Journal of Computer Science Research
https://ojs.bilpublishing.com/index.php/jcsr

ARTICLE

Bidirectional Recurrent Nets for ECG Signal Compression

Eman AL-Saidi Khalil El Hindi*

Computer Science Department, King Saud University, Riyadh, Saudi Arabia

ARTICLE INFO

Article history
Received: 25 October 2022
Revised: 23 November 2022
Accepted: 28 November 2022
Published Online: 30 November 2022

Keywords:
BERT model
Convolutional neural networks (CNN)
Data compression
Deep learning
ECG diagnosis

ABSTRACT

Electrocardiogram (ECG) is a commonly used tool in biological diagnosis of heart diseases. ECG allows the representation of electrical signals which cause heart muscles to contract and relax. Recently, accurate deep learning methods have been developed to overcome manual diagnosis in terms of time and effort. However, most of current automatic medical diagnosis use long electrocardiogram (ECG) signals to inspect different types of heart arrhythmia. Therefore, ECG signal files tend to require large storage to store and may cause significant overhead when exchanged over a computer network. This raises the need to come up with effective compression methods for ECG signals. In this work, the authors investigate using BERT (Bidirectional Encoder Representations from Transformers) model, which is a bidirectional neural network that was originally designed for natural language. The authors evaluate the model with respect to its compression ratio and information preservation, and measure information preservation in terms of the of the accuracy of a convolutional neural network in classifying the decompressed signal. The results show that the method can achieve up to 83% saving in storage. Also, the classification accuracy of the decompressed signals is around 92.41%. Furthermore, the method enables the user to balance the compression ratio and the required accuracy of the CNN classifiers.

1. Introduction

Recently, cardiovascular diseases (CVDs) have become one of the main causes of mortality in the world. The number of people died from CVDs is estimated at 17.9 million in 2019 according to the world health organization (WHO) [1]. ECG signal represents the electrical signals changes during a cardiac cycle. It shows heart muscle polarization and depolarization, and it is widely used by automatic system to inspect different types of heart arrhythmia. ECG consists of three main components: the P wave, which represents atria depolarization; QRS complex, which represents ventricles depolarization; and the T wave, which represents ventricles repolarization [2]. Figure 1 illustrates the components of an ECG signal. ECG signals need to be long to obtain an accurate diagnosis of a patient condition. Therefore, developing a practical automated system for dealing with ECG signals requires

*Corresponding Author:
Khalil El Hindi,
Computer Science Department, King Saud University, Riyadh, Saudi Arabia;
Email: khindi@ksu.edu.sa

DOI: https://doi.org/10.30564/jcsr.v4i4.5204

compressing these signals before storing or transferring them over a computer network [3].

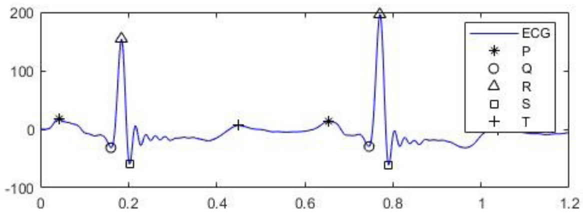

Figure 1. ECG components

The main challenge is to compress an ECG signal in such a way that retains its discriminating features. This is a major challenge especially if we want to attain a high compression ratio, at the same time. It is therefore important to evaluate compression performance in terms of both high compression ratio and quality of decompressed signals to ensure that the reduction in size does not cause a serious loss in diagnostic information.

Deep learning methods have proved to be good predictors in many domains, and according to information theory, good predictors can be utilized as effective compressors [4]. Many existing compressors attempt to learn from data and perform prediction-based compression. Deep convolutional auto-encoders have been used to compress ECG. The encoder section of the suggested model reduces ECG signal to low-dimensional vectors, and the signals are reconstructed by a decoder [5]. Also, auto-encoders with long short-term memory (LSTM) were proposed for ECG signal compression, where LSTM network helps to reconstruct the original data from its compressed file [6], a deep compression approaches can learn to compress different ECG records automatically and perform well in terms of compression ratio and quality of reconstructed signal [7].

In this paper, we introduce a novel Bert-based ECG compressor for ECG. BERT was originally designed for natural language applications. We design a Tiny BERT that is pre-trained on unlabeled ECG beats and then fine-tuned to predict next beats based on previous ones. We exploit the BERT predictor as a compressor for ECG signal. Our method is faster than transformational methods, as it can learn automatically from different ECG records without any need to analyze the domains. It achieves a competitive compression ratio while retaining much of diagnosis information. This is evident as we can use the reconstructed (decompressed) signal to classify it using 1D-CNN with a high accuracy.

We can summarize the main features of our compressor as follow:

i. The proposed method does not perform any analysis of the domain to generate the compressed file. The model is pre-trained to learn ECG beats using the MIT-BIH data-set, and then the model is fine-tuned (trained) for different ECG records to predict next beats based on a number of saved beats and thus act as a compressor for different ECG signals.

ii. The proposed model performance -in terms of its ability to retain diagnostic information- has been validated by using 1D-CNN to diagnose heart problems using the decompressed ECG signals as input data. Our empirical work shows that the classification accuracy of decompressed signal using 1D-CNN is very close to the accuracy of the original ECG signals.

iii. Our system can be used in real time clinical diagnosis, where ECG can be recorded, compressed and then transformed through different channels to other parties who can decompress it and use for classification.

