Optimizing Real-Time ECG Data Transmission in Constrained Environments

Hebatalla Ouda*, Ahmed Badr†, Abeulmonem Rashwan*, Hossam S. Hassanein*, Khalid Elgazzar†
†*Ontario Tech University, Oshawa, Ontario, Canada*
**Queen's University, Kingston, Ontario, Canada*
**20htfa@queensu.ca, arashwan@cs.queensu.ca, hossam@cs.queensu.ca*
†*ahmed.badr@ontariotechu.ca, khalid.elgazzar@ontariotechu.ca*

ICC 2022 - IEEE International Conference on Communications | 978-1-5386-8347-7/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICC45855.2022.9839049

***Abstract*—ECG monitoring systems have a significant role in detecting cardiovascular diseases and reducing the rate of sudden cardiac deaths. One of the critical factors to support real-time ECG tracking is to guarantee monitoring system availability. Hence, this work targets battery life expansion for a 12 Lead ECG patch. ECG patch operational hours are extended by reducing Bluetooth Low Energy (BLE) communication airtime, hence reducing the overall transmission power and extending the battery life. Huffman, delta, and base-delta compression techniques are implemented on a Texas Instruments CC2650 Microcontroller Unit using different sampling rates and cardiac conditions such as normal, ventricular tachycardia, and ventricular fibrillation state. The performance of each encoding algorithm is evaluated in terms of compression ratio, the execution time, and power consumption of the ECG patch. Our findings show that the base-delta encoding technique outperforms other techniques and achieves 70% data compression on normal ECG data, 41% on ventricular fibrillation, and 44% on ventricular tachycardia. The execution time of base-delta encoding takes less than 25 ms execution time and saves up to 36 % of the power consumption on the MCU environment.**

1. Introduction

The World Health Organization (WHO) reports that "The world's biggest killer is ischaemic heart disease, responsible for 16% of the world's total deaths. Since 2000, the largest increase in deaths has been for this disease, rising by more than 2 million to 8.9 million deaths in 2019." [3]. To prevent sudden cardiac attacks, developing a real-time ECG monitoring system for cardiac patients is urgently needed. ECG data must be collected and transmitted continuously to health care providers in real-time for evaluation. Internet of Things (IoT) technologies play a vital role in supporting ECG acquisition and transmission tasks.

Existing efforts in the literature focus on the number of ECG electrodes to acquire data, design of portable ECG patches, transmission mechanisms to deliver captured ECG data between the patch and the Cloud, and computational techniques that perform preprocessing on received data before making decisions [21] [7]. Nonetheless, questions related to optimizing ECG data transmission in a constrained embedded environment remain unanswered. For example, what is the efficient way to maximize the battery life of an ECG patch? How to reduce acquired data size with minimum execution time? Which compression algorithm could achieve data size reduction with minimum time and maximum energy saving? ECG platforms could witness significant improvement in terms of durability when battery lifespan is extended. In other words, if the size of collected ECG data is reduced, the required transmission power will decrease, and the total number of operational hours will increase. Expansion of ECG patch operational hours will support continuous monitoring without charging the patch frequently or worrying about ECG readings delivery interruption.

This study targets two objectives to achieve data transmission optimization in a real-time ECG platform and constrained environment. First, minimize the amount of captured ECG data before transmission to the internet gateway through the Bluetooth Low Energy (BLE) channel. Second, reduce execution time needed to reduce original data size. This will extend the ECG patch battery lifetime.

To accomplish these objectives, the proposed approach must address the constraints of embedded systems in terms of computational capabilities and airtime for BLE communication. Huffman, delta, and base-delta encoding algorithms are applied on different buffer sizes (500 and 1000 samples) to decrease the size of acquired ECG data before transmission. A quantitative comparison between these compression techniques is conducted with respect to compression ratio, execution time, and power saving.

Experimental results analysis show that base-delta encoding is the best compression choice matching our system objectives and making a trade-off between the amount of compressed data, time to execute compression, and amount of consumed power needed to transmit ECG data. Using base-delta encoding, we reach more than 70 % compression ratio, less than 25 ms to execute compression, and total 36 % reduction in power consumption for normal ECG readings.

