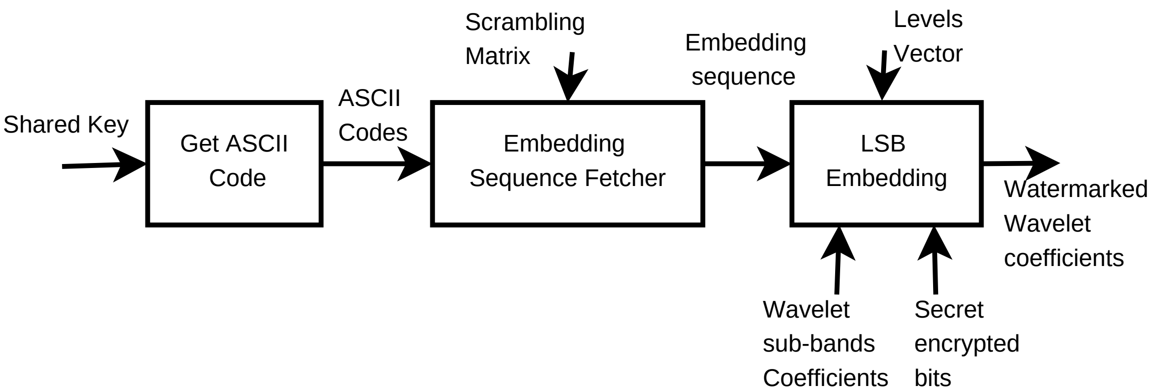


Figure 2.8: Block diagram showing the detailed construction of the watermark embedding operation

2.4.4 Stage 4: Inverse wavelet re-composition

In this final stage, the resultant watermarked 32 sub-bands are recomposed using inverse wavelet packet re-composition. The result of this operation is the new watermarked ECG signal. The inverse wavelet process will convert the signal to the time domain instead of combined time and frequency domain. Therefore, the newly reconstructed watermarked

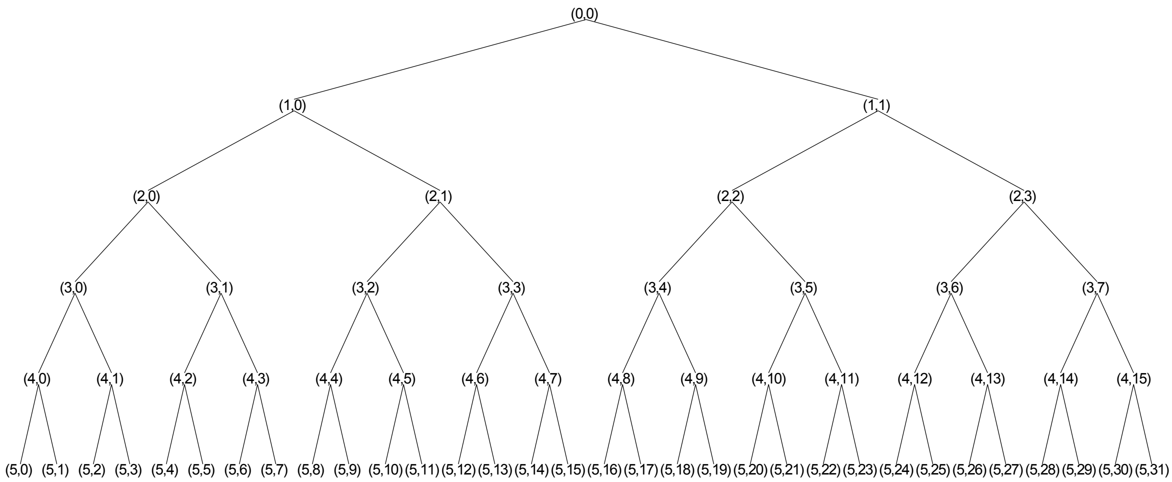


Figure 2.9: 5-level wavelet decomposition tree showing 32 sub-bands of ECG host signal and the secret data will be hidden inside the coefficients of the sub-bands

ECG signal will be very similar to the original unwatermarked ECG signal. The detailed embedding algorithm is shown in Algorithm 1.

The algorithm starts by initializing the required variables. Next, the coefficient matrix will be shifted and scaled to ensure that all coefficients values are integers. Then, the algorithm will select a node out of 32 nodes in each row of the coefficient matrix. The selection process is based on the value read from the scrambling matrix and the key. The algorithm will be repeated until the end of the coefficient matrix is reached. Finally, the coefficient matrix will be shifted again and re-scaled to return its original range and inverse wavelet transform is applied to produce the watermarked ECG signal.

2.4.5 Watermark extraction process

To extract the secret bits from the watermarked ECG signal, the following information is required at the receiver side.

Algorithm 1 The embedding algorithm

```
bits: the secret bits array
bs: size of bits array
b: index of the current bit of the secret bits array
ecg : the host ECG signal
key: encryption key
kl: size of key
kc: index of the current character in the secret key
scra: The scrambling matrix 128 × 32
sl: steganography embedding level
coef: wavelet packet coefficients at level 5 which has 32 columns
cs: number of rows of coef matrix
s: index of the current row of the coefficients matrix
coef ← coef + 20
coef ← coef × 10000
b ← 1
kc ← 1
s ← 1
while s < cs do
   for doi = 1 TO 32
      node ← scra(ascii(key(kc)), i)
      bnode ← sl(node)
      for doj = 1 TO bnode
         coef(s, node) ← embed(bits(b), position(j))
         b ← b + 1
         if b > bs then
            b ← 1
         end if
      end for
   end for
   s ← s + 1
   kc ← kc + 1
   if kc > kl then
      kc ← 1
   end if
end while
coef ← coef/10000
coef ← coef − 20
necg ← wavletpacketrecomposition(coef)
```

1. The shared key value
2. Scrambling matrix
3. Steganography levels vector

The stages of the extraction process can be shown in Fig. 2.7. The first step is to apply 5-level wavelet packet decomposition to generate the 32 sub-bands signals. Next, using the shared key and scrambling matrix the extraction operation starts extracting the secret bits in the correct order according to the sequence rows fetched from the scrambling matrix. Finally, the extracted secret bits are decrypted using the same shared key. The watermark extraction process is almost similar to the watermarking embedding process shown in Algorithm 1 except that instead of changing the bits of the selected node, it is required to read values of the bits in the selected nodes, and then resetting them to zero.

2.4.6 Security Analysis

The security of the proposed algorithm is mainly based on the idea of having several parameters shared between the transmitter and the receiver entities. Any change in any parameter will lead to extraction of wrong data. Accordingly, the receiver and transmitter should agree on the following information:

1. The scrambling matrix
2. The encryption key (ASCII text string) i.e shared secret.
3. Steganography levels vector

As a result, even if the key is stolen the attacker will require to know the scrambling matrix as well as the steganography levels vector. The scrambling matrix is stored inside the transmitter/receiver pair and it will not be transmitted under any circumstance. For example, each patient could have his own device from his medical service provider and the matrix is burnt on this device. Therefore, for each pair of transmitter and receiver, it must be a unique scrambling matrix. As a result the total number of devices pairs (that can be supported) with a unique scrambling matrix can be calculated as in Eq. 2.11.

$$N = 32! \times 128! \approx 3.8562 \times 10^{+215} \tag{2.11}$$

As shown in Eq. 2.11 the total number of devices that we can support is larger than the IPv6 protocol address space. On the other hand, the sequence of rows fetched from the scrambling matrix is tied to the user defined key. As a result, the longer the key is, the stronger the steganographic technique will be. To guarantee maximum security, the length of the key used (L_{key}) should satisfy the following condition stated in Eq. 2.12.

$$L_{key} = Max(\frac{B}{180}, M) \tag{2.12}$$

where B is the size of the embedded data in bits and M represents the minimum key size. Accordingly, Table 2.2 shows the probabilities to break the proposed technique using different key lengths and the minimum data size that can be hidden to achieve the maximum security for each key length.

Table 2.2: Security strength for different key lengths

key length	minimum data size (bits)	probabilities
4	720	1E+224
8	1440	2.7E+232
16	2880	2E+249
32	5760	1E+283
40	7200	7.4E+299

The amount of data that can be stored inside the ECG host signal using the proposed model totally depends on the steganography levels vector. In our proposed model and for ECG with 10 seconds length and sampling rate of 360, 2531 bytes (2.4KB) of data can be embedded inside ECG host signal. The amount of data that can be stored is calculated using Eq. 2.13.

$$b = \frac{t \times f_s}{32} \times 180 \tag{2.13}$$

where b is the total number of bits stored, t is the total signal time in seconds and f_s is the sampling frequency.

2.5 Diagnosability measurement of the watermarked ECG signal

To evaluate the diagnosability of the watermarked ECG signal, the work done by Amjed S. Al-Fahoum [5] has been implemented and used as a diagnosability measure to determine the effect of the watermarking process on the usability of the resultant ECG for diagnosis purposes. In this model a 5-level wavelet decomposition is applied to the original and watermarked ECG signal. As a result, the original signal will be divided into a number of sub-bands denoted by A_5, D_5, D_4, D_3, D_2 and D_1. It is found that band A_5 includes most of the T-wave contribution and some of the P-wave contribution. Therefore, its weight should include the significance of P and T. Moreover, band D_5 includes most of the P-wave contribution, part of the T-wave contribution, and a relatively low percentage of the QRS-complex contribution. The weight of D_5 should maintain the highest weight contribution of P, T, and QRS. Band D_4 also contains most of the QRS-complex contribution, and a little portion of P-wave. The weight of D_4 is higher than A_5 but lower than D_5. D_3 includes some portions of the QRS-complex, and so its weight is lower than QRS weight itself. Bands D_2, and D_1 are weighted less than any other band as they do not contribute to the spectrum of any of the main features. However, they cannot be excluded where late potentials and delta waves may exist. The researchers used a heuristics weights to mark the contribution of each sub-band. The weights used are shown in Table 2.3.

After applying 5-level wavelet decomposition, Percentage Residual Difference (PRD) measure of each sub-band is calculated. as shown in Eq. 2.14

$$WPRD_j = \sqrt{\frac{\sum_{i=1}^{N} (c_i - \tilde{c}_i)^2}{\sum_{i=1}^{N} (c_i^2)}} \tag{2.14}$$

where c_i is the original coefficient within sub-band j and $\tilde{c}_i$ is the coefficient of sub-band j for the watermarked signal. Finally, to find the Weighted Wavelet PRD Eq. 2.15 is used

$$WWPRD = \sum_{j=0}^{N_L} w_j WPRD_j \tag{2.15}$$

where N_L is the total number of sub-bands, w_j denotes the weight value corresponding to sub-band j and $WPRD_j$ represents the calculated wavelet based PRD for sub-band j.

Table 2.3: Weights for each sub-band used in measuring Diagnosability

Wavelet Bands	A5	D5	D4	D3	D2	D1
Bands weights	6/27	9/27	7/27	3/27	1/27	1/27

2.6 Experiments and results

2.6.1 Time domain steganography evaluation results

A testbed of 81 ECG segments has been set up by collecting ECGs from The Creighton University Ventricular Tachyarrhythmia Database [71]. Each ECG segment has a length of 10 seconds with 250Hz sampling frequency and consists of 2500 samples. Consequently, secret bits have been hidden in different positions (8th,16th,24th, and 32nd bits) using four special ranges. To evaluate the proposed steganography technique, the resultant ECG signals are compared with their original signals. The percentage residual difference (PRD) is used for this purpose.

Table 2.4: Different PRD values for Different range lengths

number of changed LSB	**Range Length**	**PRD**
0	2	0.02
1	4	0.0735
2	8	0.1946
3	16	0.4493

Figure 2.10 shows eight ECG segments divided into two groups, where the first group contains the original normal ECG segment and three resultant host signals when we applied different special ranges and had data hidden in secret bit positions(see Table 2.1). The

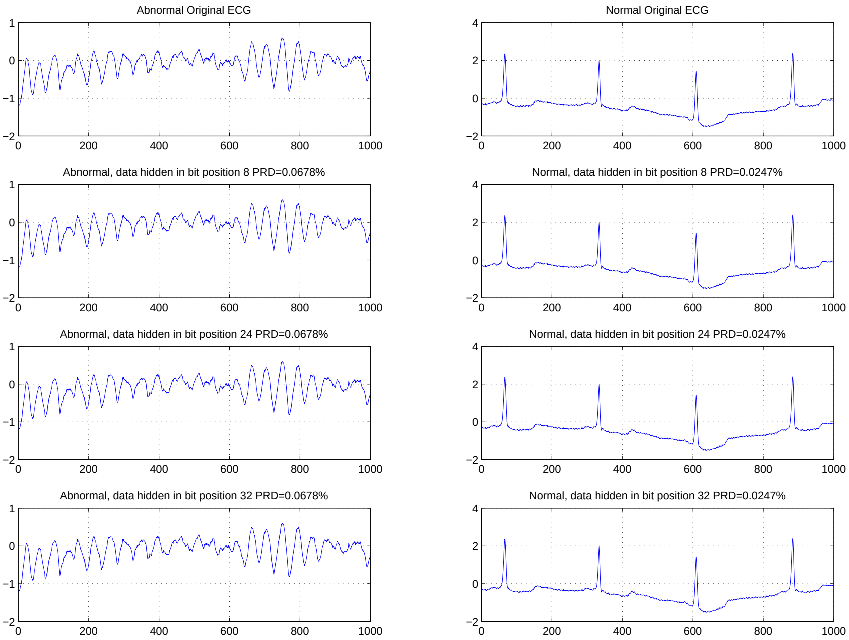


Figure 2.10: Different Cases of Watermarked ECG Signals After Hiding Secret Information in Bit 8,24 and 32 of Binary ECG Data

second group contains the original abnormal ECG segment in addition to three resultant host signals using the same special ranges and secret bit positions that we used in the first group. This figure shows that despite hiding in different secret positions, PRD error measurement remains very low and is the same for all cases of the same ECG signal. However, PRD error measurements vary slightly for normal and abnormal groups of host ECG signals. This fact of being able to generate a constant PRD value using different bit positions is a powerful characteristic of the proposed steganography technique that enables users to hide their secret messages in highly secure positions without affecting the important features of the ECG signal.

The evaluation of the proposed technique shows the strength of the hiding model by performing simulated attacks to prove the ability of extracting the embedded data completely although the least significant bits of the host signal have been changed deliberately. Table 2.4 shows several cases of the simulated attacks on the host signal and the required range lengths to protect the embedded data in addition to the resultant PRD of each case. For example, if the first LSB is changed, then a range length of 4 is required to protect the secret hidden bits. If the least two significant bits are changed, a range length of 8 is required to achieve the secret bit's protection. Finally, a range length of 16 is required to protect the secret message from changing the first three LSBs. As shown in Table 2.4, the PRD values increase when the number of the changed bits is increased. Consequently, the more the secret data is protected, the more distorted will be the original signal.

In our experiments, each 10 sec ECG segment was 10,000 bytes (2500 samples x 4 bytes/sample= 10,000) and in each sample we modified 1 bit to host the secret bits. Overall, we could host 2500 bits = 312.5 bytes. As an ECG host signal size increases we are able to add more secret bits by replacing existing bits of host data appropriately. Finally, to generalize our experiments WWPRD is used in addition to PRD measure. Different types of ECG signal with different diseases such as Ventricular Tachycardia, Ventricular Fibrillation, and Premature Ventricular Contraction, are used as well as normal samples in experimentation. Table 2.5 clearly shows how the PRD values and WWPRD does not vary widely for different types of ECG signals.

Table 2.5: PRD results for different data type and different ECG signals

ECG Type	PRD	WWPRD
Normal	0.0103	0.01
VT	0.0047	0.0052
VF	0.0092	0.0109
PVC	0.012	0.0176

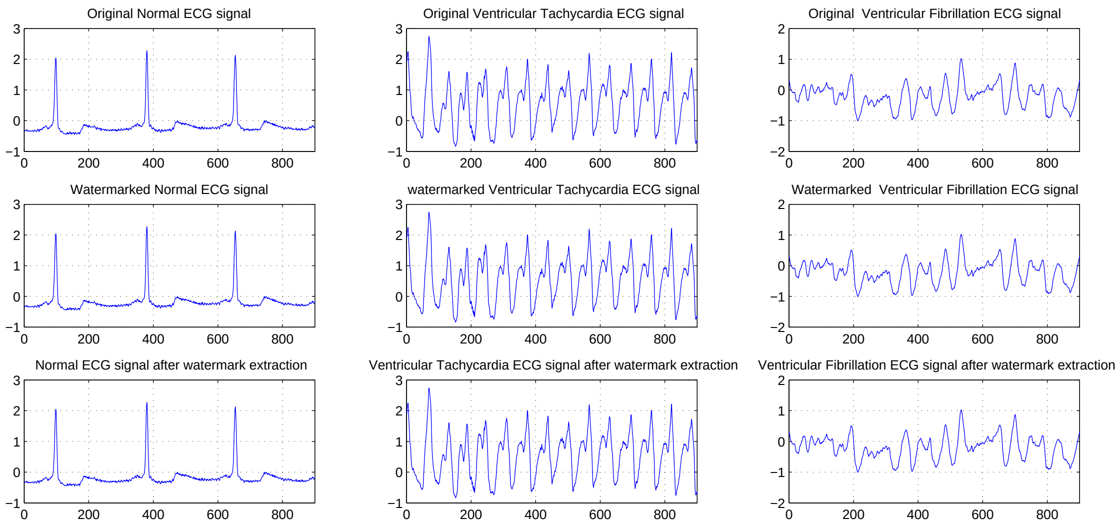


Figure 2.11: ECG signals for normal, VT and VF signal before applying the steganography operation and after the steganography operation as well as after extracting the hidden data.

2.6.2 Frequency domain steganography evaluation results

In this section, three different types of ECG signals were used for experimentation. A testbed of 59 ECG samples was used for experimentation. The set of samples consisted of 19 normal (NSR) ECG samples, 27 Ventricular fibrillation ECG samples and 13 Ventricular Tachycardia ECG samples. Each sample is 10 seconds long with 250 Hz sampling frequency.

To evaluate the proposed model, the PRD (percentage residual difference) is used to measure the difference between the original ECG host signal and the resulting watermarked

ECG signal as shown in Eq. 2.16.

$$PRD = \sqrt{\frac{\sum_{i=1}^{N}(x_i - y_i)^2}{\sum_{i=1}^{N} x_i^2}} \tag{2.16}$$

where x represents the original ECG signal, and y is the watermarked signal.

Alternatively, to evaluate the diagnostic distortion caused by the watermark, a wavelet based PRD is used as detailed in the previous section. These measures have been calculated for each sample. Accordingly, to measure distortion caused by the extraction process, PRD and diagnosis PRD have been calculated. Finally to evaluate the reliability of the extracted information, bit error rate has been used as shown in Eq. 2.17

$$BER = \frac{B_{err}}{B_{total}} \times 100\% \tag{2.17}$$

where BER represents the Bit Error Rate in percentage, B_{err} is the total number of erroneous bits and B_{total} is the total number of bits.

Table 2.6 shows the results obtained for 18 normal ECG samples. It can be seen that a maximum PRD measured was 0.6%. Secondly, it can be noticed that the difference between the normal PRD and the wavelet based PRD for diagnoses measurement is very small. Accordingly, this proves that the watermarking process does not affect the diagnosability. Finally, this table shows the PRD measured after extracting the watermark. It is obvious from the table that removal of the watermark will have a small impact on the PRD value. As a result, the ECG signal can still be used for diagnosis purposes after removing the watermark.

Table 2.6: PRD results for different normal ECG segments

Sample No	**PRD %**	**WWPRD %**	**PRD % extracted**	**WWPRD % extracted**
1	0.43446	0.39338	0.57647	0.52692
2	0.56804	0.4371	0.79583	0.59282
3	0.59837	0.44557	0.80906	0.62531
4	0.51656	0.43133	0.72957	0.60578
5	0.53641	0.41908	0.72213	0.56855
6	0.58602	0.43386	0.80906	0.61782
7	0.5064	0.62222	0.70934	0.873
8	0.26013	0.59378	0.35179	0.81591
9	0.4634	0.6083	0.63565	0.82741
10	0.51913	0.63338	0.70037	0.85416
11	0.5055	0.61394	0.6874	0.84694
12	0.45053	0.595	0.60611	0.79233
13	0.45692	0.50512	0.61693	0.68123
14	0.41861	0.50547	0.56098	0.68459
15	0.36499	0.42618	0.50238	0.59443
16	0.42648	0.33541	0.57897	0.45032
17	0.44176	0.34352	0.59529	0.46326
18	0.42957	0.34337	0.59061	0.47203

To guarantee unbiased results, we also experimented with VT and VF samples and the results are shown in Tables 2.8 and 2.7 respectively. It is obvious from these results that the maximum PRDs for VT host signal is only 0.5% and 1% for VF. This encouraging result clearly demonstrates that the watermarked ECG signals can be used for diagnoses. Figure 2.11 shows three ECG signal types, and the resultant watermarked signals before and after watermark extraction process.

Table 2.7: PRD results for Ventricular Tachycardia ECG samples

Sample No	**PRD %**	**WWPRD %**	**PRD % extracted**	**WWPRD % extracted**
1	0.24973	0.25439	0.33705	0.34314
2	0.27853	0.30552	0.37474	0.41137
3	0.29892	0.29903	0.40912	0.41751
4	0.24248	0.2822	0.33029	0.38589
5	0.26566	0.26055	0.37705	0.36925
6	0.27017	0.25964	0.37263	0.36044
7	0.28042	0.27871	0.37983	0.38101
8	0.47009	0.5803	0.49603	0.65555
9	0.16381	0.28317	0.22884	0.4103
10	0.19697	0.30666	0.27143	0.41038
11	0.27231	0.26876	0.37796	0.38309
12	0.32276	0.32799	0.43247	0.44159

Previous results have been obtained by using the same scrambling matrix. To generalize our results we performed the same experiments and calculated the average PRD values for different cases of scrambling matrices. Table 2.9 shows 10 different cases taken and their corresponding average PRD values. It can be clearly seen how the values are approximately equal for different cases. The obtained results further prove that our proposed technique will cause minimum distortion for different cases of the scrambling matrix. This is clearly shown in Fig. 2.12.

To validate diagnosability of the digitally processed ECGs, two specialist doctors were consulted. Sixty ECG Segments (each 10 seconds length) for both normal sinus rhythm and abnormal (Ventricular Tachycardia, Ventricular Fibrillation) cases were shown to them before and after watermarking, and also after removal of watermarks. They were asked the

Table 2.8: PRD results for Ventricular Fibrillation

Sample No	**PRD %**	**WWPRD %**	**PRD % extracted**	**WWPRD % extracted**
1	0.65061	0.84787	0.89994	1.1713
2	0.58442	0.78362	0.7944	1.0715
3	0.54158	0.78223	0.74391	1.0733
4	0.40013	0.41339	0.55157	0.56329
5	0.30265	0.38706	0.41009	0.53588
6	0.30569	0.4287	0.41517	0.58034
7	0.20551	0.43169	0.27795	0.5915
8	0.19213	0.31981	0.26104	0.43105
9	0.47881	0.50826	0.66257	0.71434
10	0.38448	0.3726	0.52747	0.51307
11	0.48817	0.4968	0.66364	0.677
12	0.48814	0.48671	0.66386	0.66023
13	0.41675	0.45275	0.57517	0.63255
14	0.45104	0.46792	0.60064	0.61633
15	0.38447	0.39853	0.5267	0.55549
16	0.32621	0.32604	0.4417	0.43772
17	0.66713	0.93723	0.91967	1.2995
18	0.79696	1.3659	1.0728	1.8144
19	1.0732	0.96977	1.454	1.2749
20	1.0514	0.99681	1.4297	1.3527
21	0.84326	0.99305	1.1607	1.3767
22	0.71059	0.97451	0.96605	1.3123
23	0.61859	1.066	0.8545	1.4892
24	0.66063	1.0684	0.91559	1.4727
25	0.79912	1.2993	1.1125	1.8225
26	0.92514	1.1759	1.2979	1.6486

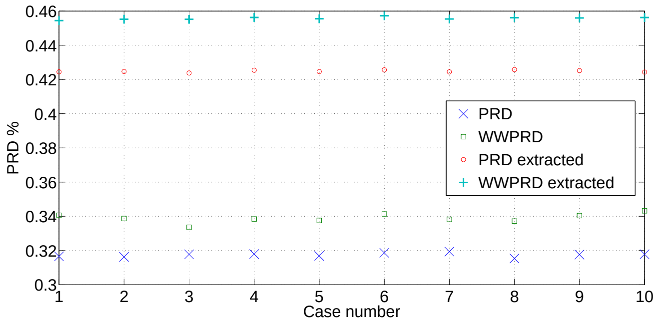


Figure 2.12: Average PRD results for different scrambling matrices

Table 2.9: Average PRD results for different scrambling matrices

Case No	**PRD %**	**WWPRD %**	**PRD % extracted**	**WWPRD % extracted**
1	0.316605	0.340602	0.424484	0.454439
2	0.316308	0.338725	0.424643	0.455201
3	0.31775	0.333594	0.423802	0.455167
4	0.317904	0.338432	0.425399	0.456254
5	0.316824	0.337616	0.42462	0.455501
6	0.318639	0.341384	0.425632	0.457272
7	0.319365	0.338233	0.424424	0.455365
8	0.315372	0.337196	0.425761	0.45611
9	0.317598	0.34042	0.425144	0.455971
10	0.317847	0.343214	0.424329	0.456203

following questions:

- How similar is the original and the watermarked ECG?
- Can the watermarked ECG be used for diagnoses?

Both the specialist doctors admitted that the similarity is so high that the difference is undetectable and both the watermarked and un-watermarked signals can be used for diagnoses. The detailed results are shown in Table 2.10.

Table 2.10: Doctors' diagnoses results

Doctor	Normal Similarity	Normal Diagnosability	Abnormal Similarity	Abnormal Diagnosability
1	99%	Yes	99%	Yes
2	100%	Yes	100%	Yes

2.7 Summary

In this chapter, two novel steganography algorithms were proposed to hide patient information as well as diagnostics information inside an ECG signal. These techniques will provide a secured communication and confidentiality in the current e-health system. A time domain steganography technique is proposed which is based on applying special range transform to provide the ability to hide the data in any bit position with minimum error. The second method is a frequency domain technique. In this model, a 5-level wavelet decomposition is applied. A scrambling matrix is used to find the correct embedding sequence based on the user defined key. Steganography levels (i.e. number of bits to hide in the coefficients of each sub-band) are determined for each sub-band by experimental methods. In this chapter, we tested the diagnoses quality distortion. It was found that the resultant watermarked ECG can be used for diagnoses and the hidden data can be totally extracted.

Chapter 3

ECG Lossy Compression

This Chapter answers the second research question discussed in Section 1.2. The size of the ECG signal is huge especially in continuous remote monitoring systems. This chapter discusses the available lossy compression technique that are proposed to overcome this problem. Moreover, this chapter will highlight the limitation of the available techniques. Furthermore, we will discuss the fractal model and how it can be modified to be used in designing cloud enabled compression technique. Fractal model is also improved and the proposed cloud enabled compression technique is improved to make it suitable for implementation inside the client side rather than the cloud side. Section 3.1 explains the problem statement and discusses the motivation behind this work. Moreover, Section 3.2 explains the current available lossy compression techniques and highlights their limitations. Section 3.3 explains the theory of fractals combined with the theory of Iterated Function System (IFS). Furthermore, the proposed cloud enabled compression and decompression techniques are explained in more details in Section 3.4. To implement fractal compression within the client side the proposed fractal model is modified and we call the new proposed technique as fast fractal model. Its concept and equation derivation as well as compression and decompression algorithms are explained in Section 3.5. Section 3.6 discusses the evaluation parameters used to evaluate the proposed technique. And, Section 3.7 explains the results for the proposed cloud enabled compression as well as the fast fractal compression technique. It shows how both algorithms can perform better compared with other compression techniques. Moreover, this section compares the performance of the normal fractal and the fast fractal techniques. Finally, Section 3.8 summarizes this chapter.