The organization of this paper is as follow: in Section 2, we review some related works. Then in Section 3, we describe the proposed method. Section 4 evaluates our compression algorithm and describes ECG data-set. In Section 5, we present and analyze the results. Section 6 is the conclusion.

2. Related Works

In this section, we describe the different methods for compressing ECG signals and we review transformers in general and the BERT model in particular.

2.1. Compression Methods for ECG Signals

ECG signal represents Cardiac electrophysiology, and it has been widely used in cardiac medical diagnosis and has the advantages of real time. ECG recording is long and there is a need to compress the recordings during storage and transmission phases to make an efficient use of channels.

Many automatic compression techniques have been proposed in the literature to compress ECG signals and they can be classified into three main classes of parameter extraction methods, transformation methods and direct methods [7-15]. These three classes differ mainly in processing domain. Direct methods analyze time domain features to extract samples from ECG to act as compressed file, and then the original signal is reconstructed by an inverse process. Direct methods require time and efforts in designing an appropriate compressing algorithm signals through the extraction of a subset of samples from the original signal sample set, they are usually fast because they use heuristic analysis [7]. Transformation domain methods transform ECG signal to frequency domain or other domains to analyze energy distribution and compress ECG signal, then an inverse transformation is performed to reconstruct the original signal [14-18]. Parameter extraction method is based on extracted optimized and discriminated features from original signal to reduce the size of ECG signal [4].

In time-domain based methods, a subset of representative samples is extracted from ECG time domain. Usually this task is tedious and time consuming; the reconstruction method is performed by an inverse process. Kumar et al. [19] developed an improved Amplitude Zone Time Epoch Coding (AZTEC) over existing (AZTEC). Statistical parameters of ECG signal are extracted and adapted in-accordance with the nature of signal by calculating a threshold value. The proposed method optimized a tradeoff between ECG information and data reduction. The performance of method was evaluated on the basis of compression ratio and reached a value of ranges between 2.76 and 9.91. L. Zheng et al. [7] proposed a decomposition method utilized singular value decomposition (SVD) to decompose ECG signals and compress it; the ECG signal in the proposed method is divided into segments by QRS peak detection of length n, each segment represents a part of ECG morphology. At end, the ECG signal is represented by a matrix composed of the normalized ECG segments to be compressed. SVD is applied on the matrix to compress ECG signal then the decompressed data are passed to a convolutional neural network (CNN) and supporting vector machine (SVM) for ECG classification. They obtained an average accuracy exceeds 96%.

In transformation-domains based methods, ECG signal is transformed to another domain and the energy distribution is analyzed to compress ECG in the new domain. S. M. Ahmeda et al. [20] used four different discrete wavelet transforms, then the wavelet coefficients are linearly predicted and errors are minimized. A compression ratio of 20 to 1 is reached after encoding the residual sequence obtained using linear prediction. J. Chen et al. [21] presented a new ECG compression technique based on adaptive quantization strategy and orthogonal wavelet transform, by which a user-specified percent root mean square difference (PRD) of the reproduced ECG segments is guaranteed at the minimum entropy.

H. M. Tun et al. [22] proposed transform ECG compression technique, and the method optimized compression ratio by removing redundancy in ECG signal. They used different types of wavelet to decompose ECG signal. The decomposed signals are compressed by applying global and local thresholding and run-length encoding. The performances of different types of wavelet are evaluated in terms of compression ratio (CR) and percent root mean square difference (PRD). M. Elgendi et al. [15] transformed ECG signal to frequency domain using discrete cosine transform to compact ECG signal energy to lower frequency coefficients, then coefficients normalization process is performed. Finally, the normalized and rounded coefficients are encoded using Huffman encoding and run length encoding to compress data.