The remainder of this paper is organized as follows: Section 2 provides a brief overview on previous work in ECG monitoring, data compression concepts, and compression techniques implemented in our system. Section 3 describes the system architecture and requirements of the proposed system. The experimental setup is explained in Section 4 concerning hardware specifications and the testing dataset. Section 5 shows the performance evaluation of each encoding algorithm in terms of compression ratio, execution time, and power consummation. Conclusion and future directions are discussed in Section 6.

2. Background & Related Work

The rapid evolution of IoT technologies draws significant interest in real-time ECG monitoring to support patients with chronic heart diseases. In the recent decade, there was a wide diversity of existing ECG monitoring systems in terms of supported features, communication protocols, and the number of ECG patch leads. For instance, some ECG monitoring systems focus on providing primary processes, such as signal acquisition, signal pre-processing, feature extraction, and signal processing [22] [26]. Other monitoring systems provide advanced processing and add more supporting features such as visualization, compression, data storage, and encryption [17] [20].

Many efforts have been introduced to support communication between ECG patches and internet gateways. Mishra et al. propose a new method for ECG monitoring based on lightweight MQTT. They collect ECG data using AD8232 Texas Instruments ECG sensor, then transmit collected data using ADS1115 16-bit ADC interface with Raspberry Pi and the I2C protocol. The digital ECG sensor data received from the ADC is published to a Cloud-MQTT broker using a mosquito client based on IEEE 802.11 WLAN [16]. Zigbee has been also used to monitor the ECG status of elderly persons outdoors, giving health care providers the real-time status of their patients [24]. However, the performance of these communication protocols in constrained environments is under investigated.

Concerning the number of leads, Herry et al. applied a Support Vector Machine (SVM) classification model on (MIT-BIH) arrhythmia database to enhance heartbeat detection and classification between normal and abnormal rhythms using a single ECG lead [10]. Similarly, Mathews et al. introduced an approach based on Restricted Boltzmann Machine (RBM) and deep belief networks (DBN) methodologies to classify ventricular and supra-ventricular heartbeats using single-lead ECG [15]. Their proposed work is limited to single-lead only with offline processing, while our platform aims to capture full 12-Lead ECG data to cover a wide range of cardiac issues in real-time. Walinjkar and Woods integrated a wearable three-lead ECG monitoring kit with a real-time arrhythmia classification and prediction model to send notification alarms while uploading the collected data to the database using HL7 and FHIR standards [23]. Although the authors claim that their approach supports real-time monitoring, they have a limited number of leads to capture precise ECG signals. A flexible 12-lead Holter is proposed using an STM32F microcontroller to support long-term monitoring. This Holter facilitates digital compression at stages close to the acquisition to overcome limitations of coverage and bandwidth of cellular networks [19]. However, their approach does not provide real-time updates and notifications to healthcare providers when abnormal conditions are detected.

Our approach targets real-time data compression with stringent execution time to satisfy resource-constrained system specifications (computation and energy) and meet real-time application requirements. This is important to enable real-time notifications when abnormal conditions are detected.

Next, we will discuss related work of data compression and different ECG compression techniques.

2.1. Data Compression Techniques

The advancement of IoT applications led to an exponential growth of generated data volumes. Data compression techniques are classified into two types: lossy and lossless algorithms. In lossy algorithms, part of data is lost after decompression. Lossless algorithms reconstruct original data without any loss. Lossy algorithms, such as Transform Coding, Discrete Cosine Transform, Discrete Wavelet Transform, and Fractal Compression are used with multimedia data (e.g., images, audio and video), while lossless algorithms, such as Run Length Encoding, Lempel–Ziv–Welch (LZW), Arithmetic Encoding, Huffman Encoding, and Shannon Fano Encoding are used with textual data [13]. In wireless sensor networks, there are five categories of compression techniques: string-based, image-based, compressed sensing, distributed source coding, and data aggregation techniques [25]. This work aims to compress ECG data after the acquisition phase to reduce the size and hence the transmission power required to send this dara. Accordingly, we are interested in applying lossless algorithms on time series ECG data, such as Huffman encoding, delta encoding, and base-delta encoding.

2.1.1. Huffman Encoding Huffman encoding is considered a leading lossless algorithm and it is widely used in most text-based applications [12] [5]. In Huffman encoding, every character in the original message is represented by binary code after generating the Huffman tree [8]. The length of binary codes depends on the frequency (i.e., times of occurrences) of each character. Huffman encoding was our first choice to apply compression on collected ECG data because of its superior compressing capabilities to the lossless compression techniques. It efficiently reduces the size of original data by assigning short codes for most repeated characters and longer codes for less repeated characters. The majority of the previous works implement the Huffman algorithm on PC as a software compression solution, while a few contributions run the Huffman algorithm on embedded systems using FPGA and VLSI [14] [11]. They use the open source MIT-BIH datasets and their main focus is primarily on power consumption.