3.1 Introduction

The use of E-health applications is increasing to a great extent around the globe. Many health-care organizations such as insurance companies, hospitals and government health sectors, require access to patient information and records including their archived biomedical signals. Therefore, it is required to store patient records in a centralized repository which will provide access services to other health-care organizations. Cloud services can be a solution to serve this purpose [53; 8; 24; 23]. As shown in Fig. 3.1, a cloud health-care system consists of Body-Sensor-Nodes (BSN) collecting patient biological signals (e.g ElectroCardioGram (ECG) and photoplethysmogram (PPG)) which are then sent to a hospital server. ECG signals require a large amount of storage capability due to their large size. In order to minimize storage requirements ECG compression is implemented in the cloud, to make use of its powerful processing resources, as it is the central point to be accessed by different health agencies. Moreover, in case of remote patient monitoring systems, body sensor nodes will send the collected ECG signals to the patient PDA device then to cloud and the compression operation will be implemented within the cloud. On the other hand, clients trying to access patient records will receive the compressed information and it will be decompressed inside client devices. The proposed decompression algorithm is designed in such a way that allows the user to retrieve part of the compressed file (i.e. ECG segment) without decompressing the whole file. Moreover, the user (such as a doctor) can request to receive the compressed ECG signal relevant to a specified period of time, for example, to check the effect of taking medicine at a specific time on the patients ECG. The proposed technique can achieve this by sending the required part of the ECG compressed file without adding any more headers or overhead. Finally, a doctor's device can decompress the part required without decompressing the whole ECG compressed file. As a result, clients can see the ECG signal as soon as the cloud starts to receive ECG from BSN and they do not need to wait for receiving the whole file (which can last for more than 12 hours in ECG continuous holter monitoring scenario) before seeing the resultant ECG signals. Therefore, the proposed fractal based compression technique is suitable for hosting and retrieval of compressed ECG data on a cloud. The compression operation is not a real-time operation and can therefore be performed offline using the powerful cloud resources. Moreover, as in many cases, the doctors would like to check the signal at specific time, the compression technique provides this feature to decompress the file starting from any position but not in a real-time manner. Another aspect is taken into consideration, the proposed method is modified to make it more reliable to be implemented within the client devices rather than within the cloud.

Generally, compression techniques can be divided into two main categories: lossless and

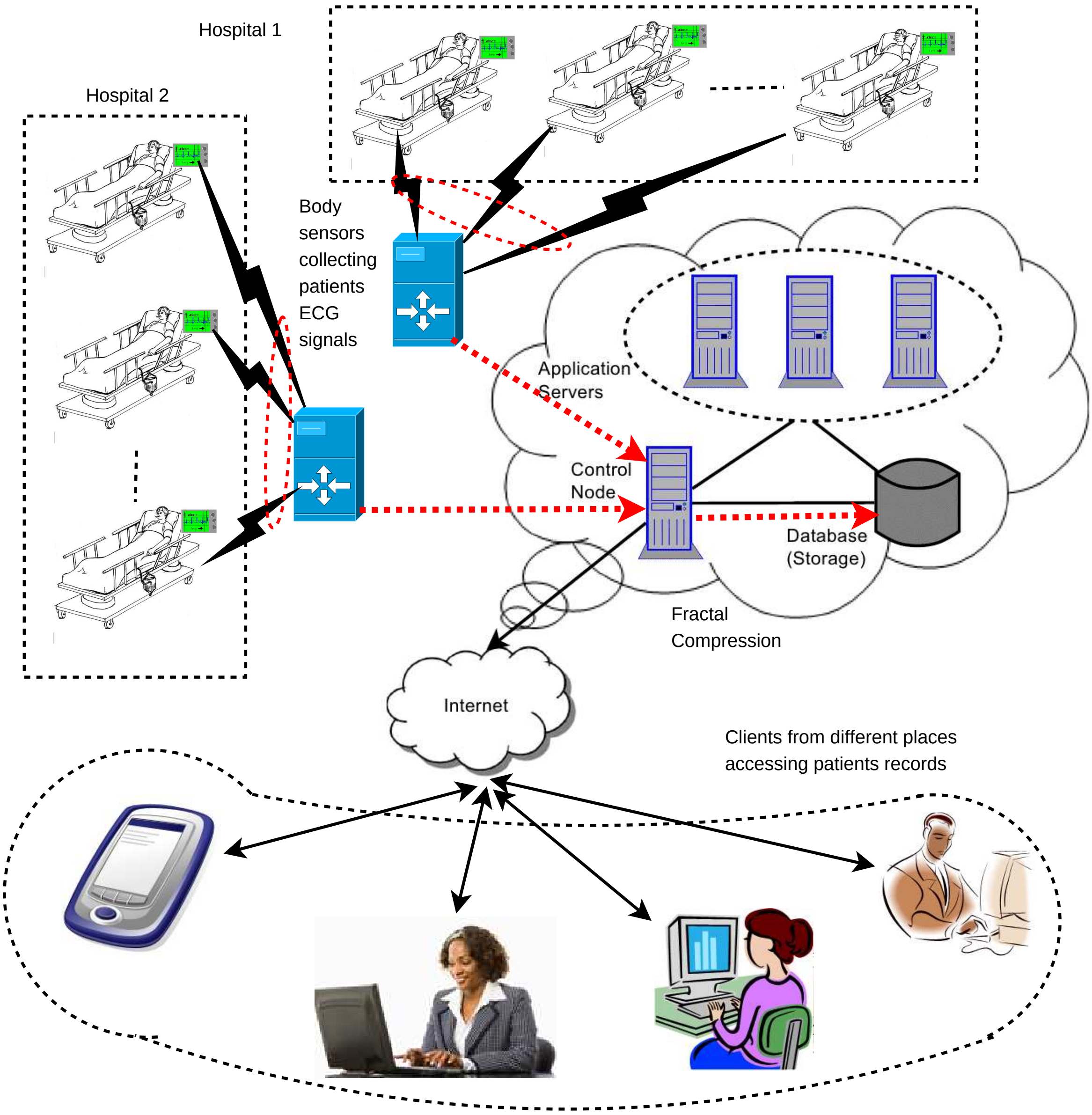


Figure 3.1: E-health Cloud consist of Cloud servers, BSN, hospital servers, and Remote patient monitoring sensors. The signals will be transmitted to the cloud, the compression technique is implemented inside the cloud to guarantee faster performance. Different health agencies such as health insurance, doctors, nurses, researchers, and government health sectors can access the cloud and retrieve the desired information in compressed format, and decompress on their devices after retrieving the information.

lossy compression. Lossless compression techniques guarantee full reconstruction of the original signal without any information loss. On the other hand, lossy compression can reconstruct an approximated version of the original signal. In lossy compression, it is possible to achieve higher compression ratios with small differences between the original signal and the reconstructed signal. Furthermore, ECG compression techniques can be classified into two major types: time domain compression techniques [7; 22; 29] and transform domain compression techniques [35; 45; 60].

Most of the proposed compression techniques are based on R peak detection or other ECG parameters detection. Therefore, they do not provide high performance when applied on abnormal ECG signals since it is extremely complicated to extract ECG signal parameters [13]. The periodic characteristic of the ECG signal and the inter and beat correlation can be powerful features to be used in ECG compression as shown in Fig. 3.2. Therefore, In this chapter, fractal model is utilized to capture ECG self similarity characteristics. Fractal technique have been developed to approximate the ECG signal. The proposed technique does not require the use of QRS detection or any ECG parameters extraction. As a result, this technique is capable of achieving high compression ratio with both normal and abnormal ECG signals. Moreover, the decompression process can partially decompress the compressed file starting from any point in the file and for any duration.

Compression techniques can be evaluated based on three important measures which are compression ratio, signal reconstruction error and compression performance. Compression ratio can be defined as the ratio between the original ECG signal file size and the compressed file size. The compression technique with high compression ratio and low distortion effect

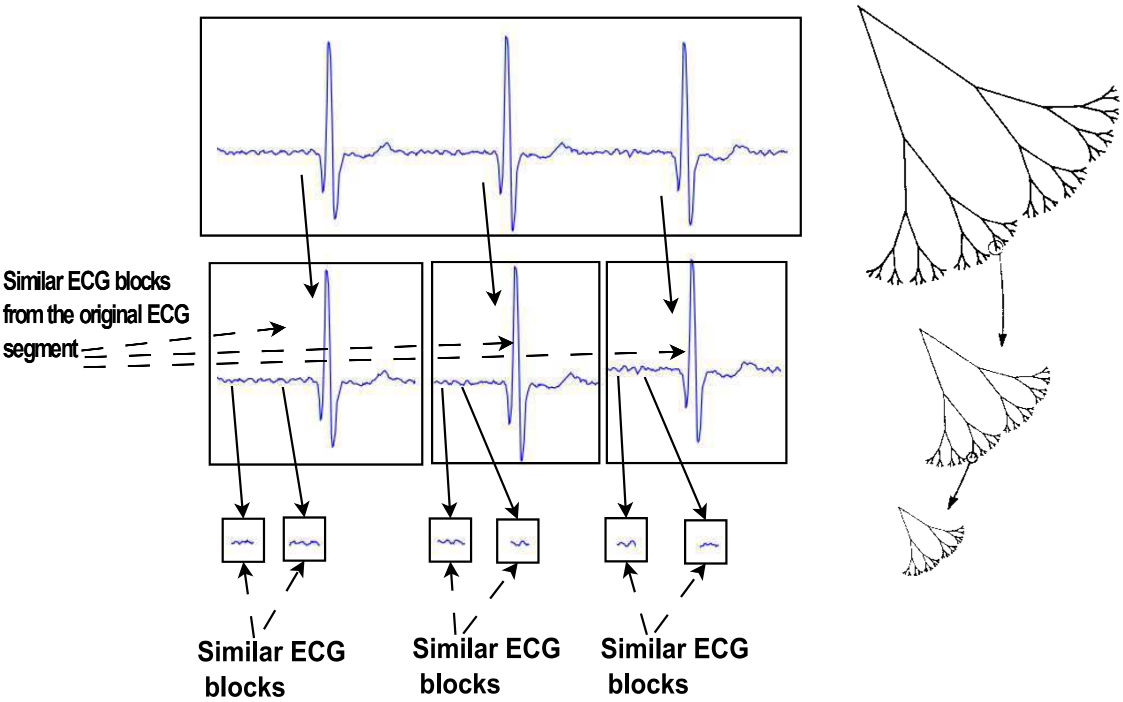


Figure 3.2: ECG fractal self similarity compared with fractal object self similarity. The figure on the left shows the ECG signal self similarity for different block sizes. The figure on the right shows a fractal object and how it is self similar. Fractal self-similarity feature of ECG signal is used in compression

provides high reliability. In lossy compression techniques, the signal reconstruction error has a great impact on the usability of the compression technique. There are numerous methods to measure this error such as Percentage Residual Difference, Wavelet Weighted PRD, Signal to Noise Ratio and Root Mean Square Error. Experimental results have shown that the proposed fractal technique outperformed other compression models in the chosen area.

3.2 Related Work

Many researchers have conducted investigation on lossy ECG compression. SangJoon Lee [47] proposed a new real time ECG compression technique. Firstly, the researcher reduces the ECG signal to half of its original sampling rate. Secondly, they apply sample differencing followed by R peak detection. Next, DCT transform is applied on two consecutive ECG periods , then, the floating point DCT coefficients will be converted to integers. and accumulated error have been calculated. Finally, Huffman coding is used as an entropy coding.

Eddie B. [73] proposed a new lossy compression technique based on converting the 1D ECG signal into 2D ECG image. The conversion process consist of QRS detector, period length normalization , period preprocessing and image transform. Their work focused on the preprocessing stage by clamping all the periods to the minimum DC level. In this way, the image will be smoother. Furthermore, the authors rearranged the periods in such a way that make the image smoother by putting the minimum variant period on the top of the image then compare other periods with this selected period using mean squared error, and arrange the most similar periods near to the top.

Hsiao-Hsuan Chou [15] proposed a new lossy compression algorithm for irregular ECG signals by converting the ECG signal to a 2D image. The conversion process starts with QRS detection stage and the period sorting technique is then applied. Lastly, period length equalization using mean technique is applied and the resultant 2D image is compressed using Jpeg2000 algorithm.

Finally, A.Khalaj and H. Miar Naimi [41] proposed a new ECG lossy compression technique based on fractal model and IFS. In their model the ECG signal is divided into non-overlapping blocks called range blocks. Each block is transformed into its most similar domain block using fractal transformation parameters. In their method they used one fractal transform parameter and the index of the most similar domain block is stored with the fractal parameter. Moreover, the block is rotated and the range block mean value is stored as well. This work is somewhat similar to our proposed model. However, in their work they used only one fractal transform parameter while in our proposed method two fractal transform parameters (scale and shift parameters) are used. Therefore, the results obtained are more accurate with low distortion and higher compression ratio.

Most of the above mentioned techniques rely on detecting QRS complex which is by itself a challenge in case of abnormal ECG signals. Therefore, those techniques will not perform well in case of abnormal ECG signal.

3.3 Fractals and ECG signals

In this chapter, new fractal based lossy compression technique is proposed. Fractal can be defined as an object which consists of smaller structures which are similar to the original object. Generally, there are several definitions of fractal that reflect the following features [6]:

1. Fractals are fine structured objects and details can be seen for any scales.
2. Fractal provides a geometrical descriptions for irregular objects that can not be described by other mathematical formulas.
3. Fractal reflects some self similarity characteristics.

The self similarity property of fractal is utilized in the proposed ECG compression technique. Accordingly, the proposed technique uses fractal self similarity and iterated function system together in order to capture accurate ECG features .

3.3.1 Iterated Function System (IFS)

A fractal object can be generated from a recursive process of different transformations applied on a part of the object. Let $W_1 W_2 W_3 ... W_N$ be different transforms and N is the total number of transforms. General fractal Iterated Function can be defined as in Eq. 3.1

$$A = W_1(A) \cup W_2(A) \cup W_3(A) \cup .... \cup W_N(A) \tag{3.1}$$

where A is the fractal object. Accordingly, the fractal object consists of smaller, similar objects transformed and assembled to form the original object. However, it is not required for the fractal object to be entirely similar to a smaller part of itself. A small part of the fractal object(normally referred to as range) can be similar to another part (normally called domain) after applying a specific transform.

The basic idea of fractal compression using IFS is to divide the original ECG signal into non overlapping blocks called range blocks. Each range block is then compared with all other overlapping blocks (called domain blocks) after applying a specific transform as shown in Fig. 3.3. The transform parameters (normally called fractal coefficients) and the domain block number are stored instead of the original ECG block in the output file. As a result, the compressed version contains only the transform parameters and indexes. Technically, it is required to use a distortion measurement metric to select the most similar domain block to the current range block. Finally, fractal coefficients (shift and scale) and indexes(block location) are stored to be quantized in the output file.

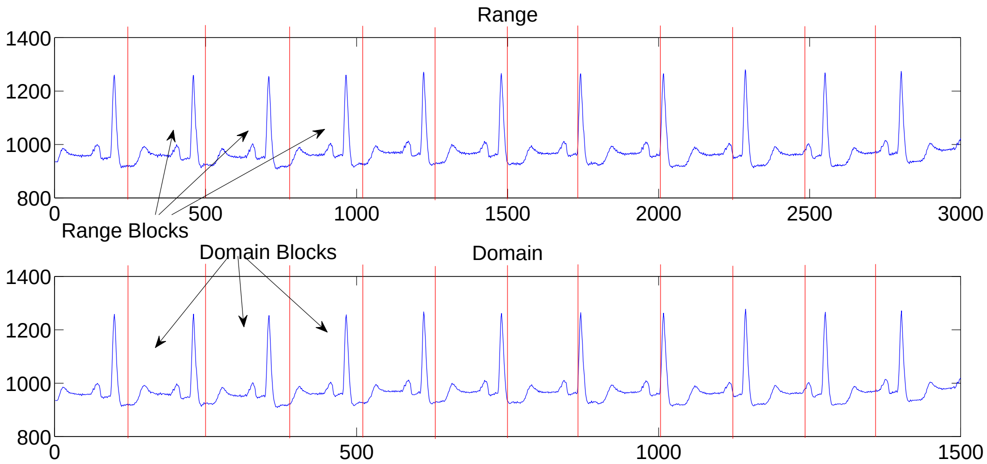


Figure 3.3: Range and Domain blocks in the original and down sampled ECG signals. Rang blocks are the non-overlapped blocks of the original ECG signal, the domain blocks are the ones of the down-sampled search domain.

3.3.2 Fractal Coefficients

Shift and Scale are the basic fractal coefficients used to generate the geometrical description of objects. The Scale (S) fractal coefficient should be found to make the difference among ECG instances in the range block matches the difference among ECG instances in the domain block. Scale coefficient can be defined as the greatest difference among ECG instances in the range block divided by the greatest difference among the ECG instances in the domain block. On the other hand, Shift fractal coefficient (O) should be found to make the ECG instances in the range block similar to the ECG instances of the domain block. Moreover, shift fractal coefficient can be defined as the difference between the average ECG instance value in the range block and the average ECG instance value in the domain block. Equation 3.2 shows how the scale and shift fractal coefficients are used to generate range block(R) from domain block (D) [6].

$$R \approx S \times D + O \tag{3.2}$$

where R represents the range block, D represents the Domain block , S and O are the Scale and shift fractal coefficients respectively. Fractal coefficients [6] can be calculated as shown in Eq. 3.3 and Eq. 3.4

$$S = \frac{n\sum_{i=1}^{n} D(i)R(i) - \sum_{i=1}^{n} D(i)\sum_{i=1}^{n} R(i)}{\sum_{i=1}^{n} D(i)^2 - \left(\sum_{i=1}^{n} D(i)\right)^2} \tag{3.3}$$

and

$$O = \frac{1}{n}\left(\sum_{i=1}^{n} R(i) - S\sum_{i=1}^{n} D(i)\right) \tag{3.4}$$

where n is the number of ECG points in one block, $R(i)$ is the ith ECG point in the range ECG block. $D(i)$ is the ith ECG point in the domain ECG block, S is the scale fractal coefficient and O is the shift (offset) fractal coefficient.

Generally these parameters are calculated for all domain blocks for each range block in one iteration. Finally, it is required to compare all the resultant transformed domain blocks with a range block in each iteration to select the most similar domain block.

To increase the performance of the proposed algorithm, some parts of the fractal equation do not need to be computed for different domain blocks. However, it should be computed only once per range block. These parts are summarized as follows:

1. $\sum_{i=1}^{n} R(i)$ the total summation of all the ECG points inside the range block is calculated once for each range.

2. $\sum_{i=1}^{n} R(i)^2$ the total summation of the squares of the ECG points in the specified range block.

3.3.3 Affine Transform

In addition to the scale and shift fractal coefficients, affine transforms [87] are applied to guarantee that maximum similarity can be achieved between the range and domain blocks. Affine transforms are geometric transforms such as mirror, reflection and rotation that can be mathematically represented in Eq. 3.5

$$\overline{D} = T \times D \tag{3.5}$$

where $\overline{D}$ is the affine transformed domain block, D is the original domain block and T is the affine transform matrix. As the ECG is a one dimensional signal, the number of affine transforms that can be used is limited. Four Affine transforms are used in the proposed method (see Table 3.1

Table 3.1: Affine transforms used in the proposed ECG fractal compression technique

encoding no	**transform**
0	No Change
1	Reflection
2	replacing fist half with second half
3	replacing each adjacent pair together

3.3.4 Comparison of range and domain blocks using fractal RMS

Fractal Root Mean Square (RMS) measurement is the basic metric of fractal matching process to find the similar domain block. Equation 3.6 is required to compute RMS for range (R) and domain (D) blocks.

$$RMS = \sqrt{\frac{1}{n}\left[\sum_{i=1}^{n} R(i)^2 + S\left(S\sum_{i=1}^{n} D(i)^2 - S\sum_{i=1}^{n} D(i)R(i) + 2O\sum_{i=1}^{n} D(i)\right) + O\left(nO - 2\sum_{i=1}^{n} R(i)\right)\right]} \tag{3.6}$$

It is called fractal RMS because as shown from the equation it is different than the normal RMS. The original RMS equation has been modified to reflect the fractal transforms applied on the original domain block (the scale and shift transforms).

3.4 Cloud-enabled fractal compression technique

3.4.1 Compression Algorithm

The proposed compression technique can be divided into three parts as shown in the block diagram in Fig. 3.4. Firstly, the ECG signal is divided into equal sized non overlapping blocks called Range Blocks. Then, another copy of the ECG signal is created to be the search domain. The search domain is initially down sampled as shown in Eq. 3.7

$$D_i = \frac{P_{2i} + P_{2i+1}}{2} \quad i = 0, 1, 2...n \tag{3.7}$$

where D_i is the i-th down sampled domain instance, P_{2i} and P_{2i+1} two consecutive points. The domain search is divided into overlapping blocks called domain blocks. The distance between two consecutive domain blocks can be determined using the jump step parameter. Then, for each range block, fractal RMS is used to determine the most similar domain block.

Algorithm 2 is required to perform the fractal ECG compression process. The algorithm starts by initializing the ECG signal and the ECG domain down sampled version. Then, the algorithm iterates for each range block and finds the most similar domain block. Finally, it stores the corresponding shift, scale, affine and domain block index in the compressed file.

Algorithm 3 is used to find the similar domain block based on the fractal RMS measurements. The algorithm inputs are the range block and the down sampled ECG signal(search domain). For the given range block the algorithm iterates through the whole search domain and calculates the fractal coefficients as well as fractal RMS between the range block and the current domain block. Next, if the calculated RMS value is less than the previous stored value, the algorithm will store the calculated fractal coefficients as well as the current domain block index in the output parameters and it will update the stored RMS value. At the end of the loop, the fractal coefficients that belong to the domain block which had minimum RMS value are returned. Affine transforms are implemented in the fractalrms algorithm in order to increase the possibility of having better similarity with domain blocks.

The final compressed file consists of several rows, one row for each range block, each row contains the following information:

1. The scale value
2. The Offset Value
3. The domain block index
4. Affine transform code as shown in Table 3.1

Algorithm 2 ECG fractal Compression algorithm

```
INPUTS:
R is the original ECG signal
BS represents block size
OUTPUTS: o,s,i,a
o ← 0 Array to store offset coefficients for all the range blocks
s ← 0 Array to store the scale coefficients for all the range blocks
i ← 0 Array to store the indexes for the selected domain blocks
a ← 0 Array to store the encoding of the affine transform applied for each range block
rm ← 0 Will store the RMS values
D is the down sampled domain created from R as shown in Eq. 3.7
z ← 0
Js represnts the jump step of the search loop in the domain
R_i Represents the Range block with size of BS
for each range block R_i in R do
    [sc, of, in, aff] = fractalrms(R_i, D, js)
    s(z) ← sc
    o(z) ← of
    i(z) ← in
    a(z) ← aff
    z ← z + 1
end for
RETURN o,s,i,a
```

Algorithm 3 fractalrms function algorithm

INPUTS:
r=the range ECG block
d=the down sampled domain
Js represents the jump step
OUTPUTS: so,oo,io,aff0
$so = 0$; {The determined scale value for the selected domain block}
$oo = 0$; {The determined shift value for the selected domain block}
$io = 0$; {The determined location for the most similar domain block to the specified range block}
$aff0 = 0$; {The determined affine transform encoding value for the selected domain block}
$rms0 = 0$;
blocksize=size(r,1); {stores the block size}
sumr=sum(r); {$sumr = \sum R(i)$ the total sum of the range block}
sumr2=0;
for j=1 to blocksize **do** {{$\sum_{j=1}^{blocksize} R(j)^2$}}
    $sumr2 = sumr2 + (r(j)^2)$;]
end for
rms0=10000; {initialize rms0 with large value}
sd=size(d,1); {Size of the down sampled ECG domain}
for i=1 to sd-blocksize step js **do** {{iterate through the whole search domain and cut domain blocks with size of sd}}
    t1=d(i:i+blocksize-1);
    for a=0 to 3 **do**
        t=affine(t1,a); {calculate the affine transform and return the transformed domain block in t}
        sumrt=0; {calculate $\sum_{j=1}^{blocksize} R(j)t(j)$}
        $sumr2 = sumr2 + (r(j)^2)$;
        for j=1 TO blocksize **do**
            sumrt=sumrt+(r(j)*t(j));
        end for
        sumt=sum(t); {calculate domain block coefficnats such as $\sum t$ and $\sum t^2$}
        sumt2=0;
        for j=1 to blocksize **do**
            $sumt2 = sumt2 + (t(j)^2)$;
        end for

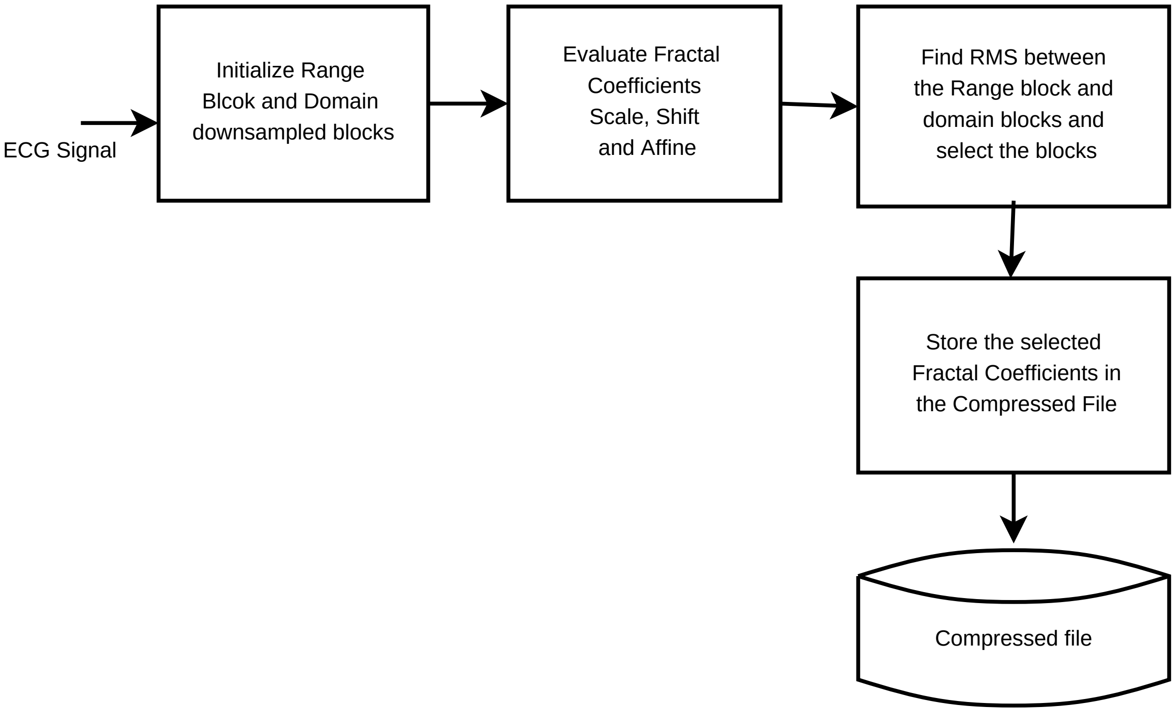


Figure 3.4: Fractal Compression Block Diagram

$s = ((size(r,1) * sumrt) - (sumr * sumt))/(size(r,1) * sumt2 - sumt^2)$; {calculate the scale fractal coefficient as shown in Eq. 3.3. Where size(r,1) get the number of rows in r}
$o = (1/size(r,1)) * (sumr - s * sumt)$; {calculate the shift fractal coefficient as shown in Eq. 3.4}
$rmsn = sqrt((1/size(r,1)) * (sumr2 + s * (s * sumt2 - 2 * sumrt + 2 * o * sumt) + o * (size(r,1) * o - 2 * sumr)))$; {calculate the fractal rms as shown in Eq. 3.6}
if $(rmsn \leq rms0)$ **then**
so=s;
oo=o;
io=i;
aff0=a;
end if
end for
end for
RETURN so,oo,io,aff0

The proposed fractal compression algorithm can be used in the cloud and make use of its multiprocessing capabilities as shown in Fig. 3.5. Firstly, the down-sampled domain is created from the original ECG segments and copied onto each processing unit to be used as the search domain. Then the ECG segment is divided into sub-segments and copied to each processing unit, one sub-segment per unit. Each unit will process its sub-segment and do the fractal compression by calculating the fractal coefficients. Finally, all fractal coefficients from all processing units are reassembled into one final compressed file. In this scenario the fractal compression algorithm is running in all the processing unit in parallel. Therefore, the proposed fractal compression algorithm is a cloud efficient algorithm capable of taking advantages of the vast computing resources available on cloud.