K.C. Hung et al. [23] proposed a genetic algorithm (GA) to optimize the quantization of wavelet ECG signals to compress data, where GA define stationary relationship property to control quantization scales of multi-resolution levels by using a single variable. Quantization scheme with linear distortion characteristic that does not depend on ECG signal is derived to reduce error. C. K. Jha et al. [24] proposed a method that used adjustable parameters to compress ECG depending on the tunable Q-wavelet transform that reduce the energy of ECG signal to lower transform coefficients, quantization and rounding of coefficients is performed to discard small values and then apply encoded using run-length coding. ECG records of the MIT-BIH arrhythmia were used to measure algorithm performance. The average compression performance obtained in terms of CR is 20.61. Also, they used support vector machine to measure the quality of reconstructed signals, results reported accuracy of 98.35%. It indicates that the proposed method compressed ECG signal efficiently and preserved diagnostic information. S. K. Mukhopadhyay et al. [25] proposed a compression technique of ECG signal, first the ECG signals was de-noised and threshold, then ECG beats were extracted and arranged into (2D) matrix, the generated matrix was decomposed by (SVD), the number of singular matrices are truncated and encoded into ASCII characters, the LLACE compressed left truncated singular matrix coefficients. Performance of method is expressed in terms of subjective measure, where results indicated that quality of reconstructed ECG is very good. Discrete wavelet transform was applied to first intrinsic mode and four sifting functions. Then quantization and rounding were performed on the transformed coefficients. ECG compressed file was generated by running length encoding. The proposed methods outperformed existing methods in terms of quality of reconstructed data [26]. P. Kanjalkar et al. [28] applied wavelet transform to the ECG signal which reduces energy to a smaller number of transform coefficients. The coefficients are quantized and all small-value coefficients were discarded while others were kept according to the formulated quantized output interval. An average value is assigned to some coefficients, and then the coefficients are encoded using modified run-length coding. The average compression ratio was 17.18 for 84 ECG records. In most cases, direct methods are better than transform methods with respect to system complexity and error control mechanism, but transform methods usually achieve higher compression ratios and elaborate noise contained in original ECG signals.

Deep learned features systems have outperformed hand-engineered features in the field of ECG arrhythmia classification systems, and it is known from theory of information that a good predictor can act as a good compressor. So, many of new suggested compressors attempt to learn models for the data and accomplish compression using prediction. Neural networks show excellent results in many predictions tasks. Recurrent neural networks were used as data compressor and the results show that deep learning models can be good alternatives for traditional compression techniques [4].

2.2 Transformers and the BERT Model

BERT, which stands for Bidirectional Encoder Representations, is based on transformers and its main principle-attention, which understands the textual relationship between different input words by determining the importance of each word with respect to others in a sentence or which words are relevant to come together. BERT produces language model through encoding rather than decoding the encoded information, so using an encoder of a transformer is sufficient. Contrasted to directional models such as RNN and LSTM which process each input sequentially (left to right or right to left). Transformer and BERT are non-directional, because both do not perform sequential ordering; they take in the whole sentence as input. The non-directional property allows models to learn the textual information of a word with respect to all other words in the sentence. BERT has been used in language representation models and it shows good results in natural language processing tasks such as text classification, sentence classification, and semantic similarity between pairs of sentences, question answering tasks, and text summarization. The novelty of BERT is that it acts as language model that uses attention to solve tasks which contains sequences without using recurrent connections, which delay the training process, so we felt motivated to use it as a language model for ECG signals.

As we mentioned earlier BERT is a transformer-based model, it composed of encoder only, while the transformer is composed of an encoder and decoder. BERT uses different hyper-parameters such as attention heads to achieve the best performance than typical transformers.

A transformer is a deep learning model that weights the significance of each input through adopting self-attention mechanism; it is designed to process the entire sequential input data at once. It has Encoder-decoder architecture, as can be seen in Figure 2 [29].

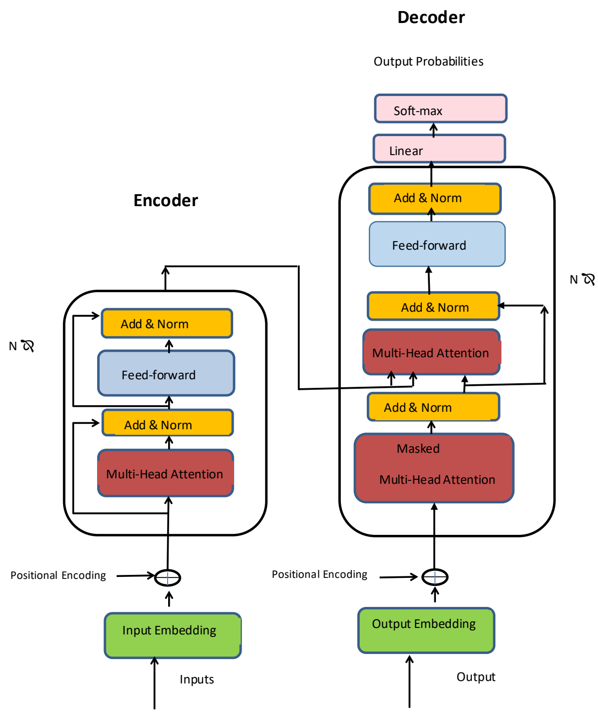

Figure 2. The transformer model architecture

Encoder: Encoder makes up of a stack of a number of identical encoding layers and each layer's input is the output of the previous layer. Each encoding layer generates encoding representation of each input, which contains information about relevancy of each input to others and passes it to the next encoding layer as input. Each layer has multi-head self-attention sub layer, and fully connected feed-forward network sub layer. A residual connection followed by layer normalization is performed around each of the sub-layers.

Decoder: The decoder is also composed of a stack of a number of identical layers; each decoder layer performs the opposite of the corresponding encoder, taking all the encoding representation of inputs to generate an output sequence. To accomplish this, the decoder has a third sub-layer, which performs multi-head attention over the output of the encoder stack. Similar to encoder a residual connection followed by layer normalization is performed around each of the sub-layers.