2.1.2. Delta Encoding Delta encoding is one of the simplest compression techniques used for storing and transmitting data. Ideally, delta compression works best when the consecutive samples are very similar. It outperforms Huffman compression for data redundancy elimination when small changes occur between sequential samples [9]. There is limited usage of delta encoding with ECG in the literature. Existing contributions integrate delta encoding with adaptive Huffman to compress static MIT/BIH database [6]. We apply delta encoding as a standalone technique on a stream of collected ECG data in real-time.

2.1.3. Base-Delta Encoding .

Base-delta encoding is a modified version of the delta algorithm that measures the difference between the first sample and the remaining samples to reduce the original message size. It was introduced in 2012 to reduce caching size [18]. In this work, we utilize base-delta to compress ECG data and obtain compressed samples with varying length compared to the fixed byte size in delta compression.

3. System Design

Our proposed platform is functionally divided into five main phases: data acquisition, data transmission, data storage and streaming, data processing, and data analytics and decision-making phase, as shown in Figure 1. Badr et al. [4] describe a novel ECG platform that provides real-time electrocardiogram monitoring using deep learning and data streaming techniques to classify ECG signals and notify healthcare providers based on analysis. We extend this work through offering a robust and effective compression technique to reduce their real-time data transmission and extend the battery life. All compression algorithms we use in this study are implemented using embedded C on the TI CC2650 microcontroller chip.

3.1. System Requirements

The data acquisition task runs at a sampling rate ranging between 250 to 500 SPS. In the case of the 250 SPS, the acquisition task captures one sample every four milliseconds (250 samples/1000 ms = 4 ms). The data acquisition task itself requires 1ms to capture each sample. Therefore, the remaining time available for other tasks (e.g., compression, logging, and transmission) to operate after each sample acquisition equals to 3 ms (4 ms - 1ms). This period is down to 1 ms at 500 SPS. However, in the proposed system, we process samples in batches every one second, in which we execute compression, logging and transmission . This means we have a total of 750 ms available for these three tasks when the sampling rate is 250. In comparison, at the 500 rates, a total of 500 ms is available for running these tasks. This setup is a stringent time constraint in our system and the data compression task will have to be completed during this time interval. Otherwise, it will be interrupted by the data acquisition task since it has the highest priority according to our setup to ensure we don't miss any data.

The ECG hardware firmware encapsulates the operations conducted by the MCU into tasks using the scheduling APIs of the onboard real-time operating system (RTOS). Data acquisition, data compression, and data transmission are each encapsulated in a separate task. Each task is ranked based on a pre-configured priority. The data acquisition is set to receive the highest priority in the firmware operating on the hardware. Accordingly, the data acquisition task fulfills its call first, then the data compression and transmission run their functions according to their priority. The data compression task has the second-highest priority after the data acquisition task. To that extent, the following objectives are set to deliver the expected outcomes by the ECG acquisition hardware: (1) reduce the digitized ECG data size before transmission over BLE to the backend system; (2) maximize the ECG acquisition hardware battery lifetime by enhancing the power consumption profile of the device.

4. Experimental Setup

Our experimental work is expressed in terms of data acquisition procedure, datasets characteristics and ECG data reduction.

4.1. ECG Datasets Acquisition

To collect ECG data, we connect our data acquisition platform to the TechPatient CARDIO V4 heart simulator [1] to generate ECG datasets. This way, we avoid connecting the proposed hardware to a real patient. The simulator can generate real-time electrocardiogram (ECG) waveforms for different cardiac conditions. It supports two modes of operation: ECG mode and Rhythmic mode. The ECG mode provides a realistic 12-leads ECG waveforms. The Rhythmic mode simulates 45 predefined arrhythmias or heart diseases, such as ventricular tachycardia and ventricular fibrillation. Using the ECG Simulator, we created three different datasets: normal ECG, ventricular tachycardia, and ventricular fibrillation heart diseases. Each dataset contains a total of 10 minutes of ECG waveform recordings.