The block size parameter is stored inside the compressed file in the header part. It is linearly related to the compression ratio. Therefore, the compression ratio can be predetermined by selecting the appropriate block size as shown in Fig. 3.11. Moreover, the distortion effect cannot be predetermined by selecting the block size parameter. However, whenever the block size is increased the distortion effect will be increased as well.

3.4.2 Decompression algorithm

The decompression process starts by generating a random signal having the same length as the original ECG. The original ECG is approximated from the random signal after applying the buffered fractal coefficients. Domain blocks are required to approximate the ECG. Therefore, the random signal is used as range blocks down sampled to create the domain blocks. Technically, each block in the approximated ECG is calculated by re-transforming

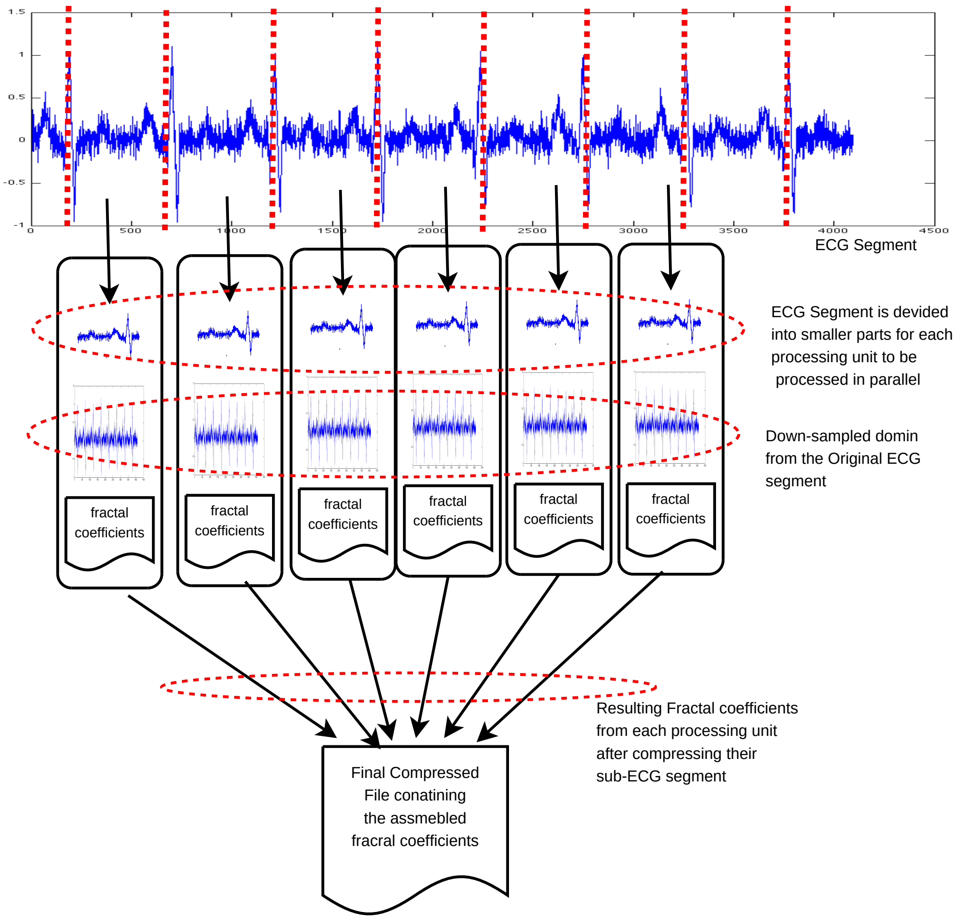


Figure 3.5: Cloud implementation of the proposed fractal compression algorithm

the selected domain block using the buffered fractal coefficients and affine transform code. Equation 3.8 is required to approximate a single block in the generated ECG.

$$\widehat{R}_i = S_i * D_{index(i)} + O_i \tag{3.8}$$

where $\widehat{R}$ represents the ith approximated block in the generated ECG. S and O are the scale and shift fractal coefficients respectively. $Index(i)$ represents the index of the domain block. Finally, all the constructed range blocks are assembled together to approximate the decompressed ECG signal. The decompression process is repeated using the new generated ECG (as input) four times in order to produce an accurate approximated ECG signal (see Algorithm 4).

Algorithm 4 ECG fractal DeCompression algorithm

```
INPUTS:
BS represents the block size
O Array of stored offset coefficients for all the range blocks
S Array of stored scale coefficients for all the range blocks
I Array of stored indexes for the selected domain blocks
A Array of stored encoding of the affine transform applied for each range block
OUTPUTS: ECG
ECG is the constructed ECG signal
for iter=1 TO 4  do {{repeat 4 iterations}}
   D is the down sampled domain created from ECG as shown in Eq. 3.7
   for each element o in O, s in S, a in A, i in I do
      d1 ← (ith Domain block with size of BS)
      d2 ← affine(d1, a)
      R ← s × d2 + o
      ECG = ECG ⋃ R
   end for
end for
RETURN ECG
```

As shown it will be possible to generate the ECG signal starting from any point in the compressed file as long as the required fractal coefficients are available as shown in Fig. 3.6. Therefore, this is a significant feature of the proposed fractal compression technique. However, currently available ECG compression techniques can achieve this by dividing an ECG signal into smaller segments, compressing each segment separately then storing it in a separate file. In this case, headers will be added to each compressed file which will create more overhead. On the other hand, the proposed fractal technique does not suffer from this limitation because there are no headers to be added and the compressed file is just a stream of rows that contains fractal coefficients and indexes. Moreover, the user (such as doctor) can specify which part he/she want to decompress (for example at specified time period), and, cloud service will send the required part only in its compressed format. Finally, the receiver can decompress it without having the full ECG compressed file. This is a significant feature as the user can retrieve part of the ECG signal without waiting for the full compressed ECG data to be downloaded.

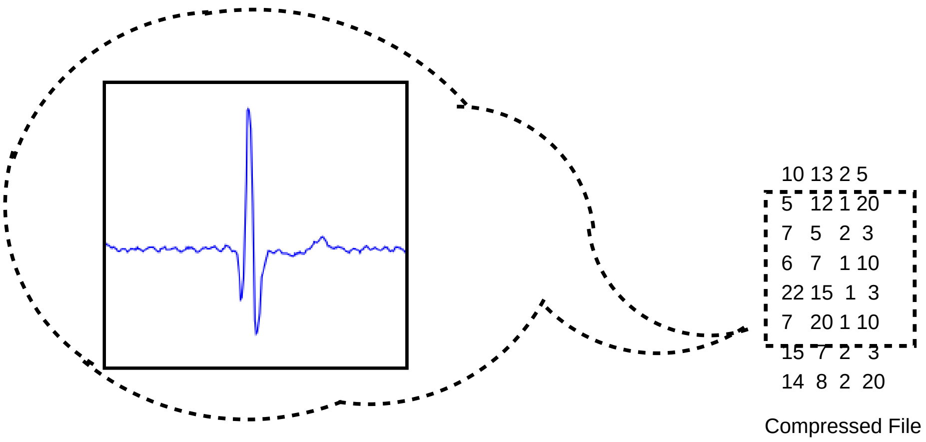


Figure 3.6: Partial decompression operation. The decompression operation is capable of processing the compressed file starting from any position in the file and not just from the beginning of the file.

3.5 Fast Fractal Model

After introducing the normal fractal compression technique proposed in previous sections, it is found that its performance is poor because, for each range block, the entire search domain should be examined to find the most similar domain block. Furthermore, this operation is repeated for each range block. To increase the compression speed it is required to limit the search domain to a small number of domain blocks rather than searching all domain blocks. To achieve this objective, a new fractal model derivation is proposed.

Let R be the range block then R_i is the i^{th} instance inside the range block. Similarly D is the domain block and D_i is the i^{th} instance inside the domain block. Moreover, let $\overline{R}, \overline{D}$ be the average value of the range block and domain block respectively. The relation between these values can be shown in Eq. 3.9, Eq. 3.10 and Eq. 3.11

$$R = S \times D + O \tag{3.9}$$

$$R_i = S \times D_i + O \tag{3.10}$$

$$\overline{R} = S \times \overline{D} + O \tag{3.11}$$

Next, two new factors will be introduced α and δ as shown in Eq. 3.12 and Eq. 3.13.

$$\alpha = \sum \left| R_i - \overline{R} \right| \tag{3.12}$$

$$\delta = \sum \left| R_i - \overline{R} \right|^3 \tag{3.13}$$

Now if Eq. 3.10 and Eq. 3.11 are substituted in Eq. 3.12 then

$$\alpha = \sum \left| S \times D_i + O - S \times \overline{D} - O \right| \tag{3.14}$$

Next Eq. 3.14 will be simplified as shown in Eq. 3.15

$$\alpha = |S| \times \sum \left|D_i - \overline{D}\right| \tag{3.15}$$

Substitute Eq. 3.10 and Eq. 3.11 in Eq. 3.13 we get

$$\delta = \sum \left|S \times D_i + O - S \times \overline{D} - O\right|^3 \tag{3.16}$$

After simplifying Eq. 3.16 we will get.

$$\delta = |S|^3 \times \sum \left|D_i - \overline{D}\right|^3 \tag{3.17}$$

Now let us set the fast fractal factor as shown in Eq. 3.18

$$F1 = \frac{\alpha^3}{\delta} \tag{3.18}$$

To write the fast fractal factor $F1$ in terms of range blocks. Equation 3.12 and Eq. 3.13 are substituted in Eq. 3.18 to get

$$F1 = \frac{\left(\sum \left|R_i - \overline{R}\right|\right)^3}{\sum \left|R_i - \overline{R}\right|^3} \tag{3.19}$$

Similarly, the fast fractal factor $F1$ can be written in terms of domain blocks by substituting Eq. 3.15 and Eq. 3.17 in Eq. 3.18 as shown in Eq. 3.20

$$F1 = \frac{S^3 \times \left(\sum \left|D_i - \overline{D}\right|\right)^3}{S^3 \times \sum \left|D_i - \overline{D}\right|^3} \tag{3.20}$$

Finally, Eq. 3.20 can be simplified as shown in Eq. 3.21

$$F1 = \frac{\left(\sum \left|D_i - \overline{D}\right|\right)^3}{\sum \left|D_i - \overline{D}\right|^3} \tag{3.21}$$

It is proven that the calculated fast fractal factor should be equal if it is calculated based on the range block or from its transformed domain block. Therefore, the new modified fast fractal model is based on calculating these fast fractal factors in advance for all range blocks as well as all domain blocks. Then, for each range block instead of searching the whole domain blocks, it is only required to search in the domain blocks with similar fast fractal factor to the specified range block. As a result, the RMS fractal and the fractal coefficients will be calculated only for these selected domain blocks. Finally, the most similar domain block to the range block will be selected from this subset of domain blocks.

3.5.1 Fast fractal compression algorithm

The fast fractal compression technique consists of several stages as shown in Fig. 3.7. The proposed technique starts by creating another copy of the original ECG signal called domain search signal. Domain search signal is down sampled as shown in Eq. 3.22.

$$\overline{D_i} = \frac{D_{2i} + D_{2i+1}}{2} \quad i = 0, 1, 2...n/2 \tag{3.22}$$

where $\overline{D_i}$ is the i-th down sampled domain instance, D_{2i} and D_{2i+1} two consecutive points. The down sampled domain search signal is divided into overlapping blocks separated by distance called jump step. On the other hand, the original ECG signal is divided into non overlapping blocks called range blocks. Both range and domain blocks have the same size.

The next step in the compression algorithm is to calculate the fast fractal factor $F1$ for all range blocks ($F1_r$) as well as all domain blocks ($F1_d$). Next, a mapping process is applied to map range blocks with the domain blocks that have similar fast fractal factors and store their indexes.

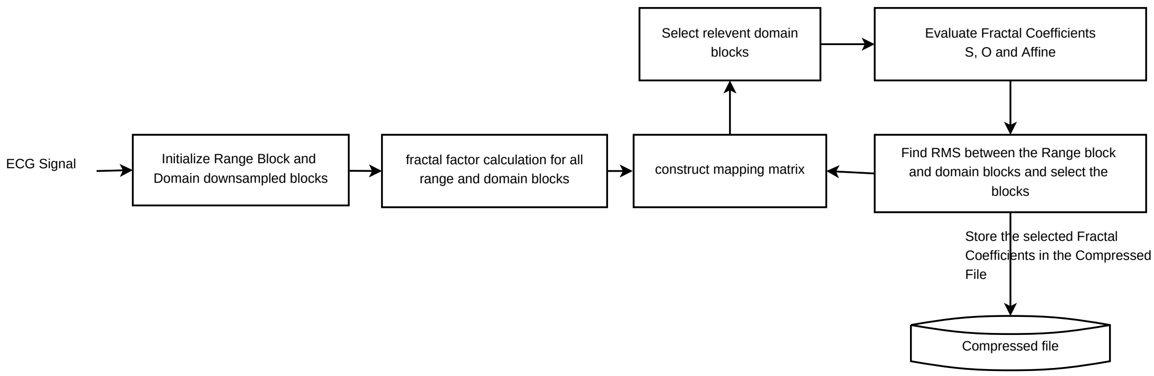


Figure 3.7: Fractal Compression Block Diagram

The mapping process begins with rescaling the fractal factor values to convert its values with in the range from 1 to 100. Then, the fractal factor array $F1_d$ is sorted in ascending order as shown in Fig. 3.8. Next, mapping matrix is constructed based on the sorted fractal factor array. For example if the range block fractal factor is 5, then, according to Fig. 3.8 the search operation will occur on the domain blocks number 2 and 4. Therefore, the most similar domain block from these two blocks to the specified range block will be selected. Finally, the fractal coefficients of the selected domain block will be stored inside the compressed file as well as the domain block index.

The proposed fast fractal compression algorithm steps can be summarized as shown in Algorithm 5. The algorithm starts by initializing some variables such as ECG signal range

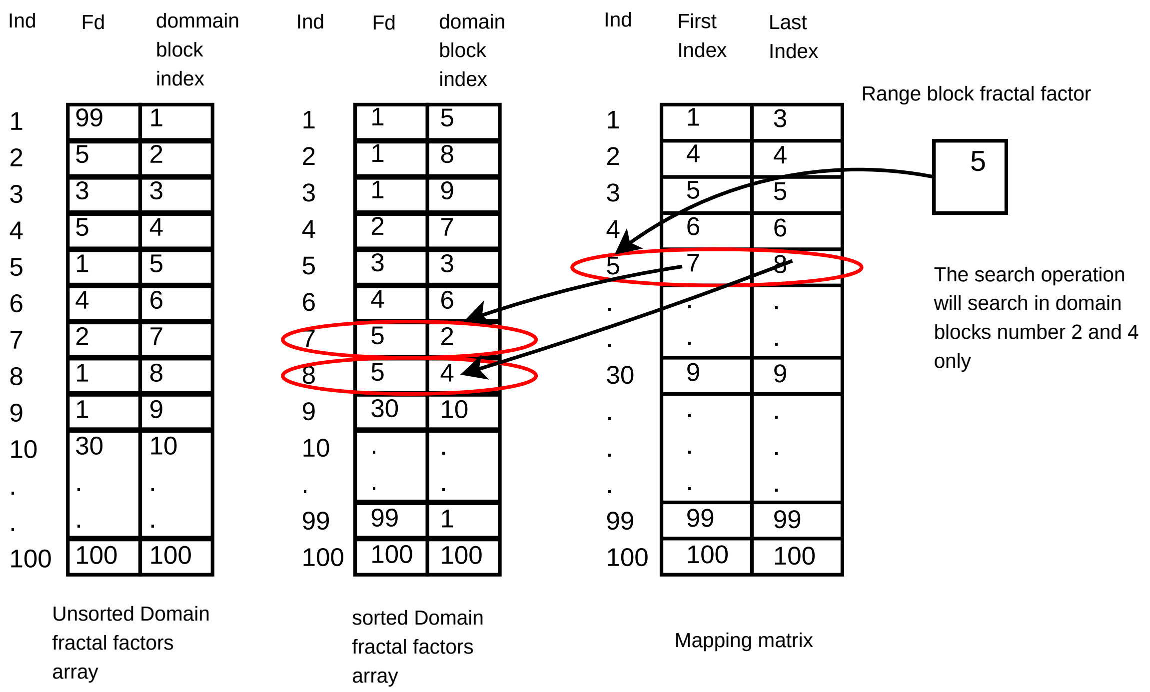


Figure 3.8: Mapping Process

blocks, ECG down sampled domain and fractal coefficients arrays. Next, it calculates the fractal factors for all range blocks and constructs the FR array. Similarly, the algorithms calculates the fractal factors for all domain blocks and stores them inside FD array. Next, the fractal factors are rescaled to fail within the range from 1 to 100. Then, mapping processes starts by sorting FD array and then constructing the mapping matrix which is a 100×2 matrix. Then, the algorithm iterates for each range block and find the starting and ending domain search limits. Consequently, it will find the most similar domain block in this specified search domain blocks. Finally, it will store the corresponding shift , scale, affine and domain block index inside the compressed file.

Fractal coefficients and similarity measurement between range block and domain blocks is performed using Algorithm 6. Its inputs are the range block and the down sampled ECG

Algorithm 5 ECG fractal Compression algorithm

$o \leftarrow 0$ Array to store offset coefficients for all the range blocks
$s \leftarrow 0$ Array to store the scale coefficients for all the range blocks
$i \leftarrow 0$ Array to store the indexes for the selected domain blocks
$rm \leftarrow 0$ Will store the RMS values
$a \leftarrow 0$ Array to store the encoding of the affine transform applied for each range block
R is the original ECG signal
D is the down sampled domain created from R as shown in Eq. 3.22
$z \leftarrow 0$
Js represnts the jump step of the search loop in the domain
{Calculate fast fractal factor for all range blocks}
for each range block R_x in R **do**
  calculate α as in Eq. 3.12
  calculate δ as in Eq. 3.13
  $FR_x \leftarrow \alpha^3/\delta$
end for
{Calculate fast fractal factor for all domain blocks}
for each domain block D_x in D **do**
  Calculate FD_x as shown in Eq. 3.21
end for
{The following steps are for rescaling the fractal factor arrays with in the range from 1 to 100}
$Frd \leftarrow arrayconcat(FR, FD)$ {frd is a matrix that its columns are the fractal factors arrays}
$maxf \leftarrow max(Frd)$ {find the maximum value in the fractal factor arrays}
$minf \leftarrow min(Frd)$ {find the minimum value in the fractal factor arrays}
$FR \leftarrow int(((FR - minf)/(maxf - minf)) * 100)$
$FD \leftarrow int(((FD - minf)/(maxf - minf)) * 100)$
{mapping process example shown in Fig. 3.8}
$FDI \leftarrow FD$
$FDI(:, 2) \leftarrow [1, 2, , n]$ {add second column to FDI to store domain block indexes initialized from 1 to n, where n is number of domain blocks}
$FDI \leftarrow sortrow(FDI)$ {sort FDI matrix based on the fast fractal factor i.e column 1}
$mp \leftarrow createmapping(FDI, n)$ {create the mapping matrix m(100,2) as illustrated in Fig. 3.8}

for each range block R_i in R **do** {i represents current range block number ff and fl represent pointers to the first and last domain blocks that will be used in the search process}
  $ff = mp(FR(i) + 1, 1)$
  $fl = mp(FR(i) + 1, 2)$
  $[sc, of, in, rms, aff] = rmsfract(R_i, D, js, FDI, ff, fl)$
  $s(z) \leftarrow sc$
  $o(z) \leftarrow of$
  $i(z) \leftarrow in$
  $a(z) \leftarrow aff$
  $z \leftarrow z + 1$
end for

signal (search domain) as well as the first and last pointers to the domain search limits. The algorithm will compare the specified range block with the domain blocks in the specified search domain. Accordingly, fractal coefficients and fractal RMS will be calculated. As a result, the most similar domain block will be determined and its fractal coefficients as well as its index and affine transform code will be stored in the compressed file.

The main difference between the normal fractal compression and fast fractal compression is the number of comparative operations and fractal coefficients calculations. In normal fractal and for one range block these operations are repeated for all domain blocks, while in fast fractal, small number of domain blocks will be searched. Therefore, these operations are repeated for small number of domain blocks.

The resultant file should contain the following information about the most similar domain block for each range block.

1. The fractal coefficients for the selected domain block (scale and shift)

2. The domain block index

Algorithm 6 fractalrms function algorithm

$so = 0$; {The determined scale value for the selected domain block}

$oo = 0$; {The determined shift value for the selected domain block}

$io = 0$; {The determined location for the most similar domain block to the specified range block}

$rms0 = 0$;

$aff0 = 0$; {The determined affine transform encoding value for the selected domain block}

r=the range ECG block

d=the down sampled domain

blocksize=size(r,1); {stores the block size}

ff represents a pointer to a position in the FDI array which represents the starting search domain block

fl represents a pointer to a position in the FDI array which represents the last search domain block

sumr=sum(r); {$sumr = \sum R(i)$ the total sum of the range block}

sumr2=0;

Js represents the jump step

for j=1 to blocksize **do** {{$\sum_{j=1}^{blocksize} R(j)^2$}}

  $sumr2 = sumr2 + (r(j)^2)$;]

end for

rms0=10000; {initialize rms0 with large value}

sd=size(d,1); {Size of the down sampled ECG domain}

for i=ff to fl **do** {{iterate through the specified search domain and cut domain blocks with size of sd}}

  $ix1 = FDI(i, 2) * js$

  $ix2 = ix1 + blocksize - 1$

  $t1 = d(ix1 : ix2)$

  for a=0 to 3 **do**

    t=affine(t1,a); {calculate the affine transform and return the transformed domain block in t}

    sumrt=0; {calculate $\sum_{j=1}^{blocksize} R(j)t(j)$}

    $sumr2 = sumr2 + (r(j)^2)$;

    for j=1 TO blocksize **do**

      sumrt=sumrt+(r(j)*t(j));

    end for

sumt=sum(t); {calculate domain block coefficnats such as $\sum t$ and $\sum t^2$}
sumt2=0;
for j=1 to blocksize **do**
$sumt2 = sumt2 + (t(j)^2)$;
end for
$s = ((size(r, 1) * sumrt) - (sumr * sumt))/(size(r, 1) * sumt2 - sumt^2)$; {calculate the scale fractal coefficient as shown in Eq. 3.3}
$o = (1/size(r, 1)) * (sumr - s * sumt)$; {calculate the shift fractal coefficient as shown in Eq. 3.4}
$rmsn = sqrt((1/size(r, 1)) * (sumr2 + s * (s * sumt2 - 2 * sumrt + 2 * o * sumt) + o * (size(r, 1) * o - 2 * sumr)))$; {calculate the fractal rms as shown in Eq. 3.6}
if $(rmsn \leq rms0)$ **then**
so=s;
oo=o;
io=i;
aff0=a;
end if
end for
end for

3. Affine transform code as shown in Table 3.1

3.5.2 Decompression algorithm

The decompression operation of the fast fractal compression is exactly identical to the decompression operation of the normal fractal discussed in Section 3.4.2. The decompression operation does not require to use the fractal factors calculated in the compression phase, therefore, all fractal factors can be safely deleted after the compression operation is completed. To decompress the resultant compressed file, several operations are performed. Firstly, a random ECG signal is generated which is similar to the original ECG signal in length. Next, this random ECG signal will be down sampled and used as the search domain. Using the information stored in the compressed file and Eq. 3.23 range blocks are reconstructed.

$$\widehat{R}_i = S_i * D_{index(i)} + O_i \tag{3.23}$$

where $\widehat{R}$ is the ith approximated range block, S and O are the fractal coefficients and $Index(i)$ is the index of the domain block. The reconstructed range blocks can be assembled to represent the resultant approximated ECG signal, Finally, the resultant signal will be again used as an input instead of the random signal generated at the start. This process can be repeated several times to guarantee maximum accuracy for the decompressed ECG signal (see Algorithm 4).

3.6 Evaluation strategy

The efficiency of the proposed lossy fractal compression technique can be measured based on two measurements. Compression ratio is the first measurement that reflects the ratio between the original to the compressed ECG size. Compression ratio can be calculated as shown in Eq. 3.24

$$CompressionRatio = \frac{Original\ ECG\ file\ size}{Compressed\ ECG\ file\ size} \tag{3.24}$$

The second measurement used to calculate the efficiency of the lossy compression technique is the distortion measurement between the original ECG signal and the reconstructed ECG signal. In this chapter, Percentage Residual Difference (PRD) error measurement is applied. To calculate the PRD between the original ECG signal and the reconstructed ECG signal, Eq. 3.25 can be used.

$$PRD = \sqrt{\frac{(\sum_{i=1}^{N}(X_i - \tilde{X}_i)^2)}{\sum_{i=1}^{N}(X_i^2)}} \tag{3.25}$$

where X_i represents the original ECG signal and $\tilde{X}_i$ represents the decompressed ECG signal. High efficiency can be obtained by producing high compression ratio with low PRD value.

3.7 Experimental Results

3.7.1 Cloud-enabled fractal compression experiments and results

The proposed lossy algorithm is tested using MIT-BIH Arrhythmia Database [1]. This database contains 47 ECG record from different patients. Each ECG record is 30 minutes length. The proposed model performance can be affected in several parameters. Therefore, the experiments are designed to show the effect of changing each parameter on the compression ratio and the PRD. The variable parameters can be summarized as follows:

1. Range block size. This parameter has a direct effect on the compression ratio as well as the accuracy of the decompressed ECG signal.
2. Jump step. It represents the distance between consecutive domain blocks. Incrementing the jump step causes a decrease in the accuracy and execution time.

Several block sizes are applied. Compression ratios and PRD values are investigated. Ten jump steps (1-10) are applied for all experiments (see Table 3.2). It is clear that increasing the block size produces a higher compression ratio. However, it leads to higher PRD values.

At the same time, PRD value is affected by the jump step as bigger jump steps would result in higher PRD. Figure 3.9 shows the relation among the compression ratio, PRD, and the jump steps. Moreover, Fig. 3.11 shows the linear relationship between the block size parameter and the compression ratio. The decompressed ECG files can be shown in Fig. 3.10 for different block sizes. The large similarity between the original signal and the decompressed ECG signal is clear even with large block sizes.