Encoder and decoder layers use attention mechanism. We give a brief description of self-attention mechanism.

Self-attention mechanism: The transformer building blocks are attention units that produce embedding for every input. Each embedding contains information of other relevant inputs, each weighted by its attention weight. Attention in transformers is based on scaled dot production attention, which is performed with the definition of three concepts: queries (Q), keys (K), and values (V). Keys represent input; queries represent requests for keys of significant information while checking all available keys. And values represent the retrieved values of queries matched keys. The definition of attention layer is given in Equation (1):

$$\boldsymbol{Attention(Q,K,V) = softmax\left(\frac{QK^T}{\sqrt{d^k}}\right)V} \quad (1)$$

where d is the dimension of the input vector.

For each an attention unit the transformer model needs to learn three weight matrices; the query weights W_Q, the key weights W_K, and the value weights W_V. One set of (W_Q, W_K, W_V) is called attention head, and each layer has multiple attention heads. Each attention head attends to the inputs that are relevant to each input, the transformer can do head attention for different types of relevance. For each set of weights (i), the head definition can be seen in Equation (2):

$$\boldsymbol{Head_i = Attention\left(QW_Q^i, KW_K^i, VW_V^i\right)} \quad (2)$$

3. Proposed BERT-based Compression (BBC) Method

In this section, we describe how to use BERT as a data compression method for ECG signals. The main idea is to save only the beats that BERT cannot predict and that are not predicted with good accuracy by the BERT model. First of all, we need to train the BERT model to understand different types of ECG arrhythmia. To accomplish this task, we pertained BERT to predict the next beat based the previous n beats. We conducted several experiments to determine n, which is the number of beats that we need to predict the next beat with good accuracy. The best accuracy was obtained when we set n to 10.

We then used a number of saved beats to predict the same number of beats; we tried four numbers of beats, 25, 20, 15 and 10 with 32 for different data sets. After performing pre-training stage, our Bert model can now understand different types of arrhythmia types and can be used in a fine-tuning phase to compress different ECG signal. It is worth mentioning that we used Tiny-BERT which is a multi-layer bidirectional transformer encoder based on implementation [29]. In our model architecture, L, H and A refer to number of layers (Transformer Blocks), Hidden size, and number of self-attention heads, respectively. Since four types of beats only, which is much smaller than the number of words in a natural language, we used Tiny-BERT [29] instead of the full BERT architecture. Tiny-BERT is very small compared to word language diversity. We trained number of tiny models with a differing numbers of self-attention, namely 6, 4, and 2. We also experimented with different number of layers namely 4, 3, and 2). The hidden size was set to 128 in all models. The best number of layers was 2 regardless the number of self-attention heads. We used a model of self-attention of 4 and number of layers of 2 with hidden size of 128. We call the chosen model CBERT (for Compression BERT).

Next, we train (fine-tune) CBERT to predict the next block of beats based on the current block of beats. A block has a fixed size of n beats. We set n=20 during this stage.

We use the trained CBERT to compress the ECG signal file as follows; we start with the first block of n beats in the ECG signal file and store it on the output file. We also use this block as input to BERT to predict the next block; also of n beats. We then find the maximum number of consecutive beats p in the predicted block with a good classification accuracy, i.e., above a certain threshold t. We replace the n-p beats that where not predicted correctly in the predicted block with the corresponding beats in the ECG file; we store the block in the output file as use it as input to BERT in the next iteration. The process continues until the end of the ECG signal file. Algorithm-1 describes the compressed method in details.

The decompression method is straight forward. We use the fine-tuned BERT to decompress the ECG signal file as follows; we a start with the first block of n beats in the ECG compressed file and store it on the ECG output file. We also use this block as input to BERT to predict the next block; also of n beats. We concatenate first predicted p beats with n-p in the corresponding compressed file, and store the block in the output file as use it as input to BERT in the next iteration. Process continues until the end of the ECG compressed file. Algorithm-2 describes the decompressed method in details.

Algorithm-1. Bert-Based-Compression

Input: *ECG signal file*
: n a block consists of n beats ;
: t is an accuracy threshold
Output: *compressed ECG signal file*

```
Let vector V be the first block of n beats in the ECG signal
Store V on the output file (the compressed file)
While not end of ECG signal file do
    Let P be the block of n beats predicted by BERT given V as the input block
    Let p first consecutive beats in P with c
    Replace the remaining n – p beats in P with the corresponding beats in the ECG signal file
    Let V=P
    Remove the first p beats from P
    insert the integer number p at the beginning of P
    Sotre P in the output file
end
```

Algorithm-2. Bert-Based-Decompression

```
Input: ECG compressed file
       : n represents a fixed number of beats
Output: ECG signal file
Let vector V be the first block of n beats in the ECG compressed file
Store V on the output file (ECG signal file)
While not end of ECG compressed signal file do
    Read p which is the number of correctly predicted beats
    Use CBERT to predict p beats, based on V
    Read n-p beats from the input file file
    Let V be the concatenation of the predicted beats with the read beats
    Store V on the decompressed output file
end
```

Our empirical work showed that CBERT gives better results than BERT; this is probably because our problem is much simpler than the natural language processing application that BERT was designed for. The full BERT seems to over-fit our training data.