4.2. Dataset Characteristics

Our ECG data is acquired at sampling rates between 250 - 500. Each sample contains the data of eight ECG channels. Every ECG channel carries 24 bits of raw data, plus an additional 24 bits representing the status of the electrodes attached to the patient's body (i.e., whether they are correctly connected or not). Respectively, one successful sample holds 216 bits of information (8 channels * 24 bits data + 24 bits for electrode status). As a result, our ECG patch processes 6750 bytes per second while operating in the low-power mode at a rate of 250 SPS and 13,500 bytes while operating in the high-resolution mode at a rate of 500 SPS.

4.3. ECG Data Reduction

To transmit data over a wireless interface, the power consumption rate is proportional to the volume of transmitted data. As indicated above, our ECG patch generates a large

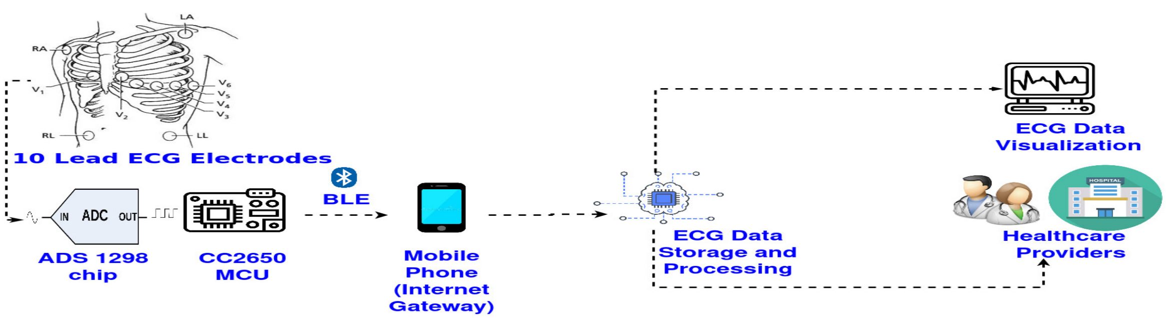

Figure 1. System Architecture of Real-time ECG monitoring Platform

volume of data each second, which requires a high amount of power to transmit. Theoretically, if we reduce the data size, we consume less power to transmit, given that we have the computational capacity to do so within the constrained time budget. Thus, we use lightweight lossless compression algorithms to increase battery lifetime and extend the ECG patch operational hours for efficient real-time monitoring . We compare the performance of Huffman, delta, and base-delta encoding on the TI CC265 MCU to determine the appropriate technique suitable for these conditions.

5. Results and Discussion

The compression techniques are assessed based on our system requirements regarding Compression Ratio (CR), Execution Time (ET), and Power Consumption Analysis with respect to the acquisition hardware constraints. The experiments evaluate the compression techniques on different buffer sizes. The buffers setup is set as double the acquisition rate per second to capture at least one complete heartbeat signal. The ECG acquisition hardware uses a sliding window of two-seconds interval to capture at least one complete cardiac cycle at a minimum heart rate of 30 bpm, and four complete cardiac cycles at a heart rate of 240 bpm. The analytical power analysis conducted shows the difference in power consumption while sending the ECG data over BLE.

The experimental environment setup consists of the following steps:

1) Collect ECG signals in real-time from the heart simulator. The CC265x powers the hardware to acquire the ECG signals in real-time from the heart simulator under three different heart conditions: normal heartbeats at 92 bpm, abnormal ventricular tachycardia and ventricular fibrillation heartbeats.
2) Create two test environments by constructing two buffers, where the buffer sizes are: 1000 and 500 samples. The buffers carry an equivalent ECG data of two seconds at 500 SPS and 250 SPS, respectively.
3) Apply Huffman, delta, and base-delta encoding compression techniques on the buffered data in two operation modes. The first operation mode is when the data acquisition hardware is working offline, which means the Radio module for BLE is turned off and the data transmission task is not scheduled. This simulates when the ECG patch is disconnected for any reason. The second mode is the real-time continuous operation mode, where the collected ECG data is compressed, saved on a local SD storage, and transmitted to the nearest paired BLE device.
4) Compare various compression techniques in terms of compression ratio and execution time.
5) Calculate the power consumption footprint and the energy-saving of applying the compression techniques on the overall system performance.

Table 1 shows the compression ratio of the three compression techniques. The Huffman encoding technique yields the highest compression ratio over delta and base-delta compression techniques. In contrast, the delta encoding fails to compress abnormal heart conditions. This is due to the uniqueness of each sample in abnormal conditions. On the other hand, the base_delta encoding shows better compression ratios in abnormal heart conditions than the delta encoding technique.