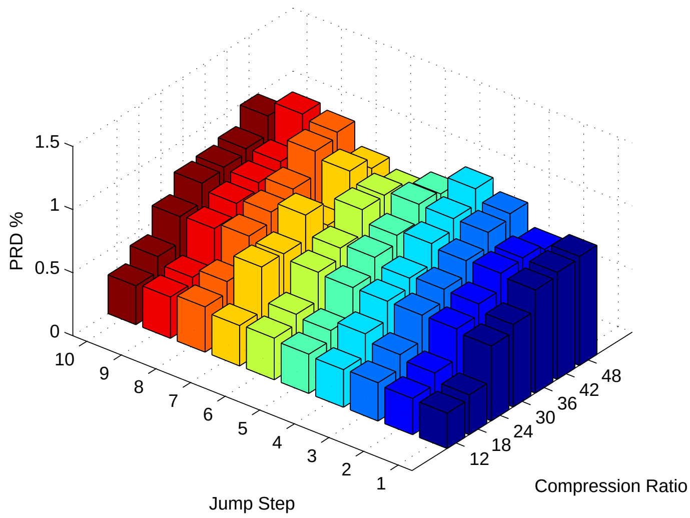


Figure 3.9: Relation between Compression ratio and PRD for different Jump Steps

To show the performance of the proposed algorithm ECG signals of 60 second length have been tested and compression time is measured for different block sizes and jump steps as shown in Table 3.4 using parallel processing mode and 8 cores processor. The proposed algorithm can achieve highest compression ratio of 42 with PRD value less then 1% and compression time of 3.2 seconds for 60 seconds ECG segment if the block size used is 35 and jump step is 10. However, the experiments are executed using MATLAB and Windows platform running on desktop computer. Moreover, additional experiments are performed

Table 3.2: PRD and compression ratio relation for different jump steps. The CR obtained is 40 with PRD of 1%

Bs	Cr	PRD									
		Jump step									
		1	2	3	4	5	6	7	8	9	10
10	12	0.279	0.286	0.303	0.297	0.310	0.329	0.319	0.357	0.327	0.307
15	18	0.314	0.379	0.415	0.471	0.388	0.407	0.673	0.449	0.373	0.429
20	24	0.593	0.619	0.619	0.619	0.619	0.632	0.668	0.699	0.650	0.636
25	30	0.652	0.707	0.714	0.693	0.749	0.713	0.866	0.781	0.740	0.791
30	36	0.817	0.839	0.830	0.857	0.813	0.907	0.675	0.852	0.794	0.808
35	**42**	**0.846**	**0.852**	**0.945**	**0.945**	**0.950**	**0.912**	**0.995**	**1.043**	**0.851**	**0.845**
40	48	0.863	0.809	0.984	1.07	0.922	0.834	0.906	1.084	1.119	0.9948

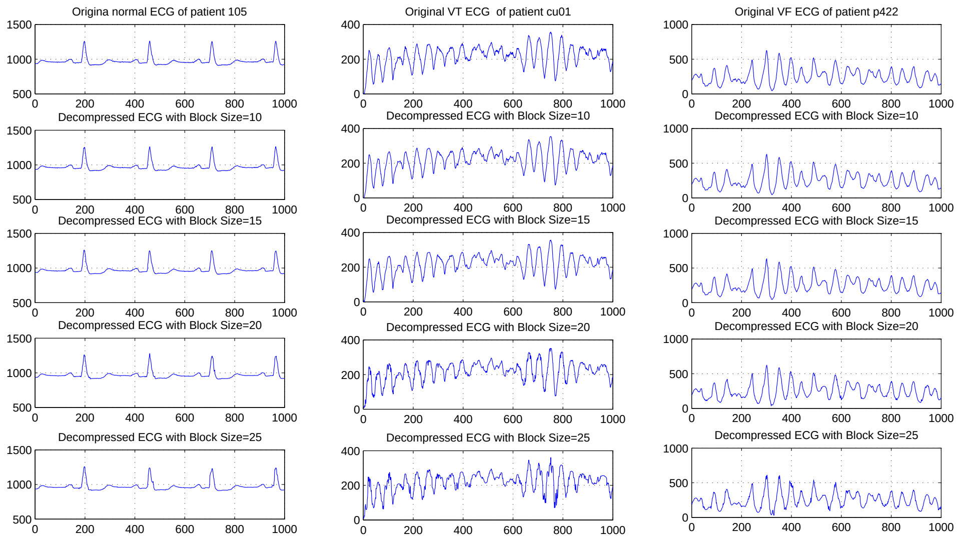


Figure 3.10: Decompressed ECG signals from different patients with different physiological conditions as well as with different block size

to compare the compression time of the proposed technique using single core(i.e serial processing) and 2,4 or 8 cores parallel processing approach to simulate cloud scenario as shown in Table 3.3. It is clear from this table how the compression time is decreased if the number of processing units is increased. As a result, in cloud environment the execution time can be 20 times faster [18; 69; 34; 36] using a cloud service of 128 processors. Accordingly,

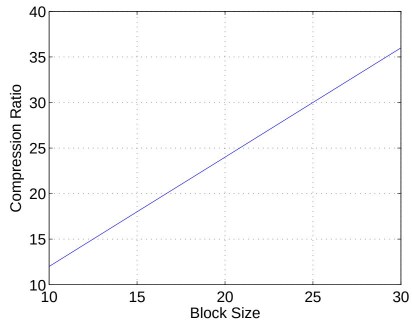


Figure 3.11: Block size and Compression ratio

the cloud execution time for the worst case of Block Size=10 and Jump step=1 will be 33 seconds, while, the cloud execution time for the case of Block size=40 and Jump step=10 is 1.1 seconds. Finally, Table 3.5 shows the decompression time for different block sizes.

Table 3.3: Comparison of the proposed algorithm compression time using single core and multi-core processing on a single machine

block size	Jump Step = 10				Jump step = 1			
	1 core	2 cores	4 cores	8 cores	1 core	2 cores	4 cores	8 cores
10	65.77	27.01	17.33	13.83	676.66	224.64	142.28	108.89
15	64.06	24.56	12.48	8.48	448.24	148.37	92.44	67.93
20	30.43	21.53	7.68	5.49	360.84	116.23	78.59	52.80
25	31.14	20.75	6.21	4.26	279.71	90.83	62.60	44.73
30	24.09	12.80	5.25	6.53	258.46	81.06	55.08	39.53
35	17.55	10.26	4.40	3.27	209.25	67.00	43.27	30.64
40	16.25	9.03	4.02	3.03	206.14	63.19	41.63	29.33

Finally, the proposed technique is compared with other existing techniques and it is clear that the proposed compression technique outperformed other methods by producing the highest compression ratio and lowest PRD as shown in Table 3.6. In this table some values are blank because other researchers did not provide results for those patient records.

Table 3.4: Average compression time in seconds for 60 second length ECG signals using parallel processing mode with 8 cores processor

Bs	Cr	Jump Step									
		1	2	3	4	5	6	7	8	9	10
10	12	108.89	59.76	37.63	29.39	20.89	21.07	19.35	16.15	15.22	13.83
15	18	67.93	33.73	21.85	16.89	14.25	11.94	12.04	9.08	8.37	8.48
20	24	52.80	26.09	17.25	14.40	11.13	9.18	8.12	7.06	9.25	5.49
25	30	44.73	21.00	14.01	10.35	8.31	7.15	6.47	5.20	4.89	4.26
30	36	39.53	18.55	15.89	10.61	7.46	6.73	5.75	8.20	4.60	6.53
35	42	30.64	15.25	10.53	8.49	6.54	5.25	4.40	4.76	3.76	3.27
40	48	29.33	14.75	13.06	7.59	6.99	4.95	4.60	4.03	3.38	3.03

Table 3.5: Decompression Time for different block sizes.

Bs	Cr	Decompression Time
10	12	**2.75**
15	18	**1.74**
20	24	**1.19**
25	30	**1.18**
30	36	**0.91**
35	42	**0.99**
40	48	**0.90**

Table 3.6: Compression ratio and PRD for different compression techniques compared with our proposed technique

Data	[47]		[73]		[15]		[41]		proposed method Bs=35 Js=10	
	CR	PRD	CR	PRD	CR	PRD	CR	PRD	CR	PRD
100	22.94	1.95	24	3.95	24	4.06	11.06	13.79	42	0.79
102	25.9	1.39					10.17	14.2	42	1
103	20.33	2.5					11	13.78	42	2.29
104	22.94	1.67					11.2	13	42	0.96
105	20.96	1.17					12	15.13	42	1.24
106	19.55	1.77							42	1.65
107	18.55	3.93							42	2.66
108	23.11	0.77					8.22	15.42	42	0.53
109	19.89	0.76					11	17.39	42	0.87
111	22.99	1.03							42	0.92
112	23.82	1							42	1.18
113	19.96	2.89							42	2.4
114	25.58	1.08							42	0.87
117	24	1.17	24	1.72	13	1.18	4.5	14.64	42	1.45
119	19	2.05	20	1.92	21	2.81			42	1.5

3.7.2 Fast fractal compression Experiments and Results

A test bed of 47 ECG signals collected from MIT-BIH Arrhythmia Database [1; 64] is used to test the proposed algorithm. Each ECG segment is of 60 seconds length. The efficiency of the proposed algorithm depends on changing two parameters which are:

1. Jump step. It can be defined as the number of ECG points that separate two consecutive overlapping domain blocks. Increasing this parameter causes an increase in PRD value with a shorter processing time.

2. Range block size. It represents the number of ECG points in one block. It will affect the compression ratio, the PRD value and the compression time.

Experiments are performed on all ECG signals. The same experiments are repeated for different block sizes. In each case average PRD, average Compression ratio and average compression and decompression times are calculated. Next, the second parameter (jump step) is changed and the same set of experiments are repeated for the new value of jump step. Table 3.7 shows the PRD values and compression ratio for different cases of block sizes and jump steps. It is clear from this table how the proposed technique could achieve a compression ratio of 48 with maximum PRD of 2.5 %. Generally, the PRD value increases when we increase the block size. Moreover, increasing block size will increase the compression ratio. Figure 3.12 illustrates the relationship between the PRD and compression ratio for different cases of jump steps.

The relation between the selected block size and the resultant compression ratio is linear and can be clearly shown in Fig. 3.14. Moreover, the proposed fast fractal compression

Table 3.7: Compression ratio veses PRD for different block sizes and jump steps

Bs	Cr	PRD									
		Jump step									
		1	2	3	4	5	6	7	8	9	10
10	12	0.497	0.642	0.800	0.751	0.804	0.706	0.819	0.787	0.912	0.719
15	18	0.654	0.811	0.800	1.002	0.954	0.862	1.016	0.925	1.034	0.945
20	24	0.677	0.761	0.900	1.016	1.034	1.023	1.065	1.092	1.15	1.239
25	30	0.861	0.990	1.062	1.187	1.146	1.088	1.131	1.237	1.245	1.401
30	36	1.153	1.151	1.7	1.374	1.694	1.669	1.499	2.66	1.457	1.620
35	42	1.815	1.199	1.956	1.933	1.471	1.487	1.565	2.03	1.689	2.020
40	48	1.557	1.684	2.122	1.524	1.730	1.787	1.619	2.432	1.85	1.82

technique performance is examined by recording the compression time. To compare between fast fractal compression and the normal fractal compression technique performance, average compression time is calculated for both techniques and for the same data sets using single processing model. Table 3.8 clearly shows how the proposed fast fractal technique performs well compared to the normal fractal compression technique. In this table, compression time is calculated for different cases of jump steps using a 60 second length ECG signal. However, the accuracy of decompressed signal is less than the normal fractal compression technique as shown in Table 3.9

The proposed fast fractal compression is applied to different ECG signals with different types of diseases, Moreover, the same experiments are repeated with different block sizes, the decompressed ECGs can be shown in Fig. 3.13. The large similarity between original ECG and the decompressed ECG is clearly shown in this figure.

To further prove the reliability of the proposed technique, it is compared with other lossy compression techniques as shown in Table 3.10. It is clear that the proposed algorithm will

Table 3.8: Average compression time for both normal and fast fractal compression for different jump steps of 60 seconds ECG signal

jump step	Normal fractal	fast fractal
1	348.4712258	6.128656172
2	178.0142764	3.173848511
3	117.9738631	2.104716943
4	88.24959419	1.681290864
5	70.95361335	1.428648108
6	59.20645766	1.253933547
7	51.83929228	1.065914267
8	45.70668542	1.205495831
9	40.05596818	0.942461358
10	36.0513147	1.097966095

Table 3.9: Average PRD value for both normal and fast fractal with respect to different jump steps

jump step	Normal fractal	Fast fractal
1	0.623819781	1.031128998
2	0.642192507	1.034331828
3	0.687568786	1.334678738
4	0.708100499	1.255598594
5	0.679156509	1.262343616
6	0.676764759	1.232162049
7	0.729272129	1.24524157
8	0.752507842	1.596466452
9	0.693918281	1.334330344
10	0.687600491	1.396449332

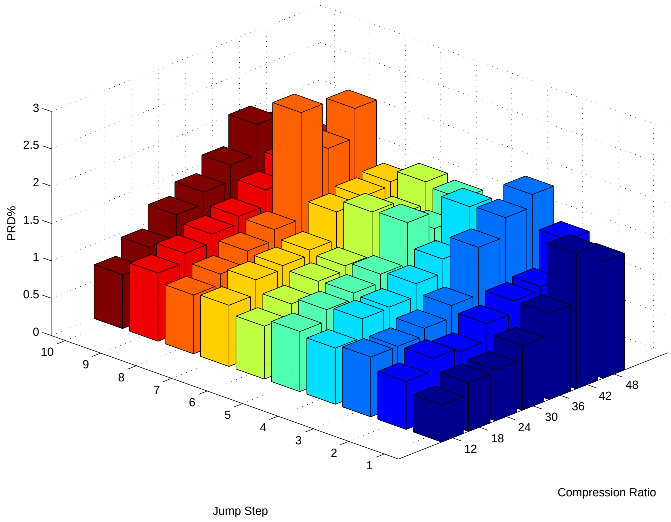


Figure 3.12: Relation between Compression ratio and PRD for different Jump Steps

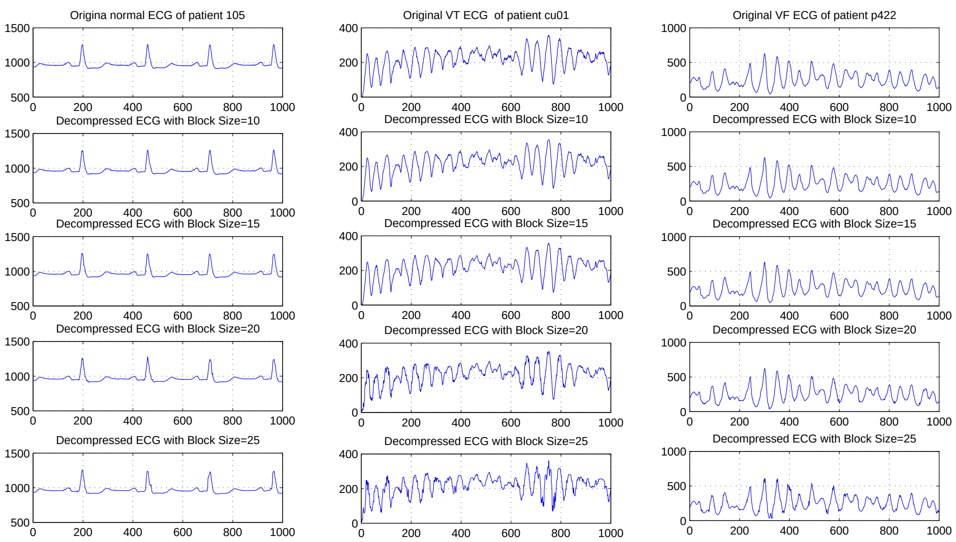


Figure 3.13: Decompressed ECG signals from different patients with different physiological conditions as well as different block size

compete with other techniques in terms of both compression ratio and PRD. Moreover, it is found that other researchers did not estimate their techniques compression time.

Table 3.10: Compression ratio and PRD for different compression techniques compared with our proposed technique

Algorithm	Record	CR	Prd
Lee et all [46]	100	24	8.1
Chou et al app [15]	100	24	4.06
Eddie B.L et al app [73]	100	24	3.95
SangJoon Lee [47]	100	23	1.94
Proposed	**100**	**24**	**0.900**
Eddie B.L et al app [73]	117	24	1.72
SangJoon Lee [47]	117	24	1.17
Proposed	**117**	**24**	**0.856**
Chou et al app [15]	117	13	1.18
Eddie B.L et al app [73]	117	13	1.07
SangJoon Lee [47]	117	12.6	0.43
Proposed	**117**	**13.2**	**0.85**
Lu et al [54]	117	10	2.96
Proposed	**117**	**12**	**0.95**
M,Wei et al [84]	117	10	1.18
Beligin et al [9]	117	10	1.03
SangJoon Lee [47]	117	10.4	0.43
Beligin et al [9]	119	21.6	3.76
Tai et al [80]	119	20	2.17
Chou et al app [15]	119	20.9	1.81
Eddie B.L et al app [73]	119	20	1.92
SangJoon Lee [47]	119	19.3	2.05
Proposed	**119**	**20.4**	**1.01**
Hyejung Kim; [43]	119	16.9	0.64
Proposed	**119**	**16.8**	**0.79**

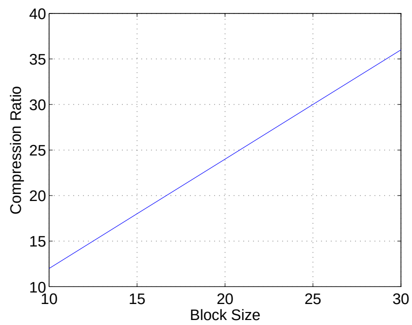


Figure 3.14: Block size and Compression ratio

3.8 Summary

Continuous cardiac monitoring systems produce huge ECG signal size. Therefore, a new fast fractal lossy compression technique is introduced in this chapter to reduce ECG signal size in an efficient way in terms of compression ratio, information loss and compression time. In this chapter, two cases were taken into consideration. Firstly, a cloud enabled ECG compression technique that can make use of the parallel processing capability of the cloud without adding a large overhead is proposed. A fractal model is used and modified to be suitable for the proposed ECG compression technique. ECG self similarity can be effectively utilized in ECG compression techniques. Finally, the proposed technique outperformed other ECG compression techniques with compression ratio of 40 and PRD value less than 1%. Secondly, a modified version of the normal fractal ECG compression is proposed that can be applied inside a client device instead of within the cloud. The proposed method is based on modifying the normal fractal compression technique to increase its performance by limiting the search domain. Further, the proposed technique out performed other ECG compression techniques with compression ratio of 30 and PRD value less than 1%. Moreover, compression time is very small (less than 6 seconds to compress 60 seconds ECGS signal).

Chapter 4

ECG Lossless Compression

In this chapter, we are addressing the third research question discussed in Section 1.2. A new lossless ECG compression technique is proposed to guarantee zero loss of information after compression. Moreover, the proposed technique guarantees that ECG features are still detectable. Accordingly, diagnoses process can be performed directly from the compressed file without decompression operation. This feature (i.e. diagnoses from compressed ECG) will be discussed in more details in the next chapter. This chapter is organized as follows: Section 4.1 discusses the problem statement and the chapter contributions. Section 4.2 investigates other available lossless compression techniques in the literature. Section 4.3 explains the proposed technique in more details. Results and experiments are explained in Section 4.4. Finally, Section 4.5 summarises this chapter.

4.1 Introduction

Electrocardiogram(ECG) signals, which represent the electrical activities of heart are widely used in cardiac disease detection. ECG signals can be collected for many hours or days to continuously monitor the heart activities of subjects being monitored in hospitals or at homes. Moreover, ECG signals are digitally converted into digital data using Analogue to Digital Converters (ADC) [56].

In currently used conventional cardiac monitoring e-health systems [32], body sensors are responsible for collecting patient ECG signals. Subsequently, huge amount of ECG signals are processed and transmitted to the hospital server or e-health cloud over a limited-bandwidth mobile network. Moreover, this enormous amount of data needs to be stored; thereby necessitating huge storage capability. Generally, sampling rate used to generate ECG signals ranges from 125 to 500 samples per second, but can be as high as 1024 samples/sec. Each sample is represented in binary using 8 to 16 bits. Therefore, for a 12-lead ECG signal, the resulting digital data size can be from 1 GB to several gigabytes for 24 hours of continuous monitoring. The increasing use of ECG signals in several e-health applications and the growing acceptance of such applications by elderly as well as healthy individuals are making ECG compression necessary and extremely important [89; 44].

In a typical remote cardiac monitoring health-care system supporting compression, patients wear body sensors to collect their ECG signal. Then, the ECG signal is sent to the patient's PDA device via Bluetooth or ZigBee. Subsequently, the compression and processing of the ECG signal is completed on the patient's PDA device. Finally, the compressed signal is transmitted to the hospital server or e-health cloud. Doctors can access patients ECG by downloading the required compressed ECG signal and decompress it on their PDAs, tablet or laptop. They can then diagnose and interpret the decompressed signal, which is the same as the original ECG signal. This scenario is shown in Fig. 4.1.

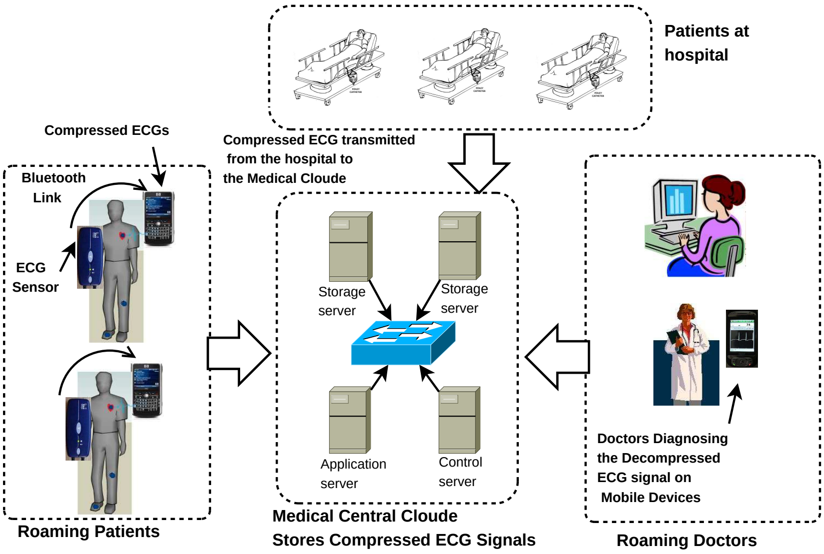


Figure 4.1: Remote cardiac monitoring system showing application scenarios of ECG compression

There are two general types of compression: lossy and lossless compression. In lossy compression, a small portion of the information is lost. Therefore, the decompressed signal is slightly different from the original ECG signal. On the other hand, lossless compression technique guarantees exact reconstruction of the ECG signal after the decompression operation [30]. Lossy compression can achieve higher compression ratio compared with lossless compression techniques. However, in some cases, the use of lossy compressed ECG signal is discouraged for diagnoses purposes because of the loss of certain information. Therefore, lossless compression has become important to guarantee exact reconstruction for ECG signals. Accordingly, decompressed signals using lossless compression can be used for diagnoses purposes.

Compression techniques can be divided into three main categories: direct techniques, transformation techniques and parameters extraction techniques. In direct techniques, ECG signals are processed directly in the time domain without applying any transformation [89; 93]. On the other hand, transformation techniques are based on applying a special transformation, such as FFT, DWT and DCT [62; 20]. Finally, in parameter extraction techniques, models are applied to extract some parameters that can be used to regenerate an approximation of the ECG signal [30].

In this chapter, a new lossless ECG compression technique is introduced. The basic idea of the proposed technique is to approximate the ECG signal using the Gaussian functions model. Next, the encoding process is applied on the differences between the approximated and the original signals. The encoding process includes the block-sorting technique, followed by the move-to-front (MTF) and run-length encoding. Finally, the resultant data will be entropy coded using Huffman coding.

4.2 Related Work

In the past, many studies have been conducted on ECG compression. Although most of the previous work focuses on lossy compression, some work, including that of Krzysztof Duda and Pawel Turcza [20], has considered lossless compression. They proposed a lifting wavelet-based ECG signal compression. The researchers used the integer-to-integer lifting wavelet transform to prevent any information loss. Moreover, the resultant integer wavelet coefficients are represented using Magnitude-Set Variable-Length Integer (MS-VLI) integer representation.

Finally, the resultant coefficients are entropy coded. The maximum compression ratio they could achieve is 3.7. Moreover, the researchers used dynamic lifting filters and calculated the system parameters based on the dataset under test. Therefore, the proposed values in their work cannot be generalized and used for any ECG signal.

Shaou-Gang Miaou and Shu-Nien Chao [62] proposed a new wavelet-based lossy to lossless ECG compression. They used dynamic vector quantization algorithm in conjunction with a distortion-constrained codebook replenishment (DCCR) mechanism, where new replenishment code vectors are SPIHT (Set Partitioning In Hierarchical Trees) encoded. The maximum compression ratio they could achieve is 3.5. In their work, the researchers selected the value of a specific parameter d_{CB} experimentally according to the dataset used. Therefore, their results cannot be generalized for all ECG signals.

Qunyi Zhou [93] proposed a new ECG lossless compression technique based on k-means clustering model. The proposed technique begins with the extraction of all the QRS complexes using the QRS detection technique. Next, the ECG signal is divided into 16 segments and each segment represents a cluster. The K-means algorithm is used to classify the extracted QRS complexes into 16 clusters. Next, differences are calculated between each QRS complex and its average cluster signal. Finally, all the differences is entropy coded using Huffman code. The maximum compression ratio achieved is 3.8. This method relies on accurate QRS detection, however, in some abnormal ECG signals, it is difficult to detect QRS complex. Therefore, this method is ineffective in such circumstances.

On the other hand, there are many researches about lossy compression techniques. Hsiao-Hsuan Chou [14] proposed a new lossy technique suitable for irregular ECG signals. Their technique starts by converting the ECG signal to a 2D image by applying QRS detection stage. Then, period-sorting technique is applied. Next, period-length equalization using mean technique is used. Finally, the researchers used Jpeg2000 encoding algorithm to compress the resultant 2D image. The Jpeg2000 compression is used in its lossless and lossy modes. They achieved a lossless compression ratio of 3.08. Because this technique is based on the accuracy of the QRS detection algorithm, it is difficult to apply it for different abnormal ECG signals.

SangJoon Lee [47] proposed a new lossy real time ECG compression technique. Firstly, the researcher down samples the ECG signal to half of its original sampling rate. Secondly, sample differencing, followed by R-peak detection is applied. Next, DCT transform is applied on two consecutive ECG periods, and subsequently, the floating-point DCT coefficients are converted to integers and accumulated error is calculated. Finally, Huffman coding is used as an entropy coding.

Generally, most lossless compression technique comprise three main stages: prediction stage, transform stage and encoding stage as shown in Table 4.1

4.3 Methodology

In this chapter, the proposed technique involves initially approximating the ECG signal and subsequently representing it with a few parameters. Next, the errors are calculated and encoded using Burrow-Wheeler transform followed by MTF encoding and Run-Length Encoding (RLE), Finally, entropy coding is applied as shown in Fig. 4.2. The main challenge in the proposed algorithm is the accuracy of the Gaussian model and evaluating the suit-

Table 4.1: Summary of the available lossless compression techniques

Reference	Prediction	Transform	Coding	CR
[20]	-	Lifting Wavelet MS-VLI	Huffman coding	3.7
[62]	-	5/3 lifting Wavelet DCCR	SPIHT	3.5
[93]	k-means and differencing	-	Huffman coding	3.5
[14]	-	ECG to 2d image	Jpeg2000	3.08
[47]	Differencing R-peak detection	DCT	Huffman coding	16 PRD = 0.6%
[65]	Differencing sign generation	Scaling and grouping	ASCII	7 PRD = 0.02%
[12]	Delta	-	Huffman coding	3

able model parameters that will produce a good approximated ECG signal. Accordingly, optimization theory is used to determine the correct parameters values.