4. Evaluating the Proposed Bert-Based Compression Method

In this section, we describe the empirical results we obtained when using our Bert-Based compression method to compress some real world ECG signal files. Data accuracy is very important to help cardiologists making an accurate diagnosis. It allows quick treatments and can save patients life. So, while compressing ECG signal during transmission and storage, the performance of ECG compression technique must evaluate. The most important issue is to verify quality of the reconstructed ECG signal and preserve the morphological features which are significant to obtain high classification accuracy. Therefore, evaluate BBC not only with respect to the compression ratio but also to the quality of the decompressed file with respect to the diagnostic information it retains. This is done be using the decompressed ECG as input to a pre-trained 1D-CNN to classify it. The closer the obtained accuracy to the accuracy of the 1D-CNN classifier on the original ECG signal, the better the compression method.

Compression ratio (CR) is defined as the ratio of the original signal size and compressed signal size. The CR provides information about how compression algorithm can remove redundant data. The higher the CR is, the less space is used to save it. CR is used as an indicator of how much data has been compressed and thus save space. Also, we use accuracy and F1- measure to measure the performance of reconstructed ECG Signal. The definitions of these measures are given by Equations (3), (4) and (5), respectively.

$$\boldsymbol{Space\ saving}\ (\boldsymbol{Reduction\ rate})\ (\boldsymbol{S}) = \left(\mathbf{1} - \frac{\boldsymbol{S_c}}{\boldsymbol{S_o}}\right) * \mathbf{100}\% \tag{3}$$

$$\boldsymbol{Accuracy}\ (\boldsymbol{Acc}) = \frac{\boldsymbol{TP+TN}}{\boldsymbol{TotalSamplesNumber}} \tag{4}$$

$$\boldsymbol{F1} = \frac{\boldsymbol{2*precision*recall}}{\boldsymbol{precision+recall}} \tag{5}$$

where, S_C and S_O represent the size of compressed file and the original file, respectively. True Positive (TP) refers to number of normal beats correctly classified; True Negative (TN) refers to number of abnormal beats classified as abnormal beats, False Negative (FN) refers to number of abnormal beats classified as normal beats, False Positive (FP) refers to number of normal beats classified as abnormal beats, $\text{Precision} = \frac{\text{TP}}{\text{TP+FP}}$, $\text{Recall} = \frac{\text{TP}}{\text{TP+FN}}$.

ECG Data

We acquired benchmark data-sets, which are available from public domains such as MIT-BIH arrhythmia data-set. MIT-BIH is the most commonly used database that was developed to act as an objective evaluation tool for different arrhythmia classification systems; it consists of 48 records of ECG data, each record is 30 minutes long, 23 records of the 48 records were chosen randomly from a large set of long-term ECG Holter recordings to represent variations of ECG data, while the remaining records were chosen specially to represent complex arrhythmia such as supra-ventricular arrhythmia that may encounter arrhythmia classification systems. The recordings were digitized at 360 samples per second per channel with 11-bit resolution over a 10 mV range. Each ECG record contains two leads, each lead is recorded by placing electrodes on classification places on the skin. MIT-BIH data-set was pre-processed and digitized, a set of rhythm labels and beat labels were added to each record, during the early usage of the data-set; some of these labels were revised and corrected several times.

5. Discussion and Results

We acquired data from ECG signal and pre-trained Tiny-Bert models, then fine-tuned multiple ones for different ECG records. In this section, we discuss the results for different cases related to multiple ECG records. Each ECG record is fine-tuned using a pre-trained defined model. During the compression process, we monitor average predication accuracy using BBC algorithm, which ensures that predication accuracy does not go beyond specific threshold. We present three different threshold values of 98%, 95%, and 90% and calculate the efficiency of compression ratio in terms of saving space of three different ECG records donated as (DS1, DS2, and DS3). Each data-set represent 1-hour ECG recording of patient extracted from MIT-BIH database and each data-set has four different types of beats, Normal beat, Left bundle branch block beat, Right bundle branch block beat and Bundle branch block beat We address different model size with a differing number of layers and attention heads to give more insight into the effect of model size on the performance of the model during compression.

We calculate the compression performance of each record in terms of saved space (S). We run different models with various multi head attention and encoders (layers) number. The results are presented in Tables 1, 2 and 3.

Then we calculated the average saved space for each multi-head attention number to clarify the effect of multi-head attention number on different datasets. The results are represented in Table 4. Based on all results in Table 4, it can be observed that there is no significant evidence that number of multi-head attention has strong effect on S. Furthermore, the classification accuracy of all cases differs in range of 0-2.62% which gives an indication that the number of multi heads attention has a small effect on model performance. In addition, we calculate the average accuracy of saved space per number of layers. The results are indicated in Table 5. In most cases, using two layers has the highest performance in term of saving space. To give more insight into our compression technique, a visualization of the reduction rate percentage in each block of the compressed ECG file of DS1 is given in Figure 3. As we can see from Figure 3, the reduction rate in blocks after the first block is between 75% and 85% which gives an indicator of a high compression ratio and thus a good compression technique.