Table 2 summarizes the execution time of the compression techniques. We utilize the Real-time Clock (RTC) module on the MCU to calculate the execution time of the three compression techniques. The initial experiment runs the compression techniques in an isolated environment regardless of the other tasks (i.e., the only task running on the MCU). The Huffman encoding technique shows the highest execution time with a maximum of 112 ms on the buffer size. In contrast, the delta and base-delta techniques show significantly less execution time of a maximum of 24 ms on the same buffer size. The result from the initial experiment satisfied the time constraints of the ECG data acquisition hardware. The second experiment runs the compression task with the data acquisition, logging, and transmission tasks. The experiment results show that the device can run all the tasks at a max sampling rate of 250 SPS. However, the time constraints are violated when the sampling rate is higher than 250 SPS.

TABLE 1. COMPRESSION RATIOS WITH BUFFER SIZE (BS) = 1000, 500 SAMPLES

Dataset	Huffman Encoding		Delta Encoding		Base-Delta Encoding	
	BS 1000	BS 500	BS 1000	BS 500	BS 1000	BS 500
Normal ECG	84.9 %	84.9 %	72.3 %	73.6 %	73.6 %	74.8 %
Ventricular Tachycardia	56.1 %	56.1 %	0 %	0 %	44 %	43.2 %
Ventricular Fibrillation	56.7 %	56.7 %	0 %	0 %	41.7 %	39.1 %

TABLE 2. EXECUTION TIME (ms) WITH BUFFER SIZE (BS) = 1000, 500 SAMPLES

Dataset	Huffman Encoding		Delta Encoding		Base-Delta Encoding	
	BS 1000	BS 500	BS 1000	BS 500	BS 1000	BS 500
Normal ECG	112	109	24	23	24	23
Ventricular Tachycardia	112	109	24	23	24	23
Ventricular Fibrillation	112	109	24	23	24	23

TABLE 3. TRANSMISSION TIME OF THE ORIGINAL DATA OVER BLE AT 1 MBPS & 2 MBPS.

Compression Technique	Huffman Encoding		Delta Encoding & Base-delta Encoding	
Heart Condition	**Normal**	**Abnormal**	**Normal**	**Abnormal**
ECG Data Size	48000 bits		32000 bits	
T_{Air} (BLE 2 Mbps — 251 payload)	33 ms	33 ms	22.32 ms	22.32 ms
T_{Air} (BLE 1 Mbps — 251 payload)	48 ms	48 ms	32 ms	32 ms

To evaluate the power consumption, we calculate the overall power consumption for all active modules on the MCU while performing the intended tasks. The list of these modules and their base current consumption can be found in [2]. The power consumption is divided into two segments: the base power consumption and radio power consumption. While values of Air Time over BLE at 1 MBPS and 2 MBPS are presented in Table 3 and Table 4. The total energy consumption is calculated using the following formula (where the I's are the active modules from the board data sheet [2]):

Total Energy = Airtime (T_{Air}) * (I_{Tx} + I_{Core} + $I_{Peri\text{-}RF\ Core}$ + $I_{Peri\text{-}Power\ Domain}$ + $I_{Peri\text{-}DMA}$) + Processing time (T_{Proc}) * (I_{Core} + $I_{Peri\text{-}RF\ Core}$ + $I_{Peri\text{-}Power\ Domain}$ + $I_{Peri\text{-}DMA}$ + $I_{Peri\text{-}SPI}$)

Table 5 compares different scenarios of operation on the ECG data acquisition hardware in terms of energy consumption. The main point of comparison is the impact of the data compression techniques on the energy consumption profile. As shown in Table 5, the Huffman encoding technique imposes additional complexity on the system for signal processing, storage, and transmission. The Huffman encoding treats ECG data samples as a set of characters. This process includes additional computations in converting the digitized ECG signals from an unsigned integer data type to a character data type. Moreover, character data types require additional memory to store the data, which increases the transmission time when sending the collected data over BLE. Accordingly, the experiments show that Huffman encoding is unsuitable due to the limited resources and the time constraints of the proposed ECG acquisition hardware.