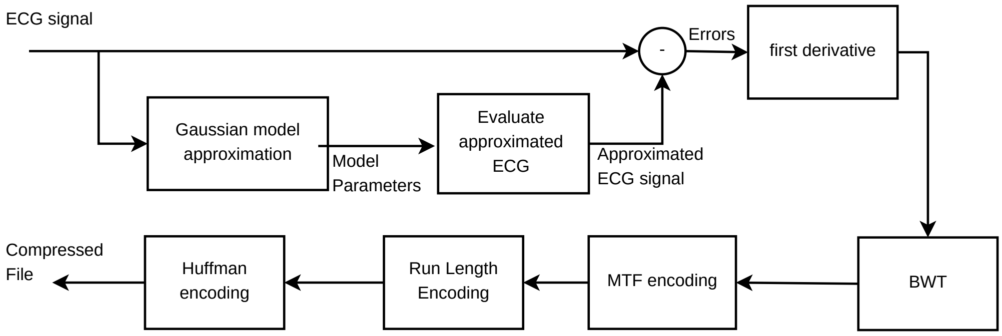


Figure 4.2: Lossless Compression Block Diagram

4.3.1 Gaussian Approximation

Typical ECG signal consists of several waves such as P and T wave as well as QRS complex as shown in Fig. 4.3. Each wave is similar to the Gaussian signal with a different amplitude or width. Therefore, Gaussian functions are utilized in the proposed compression technique to approximate the ECG signal. Gaussian function is plotted in Fig. 4.4. Moreover, it can

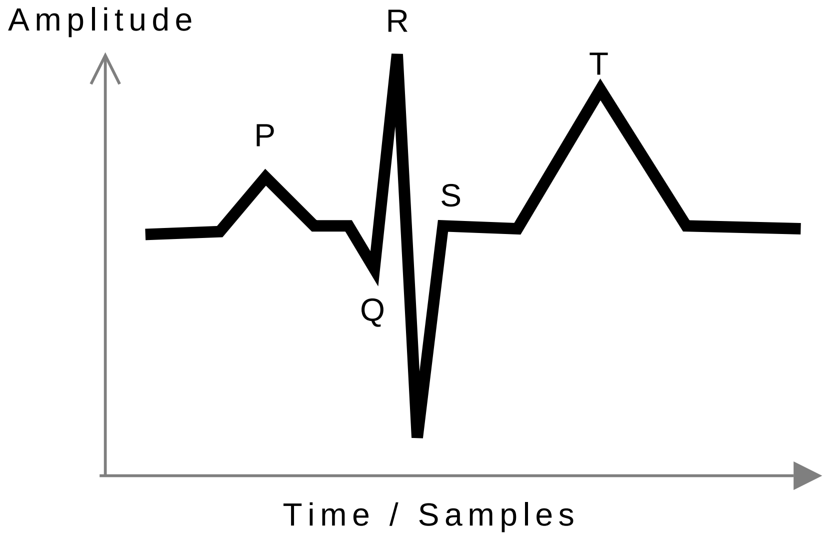


Figure 4.3: ECG signal one beat consists of P,QRS and T wave.

be represented as expressed in Eq. 4.1

$$f(x) = ae^{-\left(\frac{(x-b)}{c}\right)^2} \tag{4.1}$$

where x is an independent variable and a, b and c represent the Gaussian function parameters as follows:

- a is the amplitude of the maximum peak value.
- b is the centroid of the function.
- c is the width of the peak.

To represent more than one peak, the same Gaussian equation can be rewritten as shown in Eq. 4.2:

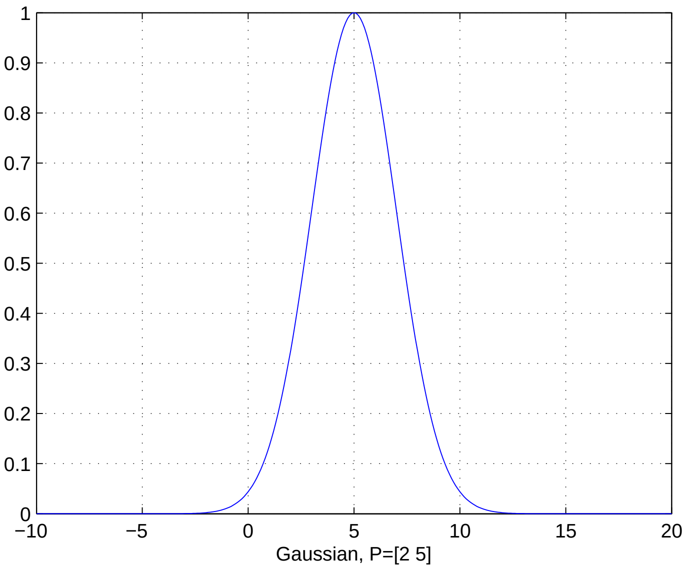


Figure 4.4: Gaussian function

$$f(x) = \sum_{i=1}^{n} a_i e^{-\left(\frac{(x-b_i)}{c_i}\right)^2} \tag{4.2}$$

where n represents the number of Gaussian functions (number of peaks required to be fitted). The ECG data represent the values of $f(x)$. Then, the values of the parameters should be calculated using the suitable optimization theory (in the proposed compression technique, trust region algorithm is used) in such a way that minimizes the difference between the fitted values and the actual values of the ECG signal. The difference between the original ECG (y) and the Gaussian approximated ECG signal ($\widehat{y}$) is calculated using a non-linear least square equation as shown in Eq. 4.3

$$f = \sum_{i=1}^{n} (y_i - \widehat{y}_i)^2 \tag{4.3}$$

where f represents the sum squares of the residuals that should be minimized. Therefore, f in Eq. 4.3 represents the objective function that must be minimized by finding the optimum Gaussian model parameters stored in vector x. Trust region algorithm [10] is used for this purpose. It is based on approximating the original function f using a quadratic equation $m_k(p)$ to determine the optimum step size p by which the parameter values x should be increased or decreased. Moreover, the step value at iteration k can be determined by solving this quadratic equation [86] as presented in Eq. 4.4

$$m_k(p) = f_k + p^T g_k + \frac{1}{2} p^T B_k p \tag{4.4}$$

where $f_k = f(x_k)$ is the value of the objective function at iteration k using the current parameters values x_k, and g_k is the gradient of both f and parameter values of x_k at iteration k. Moreover, B_k is the Hessian of f. The solution of this equation (the step size value p) is limited to a specified region Δ_k called trust region as shown in Eq. 4.5

$$\|p\| \leq \Delta_k \tag{4.5}$$

Finally, the trust region is either increased or decreased based on whether the approximated quadratic function represents good approximation to the original objective function or not. For this purpose, a reduction factor r_k is introduced to test the performance of the quadratic approximation as shown in Eq. 4.6

$$r_k = \frac{f(x_k) - f(x_k + p_k)}{m_k(0) - m_k(p_k)} \tag{4.6}$$

Finally, according to the value of r_k, the trust region Δ_k is changed as follows:

- if the value of $r_k > \frac{3}{4}$, then increase Δ_k
- if the value of $r_k < \frac{1}{4}$, then decrease Δ_k
- otherwise, do not change Δ_k

Next the parameter values are updated using the calculated step p_k and the operation is repeated until the stop condition is reached. Gaussian model parameters can be calculated using the steps shown in Algorithm 7.

Algorithm 7 Nonlinear fitting algorithm

Initialize the initial Gaussian parameters values x_0, maximum step length $\hat{\Delta}$, initial trust region size $\Delta_0 \in [0, \frac{1}{4})$
f:the objective function to be minimized which is the least square error in Eq. 4.3
k=1
while x_k is not optimum **do**
    solve Eq. 4.4 and 4.5 to get the step value p_k
    Calculate the reduction factor r_k as in Eq. 4.6.
    if $r_k > \frac{3}{4}$ **then**
        $x_{k+1} = x_k + p_k$
    else
        $x_{k+1} = x_k$
    end if
    if $r_k < \frac{1}{4}$ **then**
        $\Delta_{k+1} = \frac{1}{4}\Delta_k$
    else if $r_k > \frac{3}{4}$ and $\|p\| = \Delta_k$ **then**
        $\Delta_{k+1} = min(2\Delta_k, \hat{\Delta}_k)$
    else
        $\Delta_{k+1} = \Delta_k$
    end if
    $k = k + 1$
end while

4.3.2 Barrow-Wheeler Transform (BWT)

Before this stage, the differences between the real ECG signal and the Gaussian-approximated ECG signal is calculated. The first derivative is applied on these differences to eliminate any deviation in the errors calculated (i.e ensure there is no large difference between any consecutive error values). In this stage, a block sorting transform is used to transform the data into another format that increases its compressibility by rearranging the data to generate long sequences of consecutive similar symbols. The original BWT was first introduced to be used for text compression [11]. The general idea of this transform is that it deals with the data as one block. Alternatively, data can be divided into several blocks and the same transform can be applied on each block separately. Let S represent the required block of symbols (text or numbers) to be compressed as shown in Eq. 4.7,

$$S = \{s_1 s_2 \ldots s_n\} \tag{4.7}$$

where n is the length of the block to be compressed, and s represents either a character or a number. For ECG compression, s will represent a number. Next, the BWT starts by rotating the S vector to the left in circular fashion for each step to generate a new $n \times n$ matrix called B, as shown in Eq. 4.8

$$B = \begin{bmatrix} s_1 s_2 s_3 s_4 \ldots s_n \\ s_2 s_3 s_4 \ldots s_n s_1 \\ s_3 s_4 \ldots s_n s_1 s_2 \\ \vdots \\ s_n s_1 s_2 s_3 \ldots s_{n-1} \end{bmatrix} \tag{4.8}$$

It is obvious from Eq. 4.8 that each row in matrix B represents a different permutation of S. After generating B, the rows in B are sorted in ascending order. Therefore a new sorted matrix will be generated from B, which is called B_s. The final output of BWT is to take the last column of B_s (L) and the index (I) of the row that is similar to the original block S.

An example of an ECG signal block is shown in Eq. 4.9.

$$S = \begin{bmatrix} 0.025 \\ 0.050 \\ 0.025 \\ 0.050 \\ 0.1 \\ 0.5 \\ 0.025 \end{bmatrix} \tag{4.9}$$

To explain the concept of the proposed compression technique and for illustration purposes, the values in Eq. 4.9 are replaced by characters, as shown in Table 4.2. Accordingly, $S =$`ababcda`. To apply BWT, S will be shifted left to generate B as in Eq. 4.10

Table 4.2: Conversion from Numerical values to Character values

Character	Number
a	0.025
b	0.050
c	0.1
d	0.5

$$B = \begin{bmatrix} \mathtt{a\ b\ a\ b\ c\ d\ a} \\ \mathtt{b\ a\ b\ c\ d\ a\ a} \\ \mathtt{a\ b\ c\ d\ a\ a\ b} \\ \mathtt{b\ c\ d\ a\ a\ b\ a} \\ \mathtt{c\ d\ a\ a\ b\ a\ b} \\ \mathtt{d\ a\ a\ b\ a\ b\ c} \\ \mathtt{a\ a\ b\ a\ b\ c\ d} \end{bmatrix} \tag{4.10}$$

The next step is to sort matrix B to generate the sorted matrix B_s as in Eq. 4.11

$$B_s = \begin{bmatrix} \mathtt{a\ a\ b\ a\ b\ c\ d} \\ \mathtt{a\ b\ a\ b\ c\ d\ a} \\ \mathtt{a\ b\ c\ d\ a\ a\ b} \\ \mathtt{b\ a\ b\ c\ d\ a\ a} \\ \mathtt{b\ c\ d\ a\ a\ b\ a} \\ \mathtt{c\ d\ a\ a\ b\ a\ b} \\ \mathtt{d\ a\ a\ b\ a\ b\ c} \end{bmatrix} \tag{4.11}$$

Finally, the output will be the last column, which is $\mathtt{L} = \mathtt{dabaabc}$ and the index of the row that contains the original block. In this example, it is row number 2, that is $I = 2$.

The decoder will start from the received output (L, I) and repeat the following steps n number of times ($n = 7$ in our example)

1. insert L as the first column of a temporary matrix $Temp$
2. sort the matrix $Temp$ using rows as sorting keys.

At the end of this loop, the resultant matrix $Temp$ will be exactly similar to the matrix B_S in Eq. 4.11. Therefore, the data in row position I ($I = 2$ in this example) will be the original data.

4.3.3 Move to Front Encoder

The next stage of the proposed compression algorithm is to use MTF encoder for its ability to increase the probability of small numbers and decrease the probability of large numbers. Accordingly, the data average is minimized. Therefore, the data will be more compressible using statistical compression techniques such as Huffman coding. The output of BWT consists of a row of symbols where long runs of similar symbols are clustered together. However, in the case of ECG signal, the symbols are different numbers such as, 10, 9, 120, etc. The MTF encoder will increase the number of the symbols near zero, such as 1,2,5,10,etc. On the other hand, the number of symbols that represent large numbers will be decreased, such as 200,500,1000,etc. The MTF algorithm works as follows:

1. Initialize a list containing all the distinct symbols used in compression (in our example $\mathtt{P} = (\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d})$.

2. Each item L_i in data L will be encoded as the number of symbols preceding it in P. Next, symbol L_i is moved to the front of P.

3. Combine the codes of step 2 in a list C, which represents the final output of the MTF encoder, as shown in Table 4.3.

The decoding process is the inverse, assuming that both the encoder and the decoder know the initial value of P, which in our example is `(a,b,c,d)`. The encoding process can be explained in details, for example, if L=`(d,a,b,a,a,b,c )` and P=`(a,b,c,d)`, the first element in L is `d`, and the number of symbols preceding `d` in P is 3; therefore, the encoder will output the number 3. Moreover, the encoder will move `d` to the front of the list in P. As a result, P=`(d,a,b,c)`. The next symbol in L is `a`, which has one element preceding it in P. Therefore, the encoder will output 1 and put `a` to the front in P. The new list P=`(a,d,b,c)` is updated. Then, `b` is the next symbol in L, and the encoder will output 2 and the updated list will be P=`(b,a,d,c)`. The fourth element in L is `a`, so the output will be 1 and the updated list will be P=`(a,b,d,c)`. The fifth element in L is also an `a`. Therefore, there are no preceding symbols of `a` in the list P. Accordingly, the output will be 0 and the list P will remain unchanged. Then, `b` is the next element in L , and the output will be 1 and updated list is P=`(b,a,d,c)`. Since the last element in L is `c`, the output of the encoder will be 3 and the final list will be P=`(c,b,a,d)`. As a result, the encoded message $M = (3, 1, 2, 1, 0, 1, 3)$.

4.3.4 Run-Length Encoding

In this stage, the output of the MTF encoder contains long runs of small numbers, especially zeros. Therefore, long runs will be replaced by the symbol itself and a special code combined

Table 4.3: MTF encoding for $L = (\texttt{d}, \texttt{a}, \texttt{b}, \texttt{a}, \texttt{a}, \texttt{b}, \texttt{c})$ *and* $P = (\texttt{a}, \texttt{b}, \texttt{c}, \texttt{d})$

L_i	P	C
d	a,b,c,d	3
a	d,a,b,c	1
b	a,d,b,c	2
a	b,a,d,c	1
a	a,b,d,c	0
b	a,b,d,c	1
c	b,a,d,c	3

with the run length [74]. For example, the data (aaaaaaaab) will be (a#8b) .

4.3.5 Huffman Coding

Huffman coding [31] is the last step in the proposed lossless compression technique. It is a variable length coding technique by which symbols with a large number of occurrences will be assigned a short code length, whereas symbols with small number of occurrences will be assigned a long code length. Therefore, this method of coding is called variable length coding. To code the symbols, first, their occurrence frequencies should be calculated. Next, Huffman tree is constructed according to the following steps [74].

1. Create a leaf node for each symbol, containing the symbol itself and its frequency. Then, add all the nodes to a priority queue.

2. Repeat the following steps until only one node is left in the queue:

   (a) Remove the two nodes with the lowest frequencies from the queue

   (b) Make a new internal node in the Huffman tree with the two removed nodes as its

Table 4.4: Symbol frequency distribution in message M

Symbol	Frequency
`a`	2
`b`	3
`c`	4
`d`	6

children. Moreover, the frequency of the new internal node will be the sum of its children frequencies.

(c) Add the internal node to the queue.

3. The last node will be the root of the Huffman tree

To illustrate the Huffman code construction, let us consider the message M=`(a,b,c,d,a,b ,b,c,c,c,d,d,d,d,d)`. The frequency distribution for the symbols can be shown in Table 4.4. Initially, the Huffman tree will comprise the leaves of all the four symbols. Then, the symbols `a` and `b` will be the lowest frequency symbols that will be removed from the queue and a new internal node of the tree will be created with a frequency of 5 (summation of frequencies of `a` and `b`). The new internal node is also added to the queue. Now, the queue contents are `(ab/5,c/4,d/6)`. Next, a new internal node is created `abc/9` with children of `ab` and `c`. The new queue will be `(abc/9,d/6)`. Finally, a new internal node is created, which will be the root of the tree `(abcd/15)`. The resultant Huffman tree for this example is shown in Fig. 4.5.

Finally, to construct the Huffman code for each symbol, the tree needs to be trajected from the root to the required leaf. The resultant Huffman code for the given example is shown in Table 4.5

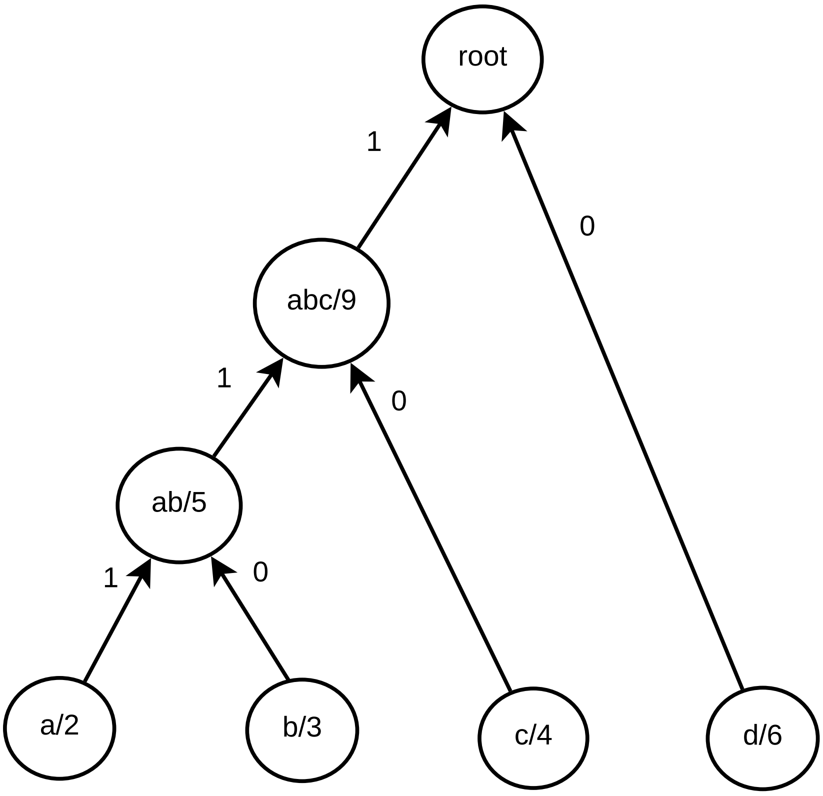


Figure 4.5: Huffman tree for the message `abcdabbcccddddd`

For decompression, either Table 4.4 or Table 4.5 should be known to the decompresser. These tables will be stored in the header of the compressed file. For ECG signal the size of this table normally varies between 60 and 150 rows. Therefore, its size is several hundred bytes.

Table 4.5: Huffman code for the symbols given in message M

Symbol	Huffman Code
`a`	111
`b`	110
`c`	10
`d`	0

4.3.6 Decompression Algorithm

The decompression phase is the inverse of the compression phase. Therefore, the decompression operation starts with Huffman decompression. Then, MTF decoding followed by Burrow Wheeler inverse transform are applied, which generate the errors values. Next, integration is applied to the resultant errors. Finally, the Gaussian function parameters stored in the compressed file header is used to generate the approximated ECG signal. Then, the approximated ECG will be added to the errors to construct the exact ECG signal as shown in Fig. 4.6

Finally, Fig. 4.7 shows the original ECG signal before the compression operation and the decompressed ECG signal. It is clear how the decompressed signal is exactly similar to the original one.

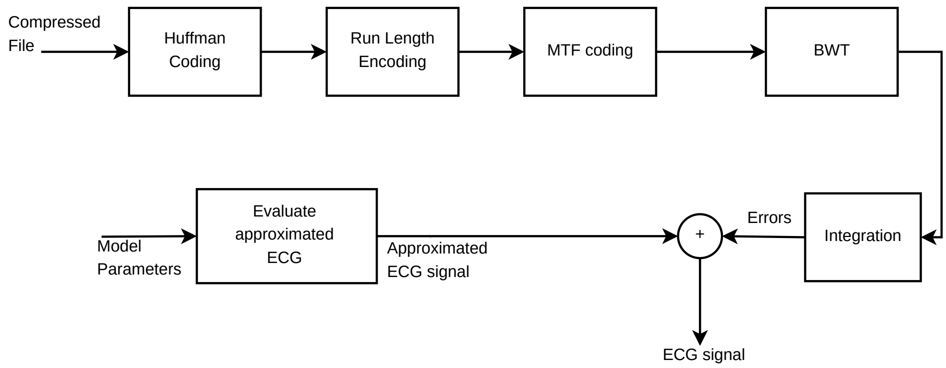


Figure 4.6: Decompression operation block diagram

4.4 Experiments and Results

In this section, a testbed of 47 ECG records are taken from 47 different patients in the MIT-BIH Arrhythmia database to evaluate the proposed algorithm. Each ECG signal is of 30 minutes length. Sampling frequency is 360 sample/s with a sample bit resolution of 12. Moreover, to further test the proposed algorithm, several combinations of compression techniques with several prediction stage algorithms such as, linear, second-degree linear and data mining clustering, is implemented and compared with the proposed technique using the same dataset. It is found that the average compression ratio achieved by the proposed algorithm is 4.09, which is more than the other techniques, as shown in Table 4.6

Table 4.6: Compression Ratio for the proposed technique compared with other techniques

Patient record	Lp 2nd order+AC	LP+BWT+ MTF+AC+ Bs= 1000	LP+HC	Diff+Kmeans +HC bs=100 C=100	Diff+Kmeans +AC bs=100 C=100	LP 2nd + encodig AC	LP 2nd + BWT MTF RLE HC	quasian + BWT MTF RLE HC
'100m'	4.24	3.95	4.23	4.02	4.10	4.06	4.42	4.47
'102m'	4.12	3.85	4.08	3.93	3.99	3.98	4.26	4.31
'103m'	3.95	3.67	3.88	3.84	3.91	3.82	4.18	4.26
'104m'	3.79	3.57	3.74	3.47	3.54	3.70	3.97	4.03
'105m'	3.59	3.54	3.50	3.44	3.50	3.53	3.91	3.92
'106m'	3.63	3.41	3.52	3.35	3.42	3.55	3.74	3.77
'107m'	3.02	3.34	2.94	3.24	3.30	3.04	3.70	3.72
'108m'	3.78	3.55	3.70	3.42	3.49	3.66	3.89	3.90
'109m'	3.61	3.65	3.53	3.71	3.76	3.56	4.06	4.11
'111m'	3.79	3.67	3.69	3.77	3.85	3.70	4.00	4.05
'112m'	4.13	3.96	4.05	3.97	4.04	3.97	4.41	4.44
'113m'	3.72	3.56	3.62	3.66	3.73	3.62	4.02	4.09
'114m'	3.72	3.53	3.63	3.54	3.61	3.62	3.97	4.00
'115m'	4.15	3.88	4.11	3.94	4.00	3.99	4.45	4.51

'116m'	3.55	3.43	3.45	3.41	3.48	3.47	3.91	3.92
'117m'	3.95	3.66	3.92	3.75	3.83	3.82	4.04	4.09
'118m'	3.36	3.26	3.26	3.20	3.26	3.28	3.63	3.63
'119m'	3.61	3.44	3.54	3.48	3.54	3.53	3.90	3.95
'121m'	4.38	4.23	4.33	4.20	4.27	4.23	4.53	4.57
'122m'	3.60	3.48	3.54	3.57	3.63	3.52	3.86	3.89
'123m'	4.01	3.71	3.95	3.78	3.85	3.87	4.12	4.16
'124m'	4.03	3.85	3.97	3.89	3.94	3.91	4.40	4.43
'200m'	3.50	3.40	3.41	3.19	3.25	3.42	3.80	3.81
'201m'	4.42	4.16	4.37	4.19	4.28	4.23	4.65	4.73
'202m'	3.86	3.67	3.79	3.80	3.86	3.76	4.09	4.12
'203m'	3.24	3.15	3.17	3.01	3.05	3.19	3.47	3.48
'205m'	4.35	4.09	4.29	4.10	4.21	4.15	4.71	4.78
'207m'	3.85	3.70	3.76	3.66	3.72	3.74	3.99	4.01
'208m'	3.45	3.36	3.34	3.29	3.35	3.39	3.74	3.77
'209m'	3.53	3.35	3.44	3.34	3.41	3.43	3.76	3.78
'210m'	3.90	3.74	3.87	3.73	3.79	3.80	4.15	4.19
'212m'	3.35	3.20	3.25	3.18	3.23	3.26	3.57	3.58
'213m'	3.14	3.31	3.03	3.27	3.34	3.14	3.90	3.94
'214m'	3.52	3.44	3.43	3.42	3.48	3.46	3.82	3.86
'215m'	3.36	3.29	3.26	3.26	3.32	3.28	3.69	3.70
'217m'	3.29	3.42	3.19	3.39	3.48	3.27	3.76	3.80
'219m'	3.95	3.74	3.84	3.79	3.86	3.82	4.34	4.39
'220m'	4.16	3.89	4.08	3.91	3.99	4.00	4.39	4.43

'221m'	3.76	3.58	3.68	3.62	3.69	3.66	4.02	4.06
'222m'	3.96	3.66	3.89	3.64	3.71	3.83	4.02	4.04
'223m'	3.90	3.80	3.82	3.78	3.85	3.78	4.31	4.36
'228m'	3.69	3.51	3.59	3.36	3.43	3.61	3.81	3.81
'230m'	3.86	3.69	3.75	3.69	3.75	3.73	4.27	4.31
'231m'	4.15	3.86	4.06	3.96	4.05	4.00	4.40	4.46
'232m'	4.33	4.01	4.22	3.99	4.05	4.14	4.39	4.42
'233m'	3.42	3.52	3.30	3.38	3.44	3.39	4.02	4.04
'234m'	3.99	3.77	3.91	3.83	3.90	3.87	4.30	4.34
Average	3.78	3.63	3.70	3.62	3.69	3.68	4.06	4.09

In this chapter, several experiments are performed using different methods as follows:

- **Case1:** The second column in Table 4.6 shows the compression ratios if Gaussian stage is removed and only linear prediction is used in conjunction with arithmetic coding instead of Huffman coding. Moreover, neither BWT nor MTF have been used in this method. The average compression ratio in this case is 3.7, which is lower than that achieved by using the proposed technique.

- **Case 2:** In column three of Table 4.6, the same method used in case 1 is modified by adding BWT and MTF coding, while retaining linear prediction and Huffman coding. The average compression ratio in this case is lower than that achieved by using our proposed technique.

- **Case 3:** Column four shows the results of using the linear prediction directly combined with Huffman coding instead of arithmetic coding. No BWT or MTF coding is used. The resultant compression ratio is 3.7, which is lower than the compression ratio achieved by using our proposed technique.