Table 1. Save space with different data-sets and threshold of 95%

Threshold 95%	**Multi heads attention number**								
Layers number	6	4	2	6	4	2	6	4	2
	DS1			DS2			DS3		
4	81.42%	80.76%	81.12%	79.25%	81.22%	81.25%	80.42%	81.87%	80.35%
3	82.87%	83.77%	83.96%	82.14%	82.35%	82.40%	82.54%	80.76%	81.12%
2	83.76%	83.91%	82.31%	83.11%	83.13%	83.25%	82.77%	81.14%	83.22%

Table 2. Save space with diffrent dat-sets and threshold of 90%

Threshold 90%	**Multi heads attention number**								
Layers number	6	4	2	6	4	2	6	4	2
	DS1			DS2			DS3		
4	82.54%	80.43%	81.10%	83.55%	81.75%	81.25%	80.63%	80.44%	81.51%
3	83.11%	82.67%	83.43%	84.14%	82.68%	82.78%	83.48%	83.89%	83.65%
2	83.56%	83.88%	83.34%	84.77%	83.44%	83.56%	81.84%	81.75%	83.23%

Table 3. Save space with different data-sets and threshold of 98%

Threshold 98%	**Multi head attention number**								
Layers number	6	4	2	6	4	2	6	4	2
	DS1			DS2			DS3		
4	80.05%	80.46%	80.50%	77.35%	80.27%	82.44%	81.55%	81.22%	81.25%
3	81.67%	82.75%	83.11%	80.13%	82.35%	81.60%	82.31%	80.51%	81.12%
2	83.11%	83.88%	84.17%	82.40%	82.44%	83.14%	81.78%	80.34%	83.15%

Table 4. Average Save space and classifcation accuracy for diffrent head attention number

Threshold 95%	**Multi head attention number**								
	6	4	2	6	4	2	6	4	2
	DS1			DS2			DS3		
S	82.6%	82.81%	82.46%	81.5%	82.23%	82.3%	81.91%	81.25%	81.56%
Acc	92.78%	91.65%	92.88%	93.22%	93.13%	93.14%	92.67%	93.13%	93.71%
Threshold 98%									
S	81.61%	82.36%	82.59%	79.69%	81.68%	82.19%	81.88%	80.69%	81.84%
Acc	93.73%	92.98%	92.24%	93.67%	92.56%	92.42%	93.43%	92.84%	92.13%
Threshold 90%									
S	83.07%	82.81%	82.46%	84.15%	82.62%	82.55%	81.98%	82.02%	82.79%
Acc	91.44%	91.54%	91.64%	92.11%	91.11%	91.13%	91.99%	91.10%	91.13%

Table 5. Average save space for different layers number

Threshold 95%									
Layers number	4	3	2	4	3	2	4	3	2
	DS1			DS2			DS3		
S	81.1%	83.53%	83.32%	80.57%	82.29%	83.16%	80.88%	81.47%	82.37%
Threshold 98%									
S	80.33%	82.51%	83.72%	80.02%	81.36%	82.66%	81.34%	81.31%	81.74%
Threshold 90%									
S	81.47%	83.61%	83.26%	82.18%	83.2%	83.95%	80.86%	83.67%	82.27%