The same observation applies to the delta encoding technique except that it shows significant improvements in processing time from 112 ms to 24 ms. The delta encoding achieves adequate processing time concerning the time constraints on the hardware but fails to compress the data in abnormal heart conditions. However, the base-delta encoding completes in roughly the same time but achieved up to 44% compression ratio for abnormal heart conditions, which yielded 36% in saving energy.

The experimental results conclude that despite high compression ratio gain with the Huffman encoding technique, it does not satisfy the execution time requirements. The base-delta encoding makes a balance between the required execution time to operate and the compression ratio to fulfill our ECG data acquisition hardware constraints.

6. Conclusion

This study investigates the impact of data compression on energy saving in constrained embedded environments using TI-CC2650 MCU. The Huffman, delta, and base-delta encoding algorithms are evaluated in regards to the compression ratio, execution time, and energy consumption for data transmission. The base-delta encoding outperforms both the Huffman and delta encoding techniques and achieved a 24 ms execution time, 70% in compression ratio in normal cardiac status, a 41% compression ratio in ventricular fibrillation and 44% on ventricular tachycardia. The delta encoding technique saved 36% of the power consumption compared to no compression. In the future, we plan to apply the base-delta encoding on various cardiac conditions and study the effect of data compression on enhancing the power consumption profile for these conditions.

References

[1] He instruments — patient simulators. retrieved december 29, 2020. https://www.heinstruments.com.

[2] https://www.ti.com/lit/ds/symlink/cc2650.pdf?ts=1635189756259&ref_url=https\%253A\%252F\%252Fwww.ti.com\%252Fproduct\%252FCC2650.

TABLE 4. TRANSMISSION TIME OF THE COMPRESSED DATA OVER BLE AT 1 MBPS & 2 MBPS.

	Huffman Encoding		Delta Encoding		Base-Delta Encoding		
Heart Condition	**Normal**	**Abnormal**	**Normal**	**Abnormal**	**Normal**	**Abnormal 1**	**Abnormal 2**
ECG Data Size	7245 bits	21032 bits	8744 bits	32000 bits	8744 bits	17896 bits	18648 bits
T_{Air} (BLE 2 Mbps — 251 payload)	5.05 ms	14.67 ms	6.09 ms	22.32 ms	6.09 ms	12.47 ms	13 ms
T_{Air} (BLE 1 Mbps — 251 payload)	7.245 ms	21.032 ms	8.744 ms	32 ms	8.744 ms	22.28 ms	23.22 ms

TABLE 5. ENERGY CONSUMPTION (mJ) WITH / OUT DATA COMPRESSION WITH TRANSMITTING DATA OVER BLE.

	Huffam Encoding		Delta Encoding		Base-delta Encoding		
Heart Condition	**Normal**	**Abnormal**	**Normal**	**Abnormal**	**Normal**	**Abnormal 1**	**Abnormal 2**
Data Compression Disabled &Transmission Enabled (BLE 2 Mbps — 251 payload)	365.61		247.28				
Data Compression Disabled &Transmission Enabled (BLE 1 Mbps — 251 payload)	531.79		354.52				
Data Compression &Transmission Enabled (BLE 2 Mbps — 251 payload)	479.19	585.77	158.16	337.97	158.16	228.85	234.72
Data Compression &Transmission Enabled (BLE 1 Mbps — 251 payload)	503.515	656.261	187.57	445.22	187.57	337.53	347.95

[3] The top 10 causes of death, https://www.who.int/news-room/fact-sheets/detail/the-top-10-causes-of-death, 2020.

[4] Ahmed Badr, Abeer Badawi, Khalid Elgazzr, and Abeulmonem Rashwan. Real-time electrocardiogram monitoring and analysis. *ACM Transactions on Computing for Healthcare (under review)*, 2021.

[5] Zhengwen Cao, Yan Li, Jinye Peng, Geng Chai, and Guang Zhao. Controlled quantum secure direct communication protocol based on huffman compression coding. *International Journal of Theoretical Physics*, 57(12):3632–3642, 2018.

[6] G. C. Chang and Y. D. Lin. An efficient lossless ecg compression method using delta coding and optimal selective huffman coding. In *6th World Congress of Biomechanics (WCB 2010), Singapore*. Springer Berlin Heidelberg, 2010.

[7] Nasir Faruk et al. A comprehensive survey on low-cost ecg acquisition systems: Advances on design specifications, challenges and future direction. *Biocybernetics and Biomedical Engineering*, 2021.

[8] Athira Gopinath and M Ravisankar. Comparison of lossless data compression techniques. pages 628–633, 2020.