- **Case 4:** The fifth column in this table describe the results of using the technique in [93], which is based on dividing the signal into blocks and then classifying those blocks into clusters, and finding the differences between each pair of blocks in the clusters and the cluster average. Finally, Huffman coding is applied on these differences. In our experiments, the best result that this technique yielded was an average compression ratio of 3.62, which was achieved when a block size of 100 was used and the number of clusters were 100.

- **Case 5:** The next column repeats the same technique as the one used in case 4, but replaces Huffman coding with arithmetic coding. The compression ratio achieved in this case is 3.69.

- **Case 6:** Column 7 uses linear prediction combined with the ASCII encoding technique used in [77], followed by arithmetic coding. The resultant compression ratio is 3.68.

- **Case 7:** The last two columns represent the difference of using steps comprising BWT, MTF, RLF, and Huffman coding. However, the first stage is either linear prediction or Gaussian prediction, as in our proposed method. This case shows how Gaussian prediction improved the compression ratio to 4.09 on an average.

The experiments clearly show how Gaussian-based lossless technique achieved better results. Moreover, Table 4.7 shows a comparison with other available compression techniques

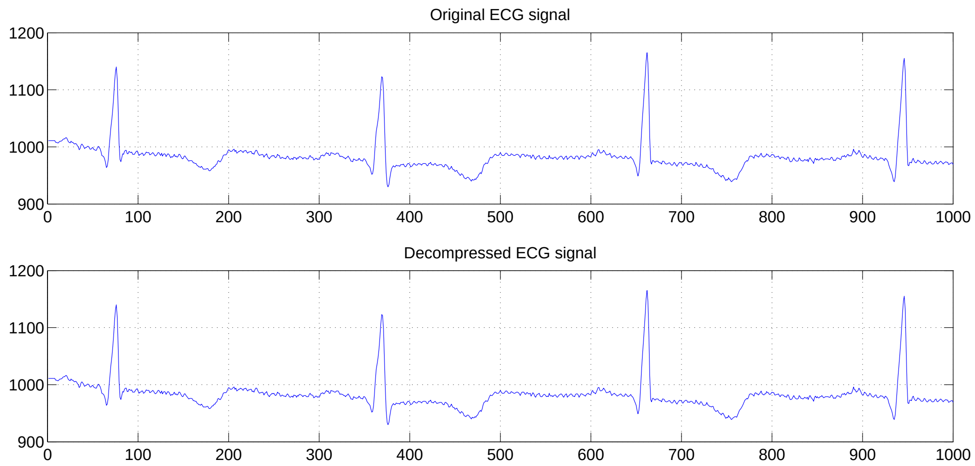


Figure 4.7: Original ECG compared with decompressed ECG signal

based on their published results. Table 4.7 clearly shows that our proposed compression technique performs better than other techniques, with an average compression ratio of 4.09.

Table 4.7: Comparison with other available techniques

Compression Method	CR	PRD(%)
Lifting wavelet [20](Lossless)	3.7	0
wavelet lossy to lossless [62] (Lossless)	3.5	0
kmeans [93] (Lossless)	3.8	0
ASCII characters [65] (Lossy)	7	0.02
wavelet filters [4] (Lossy)	22	2
Delta coding [12] (Lossless)	3	0
Proposed technique (Lossless)	4.09	0

4.5 Summary

In this chapter, we have introduced a new Gaussian-based ECG lossless compression algorithm. The proposed algorithm was based on utilizing the Gaussian function modeling to approximate the ECG signal. Gaussian functions parameters were calculated using nonlinear least square error technique. Moreover, the residuals of this approximation process were encoded using steps comprising BWT, MTF, RLF, and Huffman coding. The proposed technique was tested and compared with other existing lossless compression techniques, and it was found that the proposed technique performs better than others and provides compression ratio of 4.09. Moreover, the proposed compression technique is dynamic because no static parameters are calculated by experimentation and QRS detection is not required. Therefore, our proposed technique can work for any type of ECG signal and any database.

Chapter 5

ECG Diagnoses from Compressed ECG

In this chapter we address the fourth research question posed in Section 1.2. Diagnosing Cardiovascular diseases from Compressed ECG signal without performing any decompression stage is implemented in this chapter. Two life threatening diseases are diagnosed in this chapter: Ventricular Arrhythmia and Left Bundle Branch Block. This chapter is organized as follows: Section 5.1 describes the problem statement and chapter contributions. Section 5.2 investigates other techniques available for ECG diagnoses. Section 5.3 explains briefly the compression technique that will be utilized to diagnose ECG from compressed file. This compression technique is the same as proposed in Chapter 4. Section 5.4 discusses the proposed technique in more details starting from analyzing the compressed ECG, attribute selection processes using Principle Component Analyses and the clustering stage. Section 5.5 explains the experiments performed and discusses the results obtained for diagnoses of

Ventricular Arrhythmia as well as Left Bundle Branch Block disease. Finally, Section 5.6 summarizes this chapter.

5.1 Introduction

With the current development in information technology and communication systems, numerous types of data are collected and processed in continuous manner by networked applications [85]. As an example of data generated we are focusing on Cardiovascular patients' data. Cardiac diseases are the number one killer in the modern era. According to the Australian Bureau of statistics, 3.5 millions Australians suffer from long term cardiovascular diseases [2]. Accordingly, it is recommended that those people should be continuously monitored [50]. To diagnose cardiovascular disease an ECG signal is used [55]. For 12 lead ECG signal with 1024 sampling frequency digitized using 16 bits resolution to guarantee good quality, the amount of ECG data generated by 24 hours continuous monitoring will be 2.1 GB for one patient. Accordingly, to monitor 3.5 million patients, a huge amount of data about 6 PetaBytes daily will be generated and this data has to be stored and processed. This huge amount of data is a valid example of a big data problem in health-care applications[19]. According to IBM, big data is defined as any situation or event that includes any or all of the three V words:Volume, Variety and Velocity [68]. In our case, with remote cardiac monitoring systems, different types of data are stored such as ECG signals, diagnosis report and patient information. Accordingly, the Variety condition is satisfied. Secondly, the Velocity condition is also satisfied because the ECG signal is generated continuously during the monitoring period. Finally, Volume condition is satisfied as well, because in Australia alone the ECG data size can reach 6 PetaBytes daily. Current health-care organizations lack the required technology to handle this big data. Therefore, cloud computing capability should be used. Moreover, storing patient information and biomedical signals in the cloud will provide freedom of access to this information [19].

To decrease the size of the transmitted ECG signal, it should be compressed before it is transmitted to the cloud. Several health service providers require access to patient information and patient diagnostics reports, such as hospitals, ambulance services, clinics, health insurance, researchers and government health sectors. Therefore, cloud services should be used as a central secured storage repository that is capable of storing and processing this big data [83].

The proposed large scale e-health cloud receives chunks of compressed ECG from different sources. In remote patient monitoring, the patient wears a body sensor that reads the ECG signal and sends it to the patient's PDA device. The ECG signal is compressed then transmitted to the e-health cloud. This signal can also be collected from patients in hospitals, compressed and transmitted to the e-health cloud. Doctors can access the stored information from anywhere. Alternatively, the cloud itself can be configured to diagnose the compressed ECG and send alerts in case of any emergency. Therefore, the diagnosis process implemented inside the cloud is capable of analysing the compressed ECG signal without decompressing it as shown in Fig. 5.1. As a result, the diagnosis process is faster and more reliable. Processing the big data in its compressed format will dramatically increase the performance of the whole big data processing operation.

There are many types of cardiac diseases. In this chapter, two cardiac diseases are addressed. Firstly, a new technique for detecting Ventricular Arrhythmia directly from compressed ECG signal is proposed to enable faster diagnoses. Ventricular Arrhythmia is a fast heart rhythms generated from the ventricles. There are two types of ventricular arrhythmia: Ventricular tachycardia and Ventricular fibrillation. Both diseases are life-threatening diseases, so detecting them early increases the chance of survival [16]. Secondly, Left Bundle Branch Block (LBBB) is addressed to prove that the proposed technique can work for more than one disease. LBBB is a cardiac disease where the left ventricle contracts after the right ventricle. The ECG signal is the main tool used to diagnose cardiac disease. There are other applications for ECG signal such as biometric authentication techniques [42; 76; 78].

An ECG signal consists of P and T waves as well as QRS complex. Generally, diagnosis process is based on analysing these waves as well as other features extracted such as RR interval, PR interval, PR segment , ST interval and ST segments [79]. Extracting these features accurately is a complex process, signal processing techniques are used to achieve this goal [58; 63]. Other researchers have proposed wavelet based QRS detection technique where redundant R waves or noise peaks are removed [25]. Another techniques used template matching and neural networks to classify ECG signals. All these techniques are based on analysing the original large ECG signal. Therefore, they are not suitable in cloud scenario where compressed ECG signal is to be diagnosed.

In the proposed e-health cloud scenario the Internet represents the main communication channel. However, ECG data should be transmitted using smart devices such as mobile phones. Furthermore, current mobile communication technologies such as MMS, GPRS, HSDPA or LTE provide a bandwidth-limited channel that is not suitable for transferring

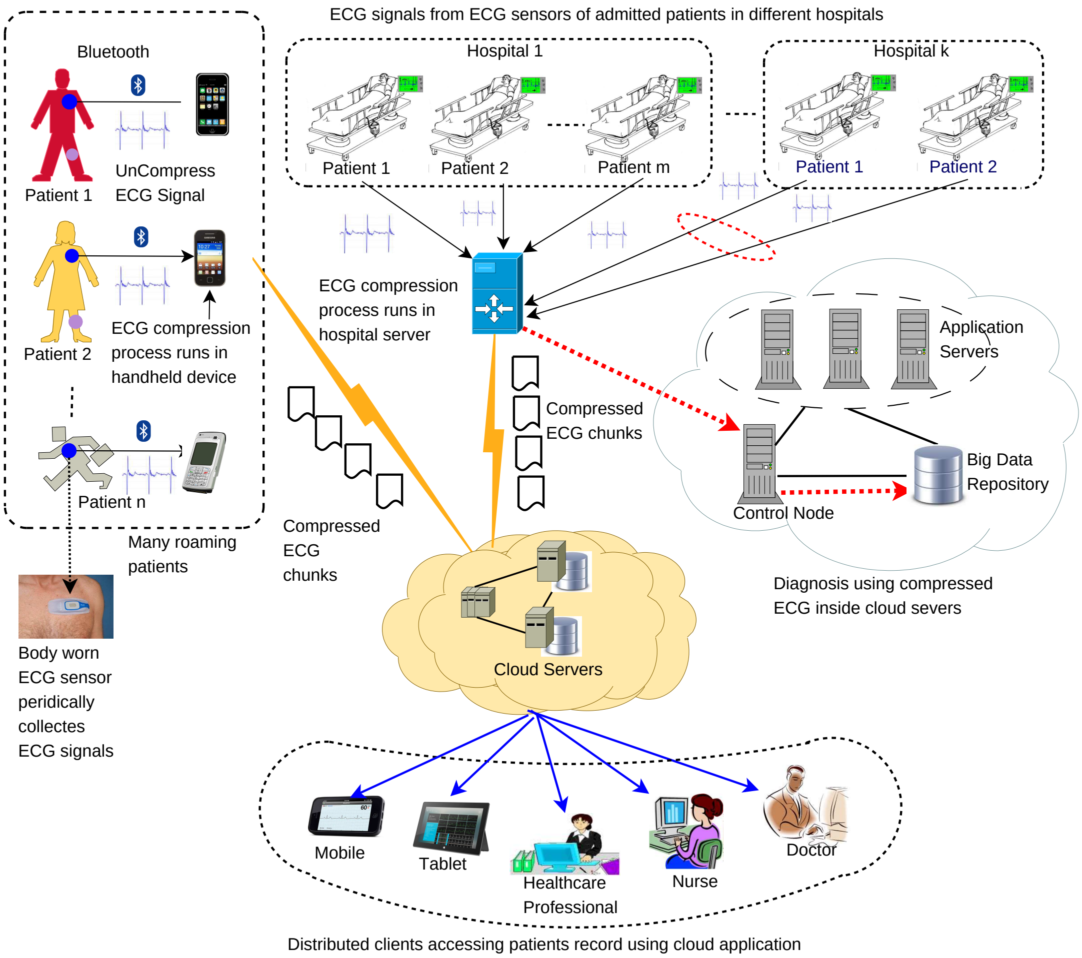


Figure 5.1: A typical wireless tele-monitoring scenario. Compression would save energy on power hungry Bluetooth device, resource constrained wireless sensor nodes and smartphone. Compression also helps transmit faster over bandwidth constrained wireless links. Diagnosis of diseases is implemented on the cloud side and applied directly to the Compressed data.

large amounts of ECG data [77]. Therefore, compression is applied to guarantee fast and energy-efficient transmission. On the cloud side, the compressed data is analysed and Cardiac Arrhythmia are detected directly from compressed ECG signals. In this chapter the lossless technique proposed in the previous chapter is used, compressed data is analysed and important features are extracted using the Principle Component Analysis technique. Finally, extracted features are classified into normal and abnormal using k-means clustering technique. Neural Network classifier is used and replaced k-means technique to further improve the results [39; 21]. The same model is tested to detect ventricular arrhythmia as well as LBBB.

In this chapter three questions are answered:

- Can normal and abnormal ECG data be directly classified from a compressed ECG signal without performing the decompression operation?
- How it is possible to utilize Principal Components Analyses (PCA) to perform attribute selection from compressed ECG signal?
- Can the same technique be used to diagnose more than one cardiac disease (Ventricular Arrhythmia as well as LBBB)?

5.2 Related works

There are many techniques proposed to detect ventricular arrhythmia such as Ventricular Tachycardia and Ventricular Fibrillation. Rahat Abbas [3] proposed new technique to detect ventricular arrhythmia. The authors used a two layers Generalized Neural Network. The first layer consists of radial bases function nodes, and the second layer represents linear function. Secondly, the authors modified the normal K Nearest Neighbour classifier and replaced the Euclidean distance used to find the similarity between two vectors with cross-correlation.

Zhen-Xing [90] proposed a wavelet based Neural Network Weighted fuzzy Membership function to detect ventricular arrhythmia. The researchers applied three levels wavelet transform. Next, they applied phase space reconstruction algorithm on D3 wavelet coefficients to extract six input features. They then used Neural Network with Weighted Fuzzy membership functions combined with non-overlap area distribution measurement method for the classification stage.

A non neural network approach was developed by Nitish and Yi-Sheng [81] using sequential hypotheses testing technique. The researchers started by filtering the ECG signal to remove noise. Next, a sequence of binary numbers is generated based on comparing the instant value of the ECG signal with an experimentally determined threshold. The probability distribution of the binary sequence is then calculated. Finally, sequential hypotheses testing algorithm is applied to classify the ECG signal.

All these diagnosis techniques are based on processing the raw ECG signal and can not diagnose the compressed ECG without performing decompression stage. Therefore, our proposed technique is capable of diagnosing the compressed ECG signal without performing the decompression stage.

5.3 Background: The Compression Algorithm

In this chapter, the ECG lossless compression technique proposed in Chapter 4 is used because of its ability to preserve ECG features after compression. The final compressed file is a text file that uses the ASCII character set shown in Fig. 5.2.

In this section, we summarize the steps used in our lossless compression in the following stages:

- Find the suitable Gaussian parameters and generate approximated ECG signal

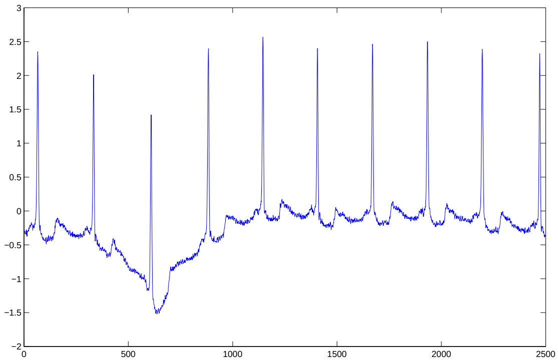
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2	SPC	!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	DEL

Figure 5.2: Character Set for the compressed ECG signal

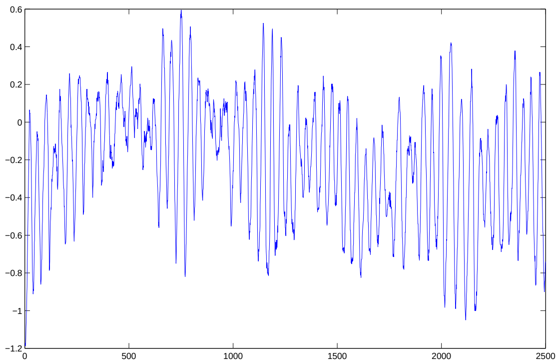
- Find the residuals between the original ECG signal and the Gaussian approximated ECG signal
- Differentiate the residuals to lower their respective amplitudes.
- Apply modified Burrow-Wheeler technique that can deal with numbers instead of text.
- Apply Move to Front encoding technique to decrease the mean of the data
- The number of similar consecutive values is decreased using Run-Length Encoding
- The resulted values are finally encoded using variable length coding called Huffman Coding.

5.4 The Methodology

The proposed diagnosis algorithm can analyse a compressed ECG signal without performing the decompression process. The compression technique used in this chapter is that proposed in Chapter 4 because it is a lossless algorithm. The resulting compressed ECG is analysed and



(a)



(b)

Figure 5.3: (a) Normal ECG sample for Patient CU01 (b) Abnormal ECG sample for Patient CU01

data mining tools are used to classify the compressed ECG signal into normal and abnormal classes for both Ventricular Arrhythmia and LBBB disease.

In Fig. 5.3(a) and (b) normal and abnormal ECG samples for Patient CU01 from the CU Ventricular Tachyarrhythmia Database are presented respectively. The classification process is applied directly to the compressed ECG as shown in Fig. 5.4.

(a)

(b)

Figure 5.4: Compressed ECG samples for patient CU01 (a) Abnormal ECG of 5.3(b) in Compressed Format (b) Normal ECG of 5.3(a) in Compressed Format

5.4.1 Analysis of Compressed ECG signal

The compression algorithm utilises the ASCII characters set shown in Fig. 5.2 to construct the compressed file. The compression operation is performed within the patient's PDA smart device. Alternatively, it can be performed within the hospital server. Inside the e-health cloud servers, a data mining module is trained with the normal and abnormal samples of compressed ECG signals. Accordingly, e-health cloud can classify compressed ECG samples into normal and abnormal classes and for different diseases.

To analyse the compressed ECG, character frequency calculation is applied as shown in Fig. 5.6. Accordingly, the number of occurrences for each character is determined. The number of distinct characters used is 127 characters. For the purpose of analysis, each character is treated as an attribute. Therefore, the total number of attributes in the resultant dataset is 127. However, datasets with large number of attributes (such as 127 attributes) cannot be classified accurately for normal and abnormal. Therefore, in this chapter, attribute selection technique called Principal Component Analysis (PCA) is applied to reduce the number of attributes.

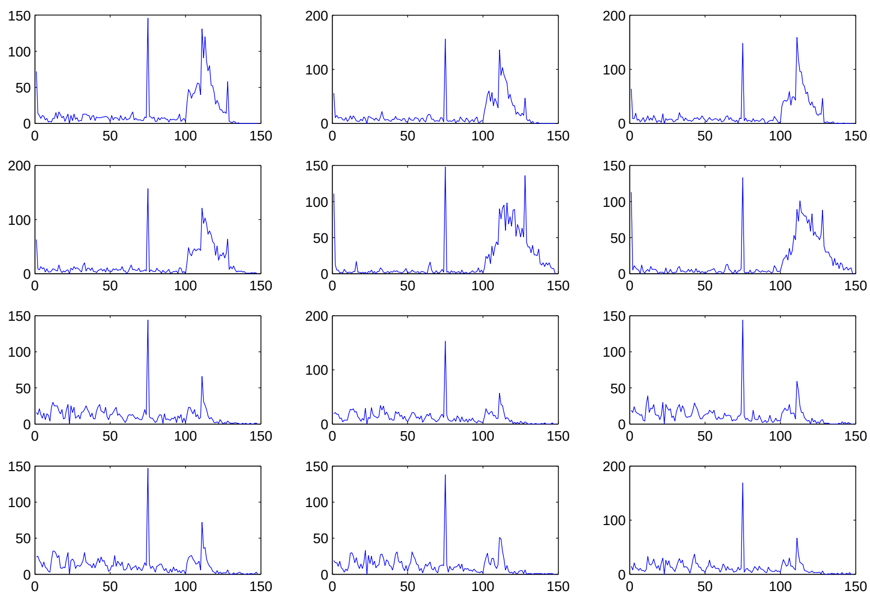


Figure 5.5: data set for patient CU01 the first six plots for abnormal samples and the second six plots for normal samples

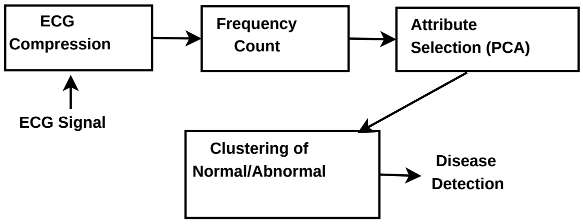


Figure 5.6: Block Diagram for the Proposed ECG detection system

5.4.2 Attribute Subset Selection

Minimising the number of attributes is a crucial process in data mining. The training process will be complex and inaccurate if large number of attributes is used. Moreover, the speed of the classifier will be very slow especially in big data scenario. Therefore, an attribute selection technique is used. This chapter uses PCA technique. Generally, PCA is used for dimensionality reduction, and is suitable to minimise datasets of a large number of attributes by reducing the number of attributes. The output of PCA algorithm is a set of artificial variables called Principal Components. Accordingly, these components are used as inputs to the neural network or k-means clustering technique for classification stage.

A dataset for Patient Cu01 is prepared as an example. 12 samples are taken. 6 of them are normal and the rest are abnormal (Ventricular Arrhythmia). Each sample is compressed and character frequency is calculated to produce the final dataset shown in Fig. 5.5 and Table 5.1.

The mathematical steps for the PCA technique can be summarised as follows

Table 5.1: Data set for CU01

Character	Sample 1	Sample 2	Sample 3	Sample 4	Sample 5	Sample 6	Sample 7	Sample 8	Sample 9	Sample 10	Sample 11	Sample 12
	a1	a2	a3	a4	a5	a6	n1	n2	n3	n4	n5	n6
@	72	56	64	63	111	113	16	19	19	24	19	15
	8	6	7	9	3	5	23	21	17	11	25	14
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
a	13	12	13	11	3	11	8	2	4	5	3	5
b	3	2	9	10	8	8	13	6	8	2	3	8
c	5	2	5	2	2	3	2	5	4	5	2	5
d	7	5	1	4	5	3	8	3	5	5	4	6
e	3	5	3	0	1	4	2	3	4	0	2	4
f	29	23	32	21	8	12	13	13	15	16	15	17
g	47	36	41	48	24	20	23	28	18	23	23	27
h	43	53	43	37	21	22	23	22	22	25	29	21
i	35	60	41	33	27	26	18	18	19	26	16	15
j	41	41	44	42	14	19	14	22	19	21	13	17
k	42	57	59	46	38	35	20	23	27	18	22	30
l	49	33	34	42	25	26	10	15	14	14	22	15
m	56	46	49	45	37	32	13	16	15	15	12	15
n	55	39	49	46	44	53	7	10	15	18	8	11
o	40	29	43	43	40	47	9	10	7	6	13	8
p	131	136	159	121	90	89	66	57	59	72	51	67
q	91	89	118	93	76	73	31	36	46	36	49	37
r	120	103	96	103	91	101	26	34	25	37	32	22
s	86	90	95	92	95	85	20	21	16	21	21	20
t	73	82	73	73	60	83	11	9	16	15	6	18
u	80	75	69	79	98	80	7	12	9	11	12	7
v	53	46	55	72	69	80	9	9	6	9	4	6
w	52	54	58	59	79	70	6	4	4	4	2	5
x	43	41	40	56	66	75	2	7	5	3	3	3
y	27	33	34	34	88	59	2	1	0	1	1	4
z	32	33	40	51	89	83	3	3	8	5	4	6
A	27	17	30	25	52	53	1	1	3	1	1	3
B	19	21	31	35	68	58	6	3	5	3	1	3
C	19	17	18	33	62	52	3	0	1	1	4	3
D	15	14	15	39	51	51	0	5	3	2	5	3
E	16	19	18	29	63	47	2	2	1	2	5	4
F	14	15	17	37	51	54	1	1	5	2	1	1
G	58	47	46	64	136	88	4	4	6	6	6	6
H	3	8	2	8	43	38	2	2	1	1	0	1
I	2	5	2	13	37	30	1	0	1	0	1	0
J	1	7	3	9	37	30	1	0	1	0	1	0
K	3	1	2	14	29	30	1	1	1	2	1	1
L	2	4	1	7	39	24	2	0	0	0	0	0
M	0	0	2	3	26	22	1	0	0	1	1	2
N	1	0	3	5	26	11	1	1	0	2	1	1
O	0	2	0	4	25	21	0	0	0	3	1	1
P	0	1	0	5	34	12	0	0	0	1	1	0
Q	0	0	1	3	14	15	1	0	0	1	1	2
R	0	0	0	4	12	7	0	1	1	0	1	0
S	0	0	0	1	15	15	1	0	0	0	0	0
T	0	0	0	1	10	13	0	2	3	1	1	3
U	0	0	1	1	15	5	0	0	1	0	0	0
V	0	0	0	1	12	7	1	0	2	1	1	0
W	0	0	0	2	15	9	0	0	0	1	1	2
X	0	0	0	1	9	5	0	0	2	0	1	0
Y	0	0	0	2	7	7	1	2	1	1	1	3
Z	0	0	0	1	7	8	1	0	0	3	0	0
\	0	0	0	0	0	0	0	0	0	0	0	0

Step 1: Prepare the dataset X

$$X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1p}; \\ x_{21} & x_{22} & \cdots & x_{2p}; \\ x_{31} & x_{32} & \cdots & x_{3p}; \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{bmatrix}$$

where n represents the number of samples (6-12) for each patient, p is the number of variables i.e 127 (number of characters in the frequency count) and X_{ij} the character frequency count (i=1,2,... n and j=1,2,... p)

Step 2: We calculate the mean of each column of X and get the row vector

$$M = \begin{bmatrix} m_1 & m_2 & \cdots & m_p \end{bmatrix}$$

where m_i i=1,2,...p is the mean of column i of matrix X and calculated as follows

$$m_i = \frac{\sum_{k=1}^{n} X_{ki}}{n}$$

Step 3: We construct the mean matrix as follows

$$M_{np} = \begin{bmatrix} m_1 & m_2 & \cdots & m_p \\ m_1 & m_2 & \cdots & m_p \\ m_1 & m_2 & \cdots & m_p \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix}$$

Step 4: Subtract the mean matrix from the original data

$$X_0 = X - M$$

Step 5: Apply singular value decomposition (SVD) to get

$$SVD(x_0) = [U, E, coff]$$

and

$$X_0 = U * E * \overline{coff}$$

where coff is $p \times p$ matrix and it is the Principal components matrix. U is $n \times n$ matrix, and E is $n \times p$ matrix

Step 6: We calculate Eigen values from E as in the following equations

$$E = diag(E)$$

$$E = \frac{E}{\sqrt{n-1}}$$

$$eigenvalues = E^2$$

where Eigenvalues is the Eigen value vector and it is $n \times 1$

Step 7: We calculate the scores matrix which is $n \times p$ matrix

$$Scores = X_0 * coff$$

For our experiments we selected the first three components

$$Scores_{n\times 3} = \begin{bmatrix} s_{11} & s_{12} & S_{13} \\ s_{21} & s_{22} & S_{23} \\ \vdots & \vdots & \vdots \\ s_{n1} & s_{n2} & S_{n3} \end{bmatrix}$$

By applying PCA on this dataset we first generate the centred input around its column mean. Next, we applied Singular Value decomposition to derive eigenvectors and eigenvalues for the centred matrix, which is then rearranged as a new matrix starting with the eigenvector that corresponds to the highest eigenvalue, and so on. Accordingly, eigenvector matrix is $p \times p$ matrix, where p denotes the number of attributes (i.e number of characters 127). Finally, a scores matrix is calculated to produce a new dataset of $n \times p$ matrix.