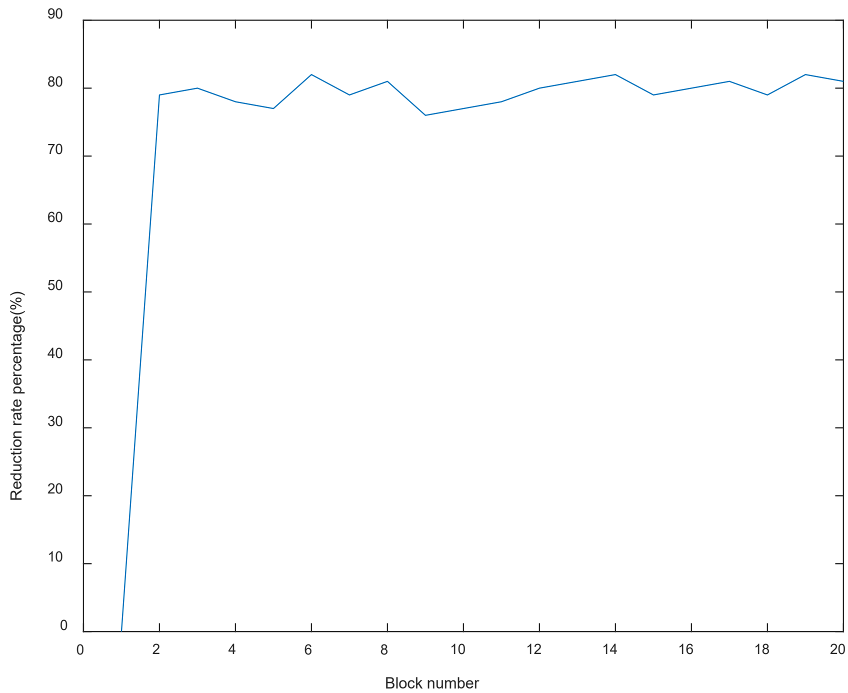

Figure 3. Reduction rate percentage (%) of DS1

To measure the quality of reconstructed ECG signal, we calculated the average classification accuracy of the reconstructed ECG with respect to original files. This process is performed by passing it to 1D-CNN and compared the classification accuracy with respect to original files. The average accuracy of all cases of reconstructed original files reached 92.41% which is less by approximate 2% than our obtained classification accuracy of ECG signal using ensemble model which prove the robustness of our compressor model and prove that it can be used in real time diagnosis. The accuracy is presented in Table 6. Also, we obtained an average F1-score of 0.8266 as indicated in Table 7. In addition that we have achieved a high classification accuracy of the reconstructed ECG with respect to original files, a visualization of an original ECG signal block (before compression) and the reconstructed one (after compression) is shown in Figure 4 and Figure 5 respectively. A key challenge for data compression is to reconstruct the original data that maintains the correct information. If data compression results in the loss of information; it will reduce the performance of the compression technique. According to our data compression it can be easily verified that it is information-preserving.

Our proposed method based on deep learning method. Conventional compression methods like parameter extraction methods, transformation methods and direct methods usually process different domains to extract features. Direct methods analyze time domain features to compress ECG signal. Which require time and efforts [7]. Transformation domain methods transform ECG signal to frequency domain or other domains to compress ECG signal, then an inverse transformation is performed to reconstruct the original signal [14-18]. Parameter extraction method is based on extracted discriminated features from original signal to reduce the size of ECG signal [4]. Since our proposed compression algorithm used deep learning model, it does not perform any analysis of domain to generate compressed file. It acts as an end-to-end model, where there is no need to extract any domain features. Also, it achieved a competitive saving space reached to 83% conventional methods in-terms of save spacing.

In contrast to M.Goyal et al. [4] method which used RNN that represents a shallow concatenation of a left-to-right and a right-to-left model. Our model is a truly bidirectional learning model based on attention mechanism, where the model learns information from left to right and from right to left, which make our model more powerful than their method. Furthermore, our model is designed for ECG signal, which is huge and consume storage over network, so our systems can be used in diagnose systems.

Table 6. The (%) accuracy of different threshold values

Threshold value	95%	98%	90%	Average
Accuracy	92.9	92.88	91.46	92.41

Table 7. The F1 score of different threshold values

Threshold value	95%	98%	90%	Average
F1-score	0.82	0.84	0.82	0.8266

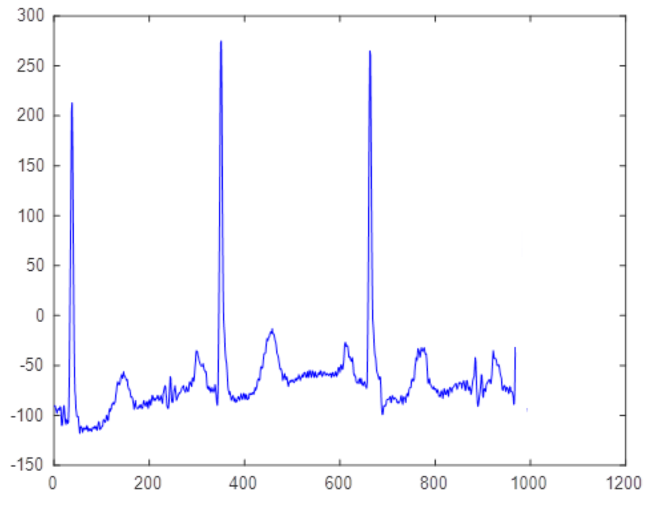

Figure 4. Original ECG signal (before compression)

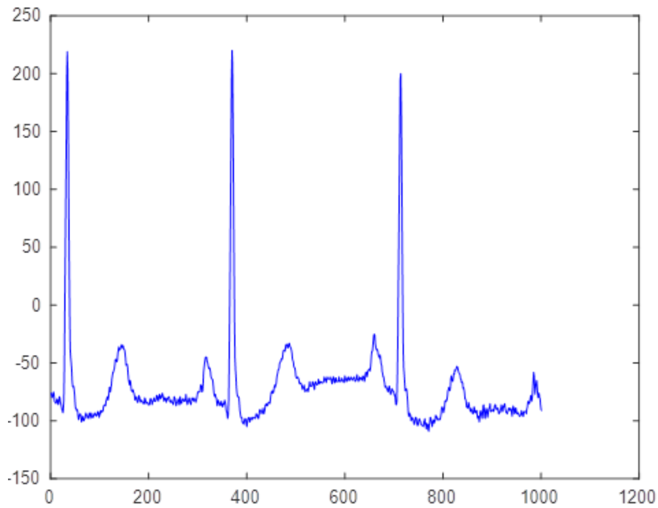

Figure 5. Reconstructed ECG signal (after compression)

6. Conclusions and Future Work

ECG plays an important role in ECG diagnosis; deep learning has become a state of the art ML approach that is widely used in many applications. In this paper, we built a compression system where ECG signal can be recorded and compressed using a novel suggested TinyBERT model, then it can be reconstructed. The quality of the reconstructed ECG is measured by passing it to another deep learning model, 1D-CNN, and achieved a high classification accuracy that is near an ensemble 1D-CNN for classifying 4 types of arrhythmia, proving the effectiveness of our compressor frame. For future work we can apply transfer learning to compress new ECG signals based on pre-trained ECG signal.