[9] Athira Gopinath and M Ravisankar. Comparison of lossless data compression techniques. In *2020 International Conference on Inventive Computation Technologies (ICICT)*, pages 628–633, 2020.

[10] Christophe L Herry, Martin Frasch, Andrew JE Seely, and Hau-tieng Wu. Heart beat classification from single-lead ecg using the synchrosqueezing transform. *Physiological measurement*, 38(2):171, 2017.

[11] Chio-In Ieong, Mingzhong Li, Man-Kay Law, Pui-In Mak, Mang I Vai, and Rui P. Martins. A 0.45 v 147–375 nw ecg compression processor with wavelet shrinkage and adaptive temporal decimation architectures. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 25(4):1307–1319, 2017.

[12] Faiz Muqorrir Kaffah, Yana Aditia Gerhana, Ihsan Miftahul Huda, Ali Rahman, Khaerul Manaf, and Beki Subaeki. E-mail message encryption using advanced encryption standard (aes) and huffman compression engineering. pages 1–6, 2020.

[13] P Kavitha. A survey on lossless and lossy data compression methods. *International Journal of Computer Science & Engineering Technology*, 7(03):110–114, 2016.

[14] PG Kuppusamy, R Sureshkumar, SA Yuvaraj, and E Dilliraj. Vlsi based lossless ecg compression algorithm implementation for low power devices. In *Journal of Physics: Conference Series*, volume 1964, page 062073. IOP Publishing, 2021.

[15] Sherin M Mathews, Chandra Kambhamettu, and Kenneth E Barner. A novel application of deep learning for single-lead ecg classification. *Computers in biology and medicine*, 99:53–62, 2018.

[16] Ayaskanta Mishra, Akanksha Kumari, Pooja Sajit, and Pranjal Pandey. Remote web based ecg monitoring using mqtt protocol for iot in healthcare. *Development*, 5(04), 2018.

[17] Andrea Němcová, Radovan Smíšek, Lucie Maršánová, Lukáš Smital, and Martin Vítek. A comparative analysis of methods for evaluation of ecg signal quality after compression. *BioMed Research International*, 2018.

[18] Gennady Pekhimenko, Vivek Seshadri, Onur Mutlu, Michael A Kozuch, Phillip B Gibbons, and Todd C Mowry. Base-delta-immediate compression: Practical data compression for on-chip caches. In *2012 21st international conference on parallel architectures and compilation techniques (PACT)*, pages 377–388, 2012.

[19] Flavio Pineda-López, Andrés Martínez-Fernández, José Luis Rojo-Álvarez, Arcadi García-Alberola, and Manuel Blanco-Velasco. A flexible 12-lead/holter device with compression capabilities for low-bandwidth mobile-ecg telemedicine applications. *Sensors*, 18(11):3773, 2018.

[20] Udit Satija, Barathram Ramkumar, and M Sabarimalai Manikandan. *A review of signal processing techniques for electrocardiogram signal quality assessment*, volume 11, pages 36–52. 2018.

[21] Mohamed Adel Serhani, Hadeel T El Kassabi, Heba Ismail, and Alramzana Nujum Navaz. Ecg monitoring systems: Review, architecture, processes, and key challenges. *Sensors*, 20(6):1796, 2020.

[22] Javier Tejedor, Constantino A. García, David G. Márquez, Rafael Raya, and Abraham Otero. Multiple physiological signals fusion techniques for improving heartbeat detection: A review. *Sensors*, 19(21), 2019.

[23] Amit Walinjkar and John Woods. Ecg classification and prognostic approach towards personalized healthcare. In *2017 International Conference On Social Media, Wearable And Web Analytics (Social Media)*, pages 1–8. IEEE, 2017.

[24] Liang Hung Wang, Yi Mao Hsiao, Xue Qin Xie, and Shuenn Yuh Lee. An outdoor intelligent healthcare monitoring device for the elderly. *IEEE Transactions on Consumer Electronics*, 62(2):128–135, May 2016.

[25] You-Chiun Wang. Data compression techniques in wireless sensor networks. *Pervasive Computing*, 61(1):75–77, 2012.

[26] Ilsun You, Kim-Kwang Raymond Choo, Chi-Lun Ho, et al. A smartphone-based wearable sensors for monitoring real-time physiological data. *Computers & Electrical Engineering*, 65:376–392, 2018.