The ultimate goal of applying PCA is to reduce the number of attributes (number of columns in the score matrix). The way the scores matrix is generated guarantees that around 95% of the data is represented by the first few columns. This is shown by Table 5.2 which demonstrates proportion of each eigenvalue of the total data.

Table 5.2: Eigenvalues for various principal components of Patient CU01

Principal	Eigenvalue	Proportion	Accumulated proportion
PC1	19030.24	79.90%	79.90%
PC2	3198.109	13.43%	93.33%
PC3	296.5323	1.25%	94.58%
PC4	257.7454	1.08%	95.66%
PC5	220.8355	0.93%	96.59%
PC6	184.5031	0.77%	97.36%
PC7	154.7827	0.65%	98.01%
PC8	143.3947	0.60%	98.61%
PC9	127.9148	0.54%	99.15%
PC10	107.755	0.45%	99.60%
PC11	94.87425	0.40%	100.00%

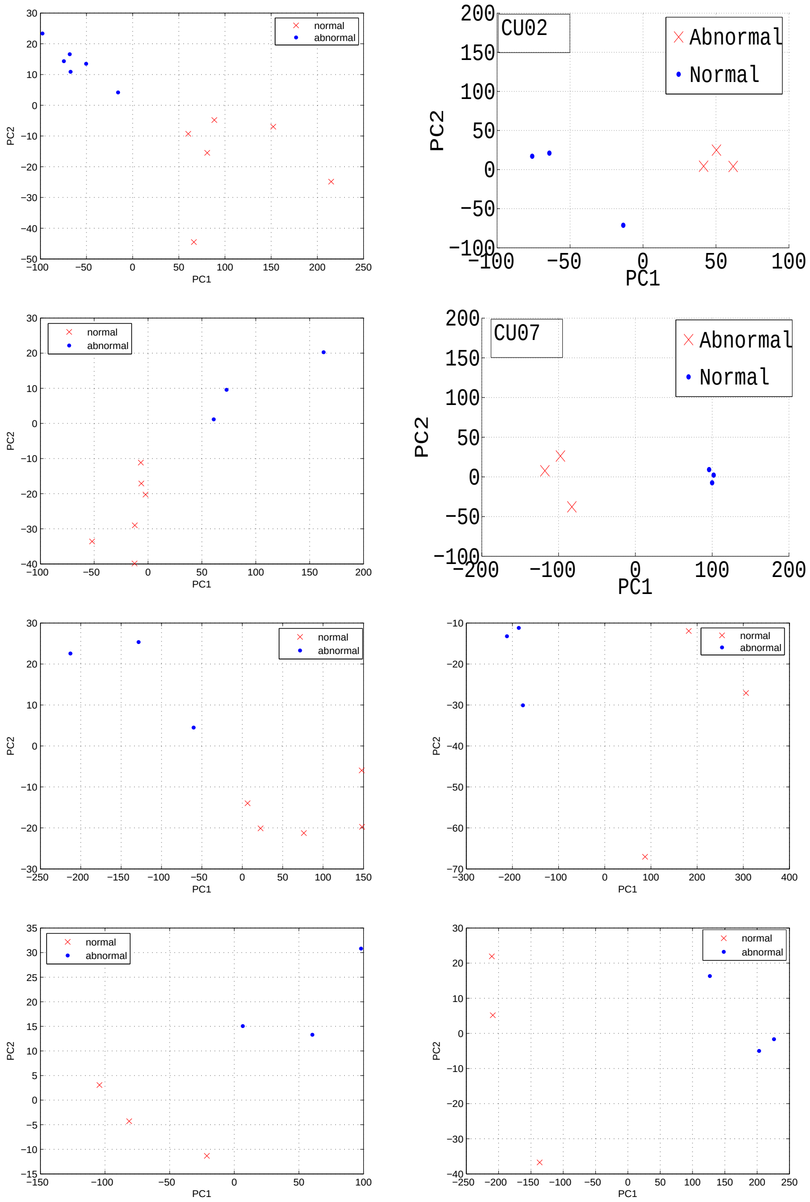


Figure 5.7: Class distribution for Principal Component 1 & 2 for patients cu01,cu02,cu03,cu07,cu09,cu10,cu1 and cu13 respectively

It is obvious from Table 5.2 how the first three eigenvalues contain approximately 94% of the total data. The calculation of eigenvalue proportion of the total data is performed by dividing the eigenvalue of the total summation of eigenvalues as shown in Eq. 5.1.

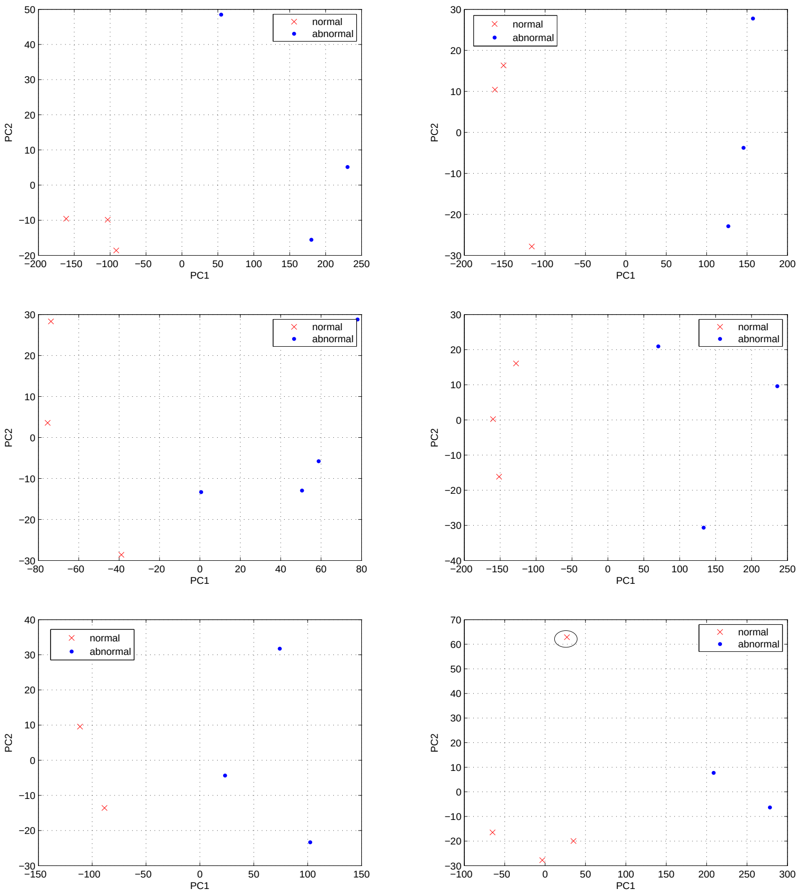


Figure 5.8: Class distribution for Principal Component 1 & 2 for patients cu15,cu16,cu17,cu18,cu19 and cu20 respectively

$$P_i = \frac{e_i}{\sum_{k=1}^{k=m} e_k} \tag{5.1}$$

where P_i is proportion of the i_{th} eigenvalue, e_i is the i_{th} eigenvalue and m number of eigenvalues which is the same number of variables.

In this proposed technique, the first three principal components and their corresponding scores are used to feed the neural network inputs in the classification stage. Therefore, the design of the neural network will be simpler. The same steps are repeated for the different patients and the first three components are taken.

Table 5.3: Scores for PC1, PC2 and PC3(Components 1 ,2 &3) of CU01

Sample No	Scores for PC1	Scores for PC2	Scores for PC3	Class
1	80.7423	-15.4819	25.7261	Abnormal
2	88.4103	-4.8090	-11.5453	Abnormal
3	60.1690	-9.2898	-1.8145	Abnormal
4	66.2988	-44.5062	10.92105	Abnormal
5	214.9726	-24.8546	-2.5129	Abnormal
6	152.2820	-6.9818	16.1482	Abnormal
7	-74.5081	14.3374	-14.1729	Normal
8	-97.7008	23.3456	5.14573	Normal
9	-68.1064	16.6038	-230418	Normal
10	-67.2722	10.8988	14.44182	Normal
11	-50.3783	13.4868	2.9953	Normal
12	-15.7506	4.1737	-11.3359	Normal

Table 5.4: K-mean results for CU01

no	Distance from Class 1	Distance from class 2	Class
1	21315.02	888.9972	Abnormal
2	23056	652.0261	Abnormal
3	15528.75	2601.073	Abnormal
4	19934.61	2672.954	Abnormal
5	46471.68	1861.369	Abnormal
6	78367.19	10970.73	Abnormal
7	149.6592	35243.77	Normal
8	1345.174	45019.88	Normal
9	41.69464	33066.42	Normal
10	33.32313	32410.83	Normal
11	141.8982	26844.89	Normal
12	2258.362	16410.42	Normal

5.5 Results

5.5.1 Ventricular Arrhythmia detection results

By applying the steps explained earlier Table 5.3 can be derived. It shows the first three PC scores of 12 ECG samples for patient CU01 of CU Ventricular Tacharrhythemia Database. The scores for other patients can be derived in the same manner. To demonstrate the results for different patients, Principal Component 1 verses Principle Component 2 is plotted for each of 14 patient testing bed as shown in Fig. 5.7 and Fig. 5.8. It is clear how the normal and abnormal ECGs can be distinguished and separated. This confirms that normal and abnormal ECG signal can be detected using PCA applied to compressed ECG signals.

In the clustering stage, two techniques are used and results are obtained for each. The first clustering technique used is k-means cluster. The number of inputs for the clustering k-means stage is three (i.e PC1,PC2,PC3). Table 5.4 demonstrates the classification results of k-means clustering for Patient CU01 using data shown in Table 5.3. In this table, samples

1-6 are classified as class 2(abnormal) because the distance to class 2 is less than the distance to class 1. On the other hand, samples 7-12 are classified as class 1(normal) because the distance to class 1 is less than the distance to class 2. K-means results for all patients are shown in Fig. 5.7, Fig. 5.8 and Table 5.10.

We applied neural networks instead of k-mean for data classification, to compare between both methods. The neural network is trained using 12 compressed ECG segments from CU01 (6 normal and 6 abnormal). The neural network architecture is shown in Fig. 5.9. The neural network consists of 3 inputs corresponding to the scores of PC1, PC2 and PC3. There are two outputs, their values and corresponding encodings are shown in Table 5.5.

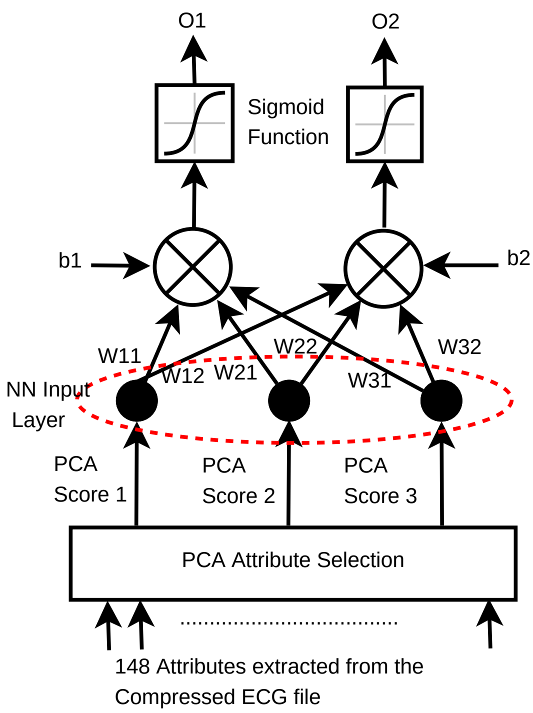


Figure 5.9: Neural Network Architecture

The developed Neural Network architecture is proposed to be simple to guarantee fast classification performance when applied in Big data. Therefore, the selected neural network

Table 5.5: Output values and their corresponding meanings

Output 1	Output 2	Class
1	1	Abnormal
0	0	Normal
1	0	Unknown
0	1	Unknown

uses Perceptron linear neural network with Sigmoid transfer function. The outputs equations of the proposed neural network are shown in Eq. 5.2 and Eq. 5.3.

$$O1 = LogSig(I1 \times W11 + I2 \times W21 + I3 \times W31 + b1) \tag{5.2}$$

$$O2 = LogSig(I1 \times W12 + I2 \times W22 + I3 \times W32 + b2) \tag{5.3}$$

where $LogSig(X)$ is the sigmoid activation function, W_{jk} is the weight value connecting input j with output k and b represents the bias values. The values of weights and biases are determined in the training phase.

The trained neural network could correctly detect all samples from 14 patients, thus 100% accuracy is achieved. On the other hand, the k-mean algorithm was capable of detecting samples correctly for all patients except for Patient CU20. This wrongly classified sample is notated by the circle in Fig. 5.8. The results for Neural Network Classification are shown in Fig. 5.10 which contains the classification line that resulted from the training phase of the neural network.

The training set and testing set details are shown in Table 5.6, in which the training set consists of 12% of the total dataset.

Table 5.6: Neural network Testing and training set distribution

Data set	Total	Normal	Abnormal	portion
Training	12	6	6	13%
Testing	83	42	41	87%

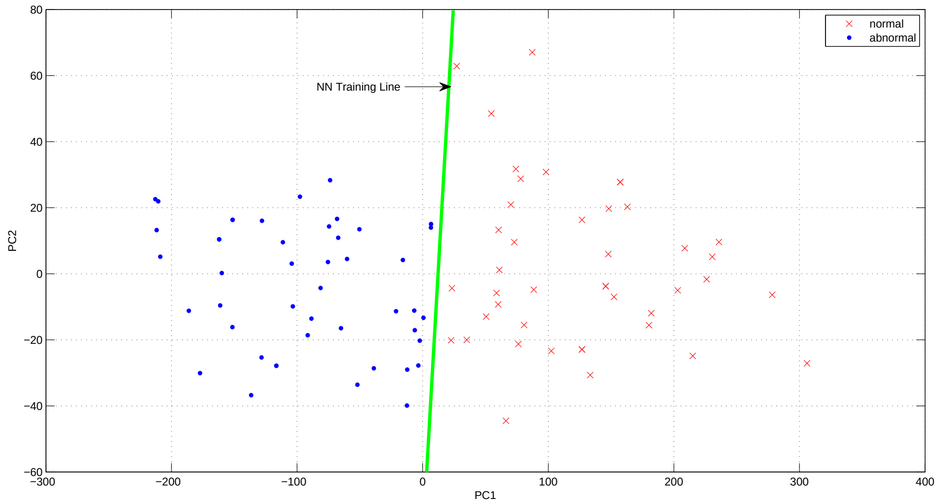


Figure 5.10: Neural Network classification

5.5.2 Detection of Left Bundle Branch Block

In order to prove the concept of detecting cardiac diseases from the compressed ECG signal, another disease (LBBB) is taken into consideration. However, this is not a life-threatening cardiac disease. From Test-bed of 27 samples collected from 5 different patients, 15 samples are LBBB and 12 samples are normal. The collected samples are downloaded from MIT-BIH Arrhythmia database. The same steps mentioned in Section 5.4 are repeated to differentiate between Normal and LBBB samples. The summarized k-means clustering results are illustrated in Table 5.7. The accuracy achieved for detecting LBBB is 96%. The k-means clustering is replaced by Neural Network with a training set and testing set specifications given in Table 5.8. In the case of LBBB the accuracy after using Neural Network did not improve and the Neural Network Classification results are shown in Fig. 5.11, the misclassified sample is surrounded by a circle in this figure.

Table 5.7: Clustering results for detecting LBBB

Aatient Number	*Number of Samples*	*Samples Class*	*Correctly Classified*	*Incorrectly Classified*
100	*9*	*Normal*	*9*	*0*
101	*3*	*Normal*	*2*	*1*
109	*6*	*LBBB*	*6*	*0*
111	*6*	*LBBB*	*6*	*0*
207	*3*	*LBBB*	*3*	*0*
total	*27*		*26*	*1*
Total Accuracy	*96%*			

Table 5.8: Training Set and Testing Set Details of the Neural Network

Data Set	*number of samples*	Normal	*LBBB*	*percentage*
Training	*9*	3	*6*	*33.33%*
Testing	*18*	9	*9*	*66.67%*

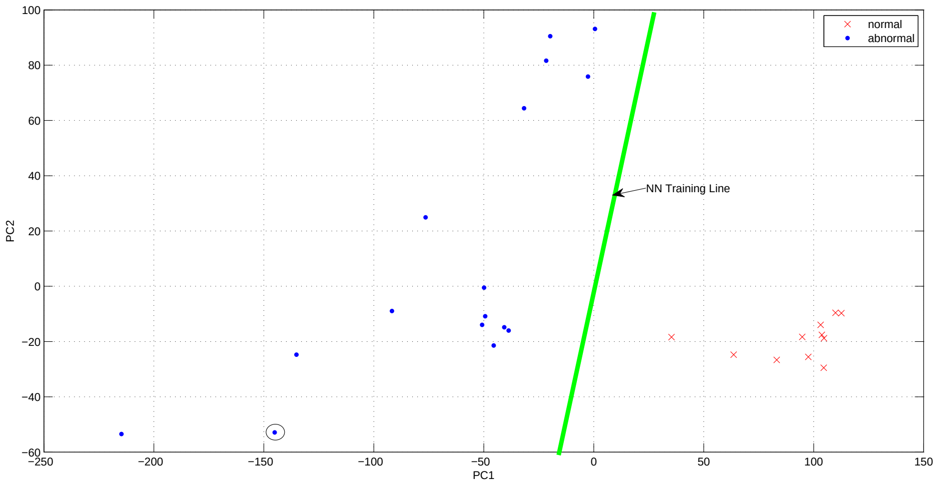


Figure 5.11: Neural Network classification for LBBB

5.5.3 Comparison with other Ventricular Arrhythmia diagnoses algorithms

The proposed algorithm is compared with other algorithms for classifying Tachyarrhythmia abnormalities(which was applied on the decompressed ECGs) such as, Neural Network with Weighted Fuzzy Membership function (NEWFM) algorithm, Generalized Regression Neural Network(GRNN) and Nearest Neighbour Method [3; 90]. Table 5.9 shows the results of sensitivity, specificity and efficiency for each algorithm. It clearly shows that our proposed method has the best efficiency although it is applied on the compressed ECG segments. Moreover, the GRNN Network will be complex since no feature selection algorithm is applied so, the number of neurons on the radial basis function layer of the GRNN will be large. The NEWFM algorithm uses wavelet transform, as well as, complex mathematical models to extract features used for learning the fuzzy neural network and is not performing well. Table 5.9 shows the achieved results of the above algorithms are not acceptable and need to be improved. There are other non-neural network approaches for detecting Ventricular Tachyarrhythmias such as, the work done by Nithish V Thakor [81]. They tested the algorithm on the same dataset used to generate the probability distribution. Xu-Sheng [88] adopted the complexity measure scheme. In this method the algorithm will produce poor results for small time window, therefore, it needs at least 7 seconds window to produce acceptable accuracy, moreover, the accuracy of their technique is based on the threshold technique that is inaccurate for different patients.

Table 5.9: Results Comparison between different algorithms for detecting Ventricular Tachyarrhythmia

Algorithm	Sensitivity	Specificity	Efficiency
NEWFM	81.2%	92.2%	90.1%
GRNN	79%	100%	87%
KNN	100%	76%	84%
PCA-kmean	100%	97.87%	98.9%
PCA-FNN	100%	100%	100%

5.6 Summary

Because ECG signal is enormous in size, compression algorithms must be used to make the whole tele-cardiology process faster and efficient to handle this big ECG data stored in the cloud. A faster solution is of crucial importance for diagnoses and treatment of cardiovascular diseases. Although ECG compression enables faster transmission, it also introduces a delay in the processing phase because of the decompression stage which makes it complex for the diagnosis process to be implemented in the cloud. Since existing methods process the original ECG signal and not the compressed one, this decompression time can be enough to threaten patient life and create additional processing overhead in the cloud. Moreover, cloud services should be able to detect ECG abnormality directly from the compressed ECG signals. To overcome the decompression delay and make body sensor network energy efficient, in this chapter we implemented the ECG analysis and data mining solution on the compressed ECG signal using PCA for feature extraction and k-mean as a clustering technique. Compressed ECG signal can be fast in transmission, and we have clearly shown we can classify and analyse the compressed ECG signal to detect cardiac Ventricular abnormalities as well as LBBB. Moreover, we developed a neural network model which was capable of detecting 100

% of the testing dataset for Ventricular Arrhythmia and 96 % for LBBB.

Table 5.10: K-Mean results for all patients

CU01			
Sample	distance from cluster 1	distance from cluster 2	Class
1	21315.02395	888.9972483	2
2	23055.99826	652.0260886	2
3	15528.75236	2601.072951	2
4	19934.6088	2672.954065	2
5	46471.68447	1861.369031	2
6	78367.19259	10970.7306	2
7	149.6591772	35243.76721	1
8	1345.174028	45019.88482	1
9	41.69463718	33066.42378	1
10	33.32313293	32410.83125	1
11	141.8982094	26844.8924	1
12	2258.362005	16410.42219	1

CU02			
Sample	Distance Class 1	distance class 2	class
1	17165.96	2656.824	2
2	10995.33	5041.396	2
3	14853.93	2331.761	2
4	287.1184	9101.931	1
5	177.8923	12989.94	1
6	256.3437	11614.72	1

cu03			
Sample	Distance Class 1	Distance class 2	class
1	679.6876299	8994.58139	1
2	1524.875217	6522.88547	1
3	4196.894779	33865.06271	1
4	14950.62389	224.6601263	2
5	24714.31148	1413.187568	2
6	11622.1543	271.6338424	2
7	11827.16756	147.4594099	2
8	13928.38301	24.03194365	2
9	11194.24977	193.7732326	2

CU07			
Sample	Distance Class 1	Distance class 2	class
1	47953.16245	1339.916885	2
2	40070.51666	1316.500561	2
3	34732.76505	1654.778195	2
4	175.9633572	39828.47468	1
5	86.19783535	38319.46551	1
6	46.60480176	40606.07433	1

cu09			
Sample	Distance Class 1	Distance class 2	class
1	4661.008	79822.77	1
2	4622.597	80883.33	1
3	5433.782	20694.87	1
4	3334.032	25889.14	1
5	42.3954	45579.49	1
6	87404.41	6271.713	2
7	20163.22	5582.137	2
8	45243.43	91.74639	2

cu10			
Sample	Distance Class 1	Distance class 2	class
1	80226.66	11925.82	2
2	139704.3	643.8511	2
3	247867.1	13129.55	2
4	352.7296	136169.5	1
5	79.48856	143426.5	1
6	427.4234	163362.6	1

cu11			
Sample	Distance Class 1	Distance class 2	class
1	70.20876938	17038.52367	1
2	2368.273125	6076.25839	1
3	1977.463725	29127.92022	1
4	19168.57572	153.1096829	2
5	6779.808085	2328.0955	2
6	25661.46022	1301.882189	2

cu13			
Sample	Distance Class 1	Distance class 2	class
1	97829.88	3593.096	2
2	169258	1684.347	2
3	150857.6	382.4457	2
4	1268.677	157070.9	1
5	3506.079	105187.4	1
6	626.3195	155428.3	1

cu15			
Sample	Distance Class 1	Distance class 2	class
1	33782.25	11380.15	2
2	122283.7	5746.04	2
3	89336.55	1425.905	2
4	243.7278	67308.1	1
5	1819.489	100575	1
6	774.0833	61804.59	1

cu16			
Sample	Distance Class 1	Distance class 2	class
1	83514.93936	22.88432729	2
2	73473.92797	811.3460046	2
3	91158.61776	947.7034203	2
4	1474.028801	68263.4493	1
5	343.6322637	87066.30807	1
6	467.8779127	93321.33294	1

cu17			
Sample	Distance Class 1	Distance class 2	class
1	14803.24	163.8341	2
2	20575.7	1841.06	2
3	13015.89	159.6142	2
4	4212.101	2308.955	2
5	863.2899	15428.44	1
6	1447.313	8150.374	1
7	167.7911	14999.68	1

cu18			
Sample	Distance Class 1	Distance class 2	class
1	47358.1	6261.362	2
2	79234.48	1109.328	2
3	146253.4	8085.473	2
4	287.7555	89032.51	1
5	182.2532	93886.83	1
6	600.0694	75540.59	1

cu19			
Sample	Distance Class 1	Distance class 2	class
1	1889.705866	41399.69573	1
2	980.8189747	31453.171	1
3	24311.59262	263.1239886	2
4	31729.55609	263.1239886	2
5	1908.461084	15198.47031	1

cu20			
Sample	Distance Class 1	Distance class 2	class
1	1261.657	44274.11	1
2	50706.92	4815.929	2
3	1261.657	78373.91	1
4	61788.17	755.3707	2
5	95489.13	4288.047	2
6	43855.68	1731.154	2

Chapter 6

Conclusion

In this chapter, a full summary of the thesis will be explained, focusing on the thesis' aims and research questions. Contributions regarding these research questions will be summarized with key findings of this thesis. Section 6.1 explains the research aims and restates the research question we answer in this thesis. Section 6.2 illustrates the contributions and explains how the research questions are answered. Section 6.3 lists some key facts discovered in this research. Finally, Section 6.4 discusses how the current work could be further developed.

6.1 Research Aims

This thesis proposes an efficient cardiac monitoring system that can securely transmit and store patient information and ECG signals in Cloud. Moreover, the proposed cardiac monitoring system should be able to provide fast and accurate diagnosis reports, especially for life-threatening cardiac diseases such as ventricular diseases. According to the Australian Bureau of Statistics, 3.5 million Australians suffer from long term cardiac diseases. Therefore, such sufferers should be continuously monitored. However, dealing with a continuous stream of ECG signals for 3.5 million patients is beyond the capability of current health care providers. Cloud services provide a solution to this problem. Furthermore, health-care experts have several concerns preventing them from using cloud services, most of which relate to patient privacy and HIPAA policies. Therefore, in this thesis, our proposed monitoring system will take into consideration these concerns. This thesis answers the following research questions.

1. **How patients' confidential information can be protected and transmitted securely in wireless tele-cardiovascular monitoring systems?**

2. **How can ECG signal self similarity characteristics be utilized to produce a powerful lossy compression algorithm that can be imblemented either in the client side or the cloud side?**

3. **How to utilize ECG morphological characteristics to implement a lossless compression algorithm that can be used for compressing ECG signal and keeping the resultant compressed file usable for diagnoses without decompression?**

4. **How can cardiac diseases be diagnosed directly from compressed ECG using the proposed lossless compression algorithms?**

6.2 Research Contributions

To answer the research questions originally posed in Section 1.2, the following contributions are introduced in this thesis:

1. **Proposing two new ECG steganography techniques for protecting patient confidential information**

   To protect patient confidential information, steganography technique is utilized. Two steganography techniques have been proposed. Time domain steganography is based on special range transform. In this technique the distortion is low but the amount of information that can be hidden is also low. The second technique is a frequency domain wavelet based steganography, which combines encryption and scrambling technique to protect patient confidential data. The proposed method allows ECG signal to be a carrier for the patient confidential data and other physiological information such as Blood pressure, Glucose level, temperature,etc. Mathematical analysis proved the technique to be highly secure. The distortion and diagnosability of the watermarked ECG were also found to be minimal using Percentage Residual Difference (PRD) and Weighted Wavelet PRD (WWPRD). Finally, we found that using our proposed wavelet based steganography technique, it is possible to store about 2.4KB data inside ECG signal of 10 seconds length and 360 samples/s sampling rate, with PRD of less than 1%.

2. **Proposing a new cloud enabled fractal based ECG lossy compression**

   In this thesis, a fractal model is utilized to make use of the ECG signal self similarity characteristics to develop a lossy compressed technique with high compression ratios and low information loss. The proposed technique is based on modifying the popular fractal model to be used in compression using the concept of Iterated Function System. The basic idea of this technique is to divide the ECG signal into equal blocks called range blocks. Subsequently, another down-sampled copy of the ECG signal is created which is called domain. For each range block, the most similar block in the domain is found by calculating fractal RMS and fractal coefficients. Accordingly, for each range block, the previously determined fractal coefficients (Scale, Offset, Affine code) as well as domain block index are stored within the compressed file. In order to make our technique cloud enabled, the decompression operation is designed in such a way that allows the user to retrieve part of the file(i.e ECG segment) without decompressing the whole file. Therefore, it is not necessary to download the full compressed file before the user can view the resultant ECG signal. Furthermore, the proposed technique is designed to guarantee that, it is suitable to be implemented inside a parallel processing environment such as cloud, without adding more overheads in the communication between the processing units. The proposed algorithm has been implemented and compared with other existing lossy ECG compression techniques. It is found that the proposed technique can achieve higher compression ratio of 40 and lower Percentage Residual Difference (PRD) Value of less than 1 %.

The proposed fractal compression technique performs poorly if it is implemented in a single processing mode. Therefore, to improve its performance and guarantee it is implementable inside a client device (which may have limited processing resources) we modified the current fractal model and proposed a new mathematical derivation to propose a new, fast fractal ECG lossy compression technique. The effect of using both methods on the ECG signal is calculated using a PRD measurement. The performance of both methods in terms of processing time is shown. It is found that using the same data set and the same processing power fast fractal is 35 times faster than normal fractal compression. However, using fast fractal doubles the distortion of the resultant ECG signal of the normal fractal.

3. **Proposing new ECG lossless compression using Gaussian-based approximation and Burrow-Wheeler Transform**

   The shape of the ECG signal consists from many Gaussian functions with different parameters. Therefore, this thesis proposes a new Gaussian-based ECG lossless compression technique. A Gaussian function can be used to model and approximate an ECG signal. Therefore, optimization theory is used to determine the best Gaussian function parameters for ECG approximation. Residuals are then calculated and differentiated. Subsequently, the residuals are encoded using the Burrow-Wheeler transform (BWT), followed by Move-to-Front (MTF) and run-length encoding. Finally, the resultant encoded signal is further encoded using Huffman coding. BWT and MTF encoding are modified to deal with numbers instead of dealing with strings. The proposed algorithm has been implemented and compared with other lossless ECG compression techniques. The proposed technique is found to achieve an average higher compression ratio of 4.09 with guaranteed exact reconstruction of the ECG signal. The processing of the signal in time domain ensures that the features and sequence of the ECG signal points will stay in the same sequence. As a result, the compressed file will be highly correlated with the original ECG signal. Accordingly, the proposed compression technique is not only a normal compression technique, but it also provides the capability for diagnosing the compressed ECG file directly without performing the decompression stage.

4. **Proposing new cardiac diseases detection algorithms from the compressed ECG signal using Principale Components Analyses and neural networks** Using our proposed Gaussian-based lossless compression technique, this thesis introduces a new diagnoses technique to detect ventricular arrhythmia as well as LBBB directly from a compressed ECG signal. The proposed algorithm uses a PCA technique for feature extraction and k-mean for clustering normal and abnormal ECG signals. Further, the k-mean algorithm is replaced with a Perceptron-Neural Network to improve the accuracy of the system. The diagnosis process is implemented in the cloud to enable diagnoses of the compressed ECG files which are already stored in the cloud. The achieved diagnosis accuracy is 100% for ventricular arrhythmia using neural network and about 96% for detecting LBBB.

6.3 Key Facts

This thesis provides a number of new key facts that can be concluded and summarized as follows:

1. The continuity of the ECG signal provides a key feature that makes it suitable to be a reliable carrier for other information. Therefore, an ECG signal is used as a host signal for our proposed steganography model. Moreover, the resultant ECG can still be

used for diagnostic purposes according to our findings, using mathematical distortion measurement as well as experts' opinions.

2. The ECG signal self-similarity is a feasible feature that can be utilized for compression. Fractal provides the mathematical model to represent these self-similar objects. However, fractals are proposed to represent self-similar images. Accordingly, the fractal model can be modified to be used for modeling an ECG signal. Therefore, fractal compression proposed in this thesis could achieve a higher compression ratio.

3. The fractal model is a very flexible mathematical model which can be modified for performance improvements as shown in this thesis.

4. Homomorphic encryption is a tool that provides the capability to perform mathematical operations on the encrypted data. This thesis proves that homomorphic lossless ECG compression is achievable to provide not only size reduction but also compressed data which can be analysed and diagnosed. This is because the Gaussian function model provides reliable modeling for the ECG signal.

5. Normal diagnostics techniques are based on diagnosing the raw ECG signal. However, this thesis has shown that diagnosing compressed ECG directly without decompression is achievable with very encouraging results.

6. PCA dimensionality reduction is a powerful technique that will eliminate redundant data and keep the useful features.

6.4 Work limitations and Future Work

In this thesis we have tried to solve some problems that prevent health-care providers from utilizing the powerful cloud services in their systems and especially for remote cardiac monitoring systems. However, the proposed solutions provide small steps toward the final solution. In this section we summarise some limitations of the proposed techniques and provide some suggestions to overcome these limitations through future research.

- In Chapter 2, the steganography technique is used to protect patient confidential data during transmission to and storage in cloud. The proposed technique provided an authentication feature to prevent unauthorized persons from extracting or accessing patient confidential information. The technique is based on a shared key security which will provide access to those who have the key and the scrambling matrix. However, this approach is weak if more than one person requires access to the same patient information. In some circumstances, one doctor may need to extract one part of a patient's health record and another doctor extracts different parts of the same record. Therefore, the use of the steganography technique is required as the lower layer of a complex access control framework that can provide role based access control and other security features to protect confidential data.
- In Chapter 3, a fractal model showed very encouraging results in terms of compression ratio. However, the proposed fractal compression is lossy compression. It is possible to improve the fractal model to convert it from lossy compression to a lossless compression technique.

- Chapter 4 proposed the use of Gaussian-based lossless compression, in which different encoding algorithms are used. However, the limitation of our proposed technique is that Gaussian parameters evaluation using a Trust region optimization technique should be performed for each ECG signal. This creates a processing overhead, so a predetermined Gaussian parameter is required that can be unified for all ECG signals - or at least to be used as an initial value for the optimization technique to improve the compression performance.

- In Chapter 5, diagnoses from the compressed ECG signal is implemented. However, the proposed model deals with different diseases separately. Therefore, the training and testing for the neural network for detecting Ventricular Arrhythmia is done separately from the training and testing of the neural network for detecting LBBB. Accordingly, our model should be developed to enable differentiation between different arrhythmia. Moreover, other diseases such as PVC, ST segment Elevation and RBBB should be added to our datasets to guarantee that our model can diagnose and distinguish between them.

Bibliography

[1] Physiobank, physiotoolkit, and physionet: Components of a new research resource for complex physiologic signals.

[2] Australian bureau of statistics - 4821.0.55.001 - cardiovascular disease in australia: A snapshot, 2004-05. URL "http://www.abs.gov.au/ausstats/abs@.nsf/mf/4821.0.55.001".

[3] R. Abbas, W. Aziz, and M. Arif. Prediction of ventricular tachyarrhythmia in electrocardiograph signals using neural network and modified nearest neighbour method. In *Engineering, Sciences and Technology, Student Conference On*, pages 1–6, Dec. 2004.

[4] M. Abo-Zahhad, A. F. Al-Ajlouni, S. M. Ahmed, and R. Schilling. A new algorithm for the compression of {ECG} signals based on mother wavelet parameterization and best-threshold levels selection. *Digital Signal Processing*, 23(3):1002 – 1011, 2013. ISSN 1051-2004.

[5] A. Al-Fahoum. Quality assessment of ECG compression techniques using a wavelet-based

diagnostic measure. *IEEE Transactions on Information Technology in Biomedicine*, 10 (1), 2006.

[6] D. Al-Shammary and I. Khalil. Dynamic fractal clustering technique for soap web messages. In *IEEE International Conference on Services Computing (SCC),*, pages 96–103. IEEE, 2011.

[7] Z. Arnavut. Ecg signal compression based on burrows-wheeler transformation and inversion ranks of linear prediction. *IEEE Transactions on Biomedical Engineering*, 54 (3):410–418, 2007.

[8] F. Bellifemine, G. Fortino, R. Giannantonio, R. Gravina, A. Guerrieri, and M. Sgroi. Spine: a domain-specific framework for rapid prototyping of wbsn applications. *Software: Practice and Experience*, 41(3):237–265, 2011.

[9] A. Bilgin, M. Marcellin, and M. Altbach. Compression of electrocardiogram signals using jpeg2000. *Consumer Electronics, IEEE Transactions on*, 49(4):833–840, 2003.

[10] M. Branch, T. Coleman, and Y. Li. A subspace, interior, and conjugate gradient method for large-scale bound-constrained minimization problems. *SIAM Journal on Scientific Computing*, 21(1):1–23, 1999.

[11] M. Burrows and D. Wheeler. A block-sorting lossless data compression algorithm. 1994.

[12] G. Chang and Y. Lin. An efficient lossless ecg compression method using delta coding and optimal selective huffman coding. In C. Lim and J. Goh, editors, *6th World Congress*

of Biomechanics (WCB 2010). August 1-6, 2010 Singapore, volume 31 of *IFMBE Proceedings*, pages 1327–1330. Springer Berlin Heidelberg, 2010. ISBN 978-3-642-14514-8.

[13] W. Chen, L. Hsieh, and S. Yuan. High performance data compression method with pattern matching for biomedical ecg and arterial pulse waveforms. *Computer methods and programs in biomedicine*, 74(1):11–27, 2004.

[14] H.-H. Chou, Y.-J. Chen, Y.-C. Shiau, and T.-S. Kuo. An effective and efficient compression algorithm for ecg signals with irregular periods. *Biomedical Engineering, IEEE Transactions on*, 53(6):1198–1205, 2006.

[15] H.-H. Chou, Y.-J. Chen, Y.-C. Shiau, and T. son Kuo. An effective and efficient compression algorithm for ecg signals with irregular periods. *IEEE Transactions on Biomedical Engineering*, 53(6):1198 –1205, june 2006. ISSN 0018-9294. doi: 10.1109/TBME.2005.863961.

[16] G. Clifford, F. Azuaje, and P. McSharry. *Advanced methods and tools for ECG data analysis*. Artech House.

[17] A. De la Rosa Algarin, S. Demurjian, S. Berhe, and J. Pavlich-Mariscal. A security framework for xml schemas and documents for healthcare. In *Bioinformatics and Biomedicine Workshops (BIBMW), 2012 IEEE International Conference on*, pages 782–789, 2012.

[18] E. Deelman, G. Singh, M. Livny, B. Berriman, and J. Good. The cost of doing science on the cloud: the montage example. In *Proceedings of the 2008 ACM/IEEE conference*

on Supercomputing, SC 08, pages 50:1–50:12. IEEE Press, Piscataway, NJ, USA, 2008. ISBN 978-1-4244-2835-9. URL http://dl.acm.org/citation.cfm?id=1413370.1413421.

[19] M. Diaz, G. Juan, O. Lucas, and A. Ryuga. Big data on the internet of things: An example for the e-health. In *Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2012 Sixth International Conference on*, pages 898–900, 2012.

[20] K. Duda, P. Turcza, and T. Zielinski. Lossless ecg compression with lifting wavelet transform. In *Instrumentation and Measurement Technology Conference, 2001. IMTC 2001. Proceedings of the 18th IEEE*, volume 1, pages 640–644. IEEE, 2001.

[21] M. Dunham. *Data mining introductory and advanced topics.* Prentice Hall/Pearson Education, 2003.

[22] C. Fira and L. Goras. An ecg signals compression method and its validation using nns. *IEEE Transactions on Biomedical Engineering*, 55(4):1319–1326, 2008.

[23] G. Fortino, M. Pathan, and G. Di Fatta. Bodycloud: Integration of cloud computing and body sensor networks. In *IEEE 4th International Conference on Cloud Computing Technology and Science (CloudCom)*, pages 851–856. IEEE, 2012.

[24] G. Fortino, R. Giannantonio, R. Gravina, P. Kuryloski, and R. Jafari. Enabling effective programming and flexible management of efficient body sensor network applications. *IEEE Transactions on Human-Machine Systems*, 43(1):115–133, 2013.

[25] A. Ghaffari, H. Golbayani, and M. Ghasemi. A new mathematical based QRS detector

using continuous wavelet transform. *Computers and Electrical Engineering*, 34(2):81–91, 2008.

[26] H. Golpira and H. Danyali. Reversible blind watermarking for medical images based on wavelet histogram shifting. In *IEEE International Symposium on Signal Processing and Information Technology (ISSPIT), 2009*, pages 31–36. IEEE, 2010.

[27] R. N. Horspool and W. J. Windels. An lz approach to ecg compression. In *Computer-Based Medical Systems, 1994., Proceedings 1994 IEEE Seventh Symposium on*, pages 71–76. IEEE, 1994.

[28] F. Hu, M. Jiang, M. Wagner, and D. Dong. Privacy-preserving telecardiology sensor networks: toward a low-cost portable wireless hardware/software codesign. *IEEE Transactions on Information Technology in Biomedicine,*, 11(6):619–627, 2007.

[29] B. Huang, Y. Wang, and J. Chen. 2-d compression of ecg signals using roi mask and conditional entropy coding. *IEEE Transactions on Biomedical Engineering*, 56(4):1261–1263, 2009.

[30] B. Huang, Y. Wang, and J. Chen. Ecg compression using the context modeling arithmetic coding with dynamic learning vector–scalar quantization. *Biomedical Signal Processing and Control*, 2012.

[31] D. Huffman. A method for the construction of minimum-redundancy codes. *Proceedings of the IRE*, 40(9):1098–1101, 1952.

[32] A. Ibaida and I. Khalil. Wavelet based ecg steganography for protecting patient confiden-

tial information in point-of-care systems. *IEEE transactions on bio-medical engineering*, 2013.

[33] A. Ibaida, I. Khalil, and F. Sufi. Cardiac abnormalities detection from compressed ECG in wireless telemonitoring using principal components analysis (PCA). In *5th International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), 2009*, pages 207–212. IEEE, 2010.

[34] A. Iosup, S. Ostermann, M. N. Yigitbasi, R. Prodan, T. Fahringer, and D. H. Epema. Performance analysis of cloud computing services for many-tasks scientific computing. *IEEE Transactions on Parallel and Distributed Systems*, 22(6):931–945, 2011.

[35] R. Istepanian, L. Hadjileontiadis, and S. Panas. Ecg data compression using wavelets and higher order statistics methods. *IEEE Transactions on Information Technology in Biomedicine*, 5(2):108–115, 2001.

[36] K. R. Jackson, L. Ramakrishnan, K. Muriki, S. Canon, S. Cholia, J. Shalf, H. J. Wasserman, and N. J. Wright. Performance analysis of high performance computing applications on the amazon web services cloud. In *IEEE Second International Conference on Cloud Computing Technology and Science (CloudCom)*, pages 159–168. IEEE, 2010.

[37] I. Jekova. Comparison of five algorithms for the detection of ventricular fibrillation from the surface ecg. *Physiological measurement*, 21(4):429, 2000.

[38] N. Johnson and S. Jajodia. *Information hiding: steganography and watermarking: attacks and countermeasures*, volume 1. Springer, 2001.

[39] I. Jolliffe. *Principal component analysis.* Springer verlag, 2002.

[40] S. Kaur, R. Singhal, O. Farooq, and B. Ahuja. Digital Watermarking of ECG Data for Secure Wireless Commuication. In *2010 International Conference on Recent Trends in Information, Telecommunication and Computing*, pages 140–144. IEEE, 2010.

[41] A. Khalaj and H. M. Naimi. New efficient fractal based compression method for electrocardiogram signals. In *Canadian Conference on Electrical and Computer Engineering. CCECE09*, pages 983–986. IEEE, 2009.

[42] I. Khalil and F. Sufi. Legendre Polynomials based biometric authentication using QRS complex of ECG. In *Intelligent Sensors, Sensor Networks and Information Processing, 2008. ISSNIP 2008. International Conference on*, pages 297–302, 2008.

[43] H. Kim, R. Yazicioglu, P. Merken, C. Van Hoof, and H. Yoo. Ecg signal compression and classification algorithm with quad level vector for ecg holter system. *Information Technology in Biomedicine, IEEE Transactions on*, 14(1):93–100, 2010.

[44] T. Kim, S. Kim, J. Kim, B. Yun, and K. Park. Curvature based ecg signal compression for effective communication on wpan. *Communications and Networks, Journal of*, 14 (1):21–26, 2012.

[45] C. Ku, K. Hung, T. Wu, and H. Wang. Wavelet-based ecg data compression system with linear quality control scheme. *IEEE Transactions on Biomedical Engineering*, 57 (6):1399–1409, 2010.

[46] H. Lee and K. Buckley. Ecg data compression using cut and align beats approach and 2-d transforms. *IEEE Transactions on Biomedical Engineering,*, 46(5):556–564, 1999.

[47] S. Lee, J. Kim, and M. Lee. A real-time ecg data compression and transmission algorithm for an e-health device. *IEEE Transactions on Biomedical Engineering*, 58(9):2448–2455, 2011.

[48] W. Lee and C. Lee. A cryptographic key management solution for hipaa privacy/security regulations. *IEEE Transactions on Information Technology in Biomedicine,*, 12(1):34–41, 2008.

[49] M. Li, S. Yu, Y. Zheng, K. Ren, and W. Lou. Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption. *Parallel and Distributed Systems, IEEE Transactions on*, 24(1):131–143, 2013. ISSN 1045-9219. doi: 10.1109/TPDS.2012.97.

[50] C.-T. Lin, K.-C. Chang, C.-L. Lin, C.-C. Chiang, S.-W. Lu, S.-S. Chang, B.-S. Lin, H.-Y. Liang, R.-J. Chen, Y.-T. Lee, and L.-W. Ko. An intelligent telecardiology system using a wearable and wireless ecg to detect atrial fibrillation. *Information Technology in Biomedicine, IEEE Transactions on*, 14(3):726–733, 2010.

[51] Y. Lin, I. Jan, P. Ko, Y. Chen, J. Wong, and G. Jan. A wireless PDA-based physiological monitoring system for patient transport. *IEEE Transactions on information technology in biomedicine*, 8(4):439–447, 2004.

[52] B. Lo, S. Thiemjarus, R. King, and G.-Z. Yang. Body sensor network-a wireless sensor

platform for pervasive healthcare monitoring. In *The 3rd International Conference on Pervasive Computing*, volume 13, pages 77–80, 2005.

[53] H. Löhr, A.-R. Sadeghi, and M. Winandy. Securing the e-health cloud. In *Proceedings of the 1st ACM International Health Informatics Symposium*, pages 220–229. ACM, 2010.

[54] Z. Lu, D. Kim, and W. Pearlman. Wavelet compression of ECG signals by the set partitioning in hierarchical trees (SPIHT) algorithm. *IEEE Trans. Biomed. Eng*, 47(7): 849–856, 2000.

[55] A. Luthra. *ECG made easy.* JP Medical Ltd, 2011.

[56] A. Luthra. *ECG made easy.* Jaypee Brothers Medical Pub, 2012.

[57] I. Maglogiannis, L. Kazatzopoulos, K. Delakouridis, and S. Hadjiefthymiades. Enabling location privacy and medical data encryption in patient telemonitoring systems. *IEEE Transactions on Information Technology in Biomedicine,*, 13(6):946–954, 2009.

[58] S. Mahmoodabadi, A. Ahmadian, and M. Abolhasani. ECG feature extraction using Daubechies wavelets. In *Proceedings of the Fifth IASTED International Conference, Visualization, Imaging, and Image Processing, Benidorm, Spain*, 2005.

[59] K. Malasri and L. Wang. Addressing security in medical sensor networks. In *Proceedings of the 1st ACM SIGMOBILE international workshop on Systems and networking support for healthcare and assisted living environments*, page 12. ACM, 2007.

[60] H. Mamaghanian, N. Khaled, D. Atienza, and P. Vandergheynst. Compressed sensing

for real-time energy-efficient ecg compression on wireless body sensor nodes. *IEEE Transactions on Biomedical Engineering*, 58(9):2456–2466, 2011.

[61] L. Marvel, C. Boncelet, and C. Retter. Spread spectrum image steganography. *IEEE Transactions on Image Processing*, 8(8):1075–1083, 1999.

[62] S. Miaou and S. Chao. Wavelet-based lossy-to-lossless ecg compression in a unified vector quantization framework. *Biomedical Engineering, IEEE Transactions on*, 52(3): 539–543, 2005.

[63] K.-i. Minami, H. Nakajima, and T. Toyoshima. Real-time discrimination of ventricular tachyarrhythmia with fourier-transform neural network. *Biomedical Engineering, IEEE Transactions on*, 46(2):179–185, 1999.

[64] G. Moody and R. Mark. The impact of the mit-bih arrhythmia database. *Engineering in Medicine and Biology Magazine, IEEE*, 20(3):45–50, 2001.

[65] S. Mukhopadhyay, S. Mitra, and M. Mitra. A lossless ecg data compression technique using ascii character encoding. *Computers & Electrical Engineering*, 37(4):486 – 497, 2011. ISSN 0045-7906.

[66] M. Naghavi, P. Libby, E. Falk, S. W. Casscells, S. Litovsky, J. Rumberger, J. J. Badimon, C. Stefanadis, P. Moreno, G. Pasterkamp, et al. From vulnerable plaque to vulnerable patient a call for new definitions and risk assessment strategies: part i. *Circulation*, 108 (14):1664–1672, 2003.

[67] M. Nambakhsh, A. Ahmadian, M. Ghavami, R. Dilmaghani, and S. Karimi-Fard. A

novel blind watermarking of ecg signals on medical images using ezw algorithm. In *Engineering in Medicine and Biology Society, 2006. EMBS '06. 28th Annual International Conference of the IEEE*, pages 3274–3277, 2006.

[68] D. O'Leary. Artificial intelligence and big data. *Intelligent Systems, IEEE*, 28(2):96–99, 2013.

[69] S. Ostermann, A. Iosup, N. Yigitbasi, R. Prodan, T. Fahringer, and D. Epema. A performance analysis of ec2 cloud computing services for scientific computing. In D. Avresky, M. Diaz, A. Bode, B. Ciciani, and E. Dekel, editors, *Cloud Computing*, volume 34 of *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering*, pages 115–131. Springer Berlin Heidelberg, 2010. ISBN 978-3-642-12635-2.

[70] S. Pandey, W. Voorsluys, S. Niu, A. Khandoker, and R. Buyya. An autonomic cloud environment for hosting ecg data analysis services. *Future Generation Computer Systems*, 28(1):147–154, 2012.

[71] PhysioNet. The creighton university ventricular tachyarrhythmia database.

[72] A. Poularikas. *Transforms and Applications Handbook*. CRC, 2009.

[73] N. Rodrigues, E. da Silva, S. de Faria, V. da Silva, M. de Carvalho, et al. Ecg signal compression based on dc equalization and complexity sorting. *IEEE Transactions on Biomedical Engineering*, 55(7):1923–1926, 2008.

[74] D. Salomon. *Data compression: the complete reference*. Springer-Verlag New York Incorporated, 2004.

[75] D. Stinson. *Cryptography: theory and practice*. CRC press, 2006. ISBN 1584885084.

[76] F. Sufi and I. Khalil. An automated patient authentication system for remote telecardiology. In *Intelligent Sensors, Sensor Networks and Information Processing, 2008. ISSNIP 2008. International Conference on*, pages 279–284, 2008.

[77] F. Sufi and I. Khalil. Enforcing secured ECG transmission for real time telemonitoring: A joint encoding, compression, encryption mechanism. *Security and Communication Networks*, 1(5), 2008.

[78] F. Sufi, I. Khalil, and I. Habib. Polynomial distance measurement for ECG based biometric authentication. John Wiley & Sons, Ltd. Chichester, UK, 2008.

[79] Y. Suzuki. Self-organizing QRS-wave recognition in ECG using neural networks. *IEEE Transactions on Neural Networks*, 6(6):1469–1477, 1995.

[80] S. Tai, C. Sun, and W. Yan. A 2-d ecg compression method based on wavelet transform and modified spiht. *Biomedical Engineering, IEEE Transactions on*, 52(6):999–1008, 2005.

[81] N. Thakor, Y. Zhu, and K. Pan. Ventricular tachycardia and fibrillation detection by a sequential hypothesis testing algorithm. *IEEE Transactions on Biomedical Engineering*, 37(9):837–843, 1990.

[82] H. Wang, D. Peng, W. Wang, H. Sharif, H. Chen, and A. Khoynezhad. Resource-aware secure ecg healthcare monitoring through body sensor networks. *Wireless Communications, IEEE*, 17(1):12–19, 2010.

[83] X. Wang, Q. Gui, B. Liu, Z. Jin, and Y. Chen. Enabling smart personalized healthcare: a hybrid mobile-cloud approach for ecg telemonitoring, 2013.

[84] J. Wei, C. Chang, N. Chou, and G. Jan. Ecg data compression using truncated singular value decomposition. *Information Technology in Biomedicine, IEEE Transactions on*, 5 (4):290–299, 2001.

[85] M. Wigan and R. Clarke. Big data's big unintended consequences. *Computer*, 46(6): 46–53, 2013. ISSN 0018-9162. doi: 10.1109/MC.2013.195.

[86] C. Xu. *Encyclopedia of Optimization*. Springer US, 2009. ISBN 978-0-387-74758-3.

[87] P. Yip. *The transform and data compression handbook*. CRC, 2001.

[88] X. Zhang, Y. Zhu, N. Thakor, and Z. Wang. Detecting ventricular tachycardia and fibrillation by complexity measure. *IEEE Transactions on Biomedical Engineering*, 46 (5):548–555, 1999.

[89] Y. Zhang and Y. Zhang. An ecg compression algorithm based on gradient difference for body sensor networks. In *Biomedical Engineering and Biotechnology (iCBEB), 2012 International Conference on*, pages 712–715. IEEE, 2012.

[90] Z.-X. Zhang, S.-H. Lee, H. Jang, and J. Lim. Detecting ventricular arrhythmias by

newfm. In *Granular Computing, 2008. GrC 2008. IEEE International Conference on*, pages 822–825, Aug. 2008. doi: 10.1109/GRC.2008.4664646.

[91] Z.-X. Zhang, S.-H. Lee, H. J. Jang, and J. S. Lim. Detecting ventricular arrhythmias by newfm. In *Granular Computing, 2008. GrC 2008. IEEE International Conference on*, pages 822–825. IEEE, 2008.

[92] K. Zheng and X. Qian. Reversible Data Hiding for Electrocardiogram Signal Based on Wavelet Transforms. In *International Conference on Computational Intelligence and Security, 2008. CIS'08*, volume 1, 2008.

[93] Q. Zhou. Study on ecg data lossless compression algorithm based on k-means cluster. In *Future Computer and Communication, 2009. FCC'09. International Conference on*, pages 91–93. IEEE, 2009.