Conflict of Interest

There is no conflict of interest.

References

[1] WHO, 2021. Cardiovascular diseases (CVDs).

[2] Aspuru, J., Ochoa-Brust, A., Félix, R., et al., 2019. Segmentation of the ECG signal by means of a linear regression algorithm. Sensors. 19(4), 775.

[3] Eren, H., Webster, J.G., 2018. Telemedicine and Electronic Medicine. CRC Press.

[4] Goyal, M., Tatwawadi, K., Chandak, S., et al., 2018. Deepzip: Lossless data compression using recurrent neural networks. arXiv Prepr. arXiv1811.08162.

[5] Yildirim, O., San Tan, R., Acharya, U.R., 2018. An efficient compression of ECG signals using deep convolutional autoencoders. Cognitive Systems Research. 52, 198-211.

[6] Dasan, E., Panneerselvam, I., 2021. A novel dimensionality reduction approach for ECG signal via convolutional denoising autoencoder with LSTM. Biomedical Signal Processing & Control. 63, 102225.

[7] Zheng, L., Wang, Z., Liang, J., et al., 2021. Effective compression and classification of ECG arrhythmia by singular value decomposition. Biomedical Engineering Advances. 2, 100013.

[8] Rebollo-Neira, L., 2019. Effective high compression of ECG signals at low level distortion. Scientific Reports. 9(1), 1-12.

[9] Zhang, B., Zhao, J., Chen, X., et al., 2017. ECG data compression using a neural network model based on multi-objective optimization. PLoS One. 12(10), e0182500.

[10] Polania, L.F., Plaza, R.I., 2018. Compressed sensing ECG using restricted Boltzmann machines. Biomedical Signal Processing & Control. 45, 237-245.

[11] Abo-Zahhad, M., Ahmed, S.M., Zakaria, A., 2012. An efficient technique for compressing ECG signals using QRS detection, estimation, and 2D DWT coefficients thresholding. Modelling & Simulation in Engineering.

[12] Qian, J., Tiwari, P., Gochhayat, S.P., et al., 2020. A Noble Double-Dictionary-Based ECG Compression Technique for IoTH. IEEE Internet of Things Journal. 7(10), 10160-10170.
DOI: https://doi.org/10.1109/JIOT.2020.2974678

[13] Rajankar, S.O., Talbar, S.N., 2019. An electrocardiogram signal compression techniques: a comprehensive review. Analog Integrated Circuits & Signal Processing. 98(1), 59-74.

[14] Sahoo, G.K., Ari, S., Patra, S.K., 2015. Performance

evaluation of ECG compression techniques. 2015 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT). pp. 1-5.

[15] Elgendi, M., Mohamed, A., Ward, R., 2017. Efficient ECG compression and QRS detection for e-health applications. Scientific Reports. 7(1), 1-16.

[16] Padhy, S., Sharma, L.N., Dandapat, S., 2016. Multilead ECG data compression using SVD in multiresolution domain. Biomedical Signal Processing & Control. 23, 10-18.

[17] Jha, C.K., Kolekar, M.H., 2017. ECG data compression algorithm for tele-monitoring of cardiac patients. International Journal of Telemedicine & Clinical Practices. 2(1), 31-41.

[18] Chowdhury, M.H., Cheung, R.C.C., 2019. Reconfigurable architecture for multi-lead ecg signal compression with high-frequency noise reduction. Scientific Reports. 9(1), 1-12.

[19] Kumar, V., Saxena, S.C., Giri, V.K., et al., 2005. Improved modified AZTEC technique for ECG data compression: Effect of length of parabolic filter on reconstructed signal. Computers & Electrical Engineering. 31(4-5), 334-344.

[20] Ahmeda, S.M., Abo-Zahhad, M., 2001. A new hybrid algorithm for ECG signal compression based on the wavelet transformation of the linearly predicted error. Medical Engineering & Physics. 23(2), 117-126.

[21] Chen, J., Itoh, S., 1998. A wavelet transform-based ECG compression method guaranteeing desired signal quality. IEEE Transactions on Biomedical Engineering. 45(12), 1414-1419.

[22] Tun, H.M., Moe, W.K., Naing, Z.M., 2017. Analysis on ECG data compression using wavelet transform technique. International Journal of Psychological and Brain Sciences. 2(6), 127-140.

[23] Wu, T.C., Hung, K.C., Liu, J.H., et al., 2013. Wavelet-based ECG data compression optimization with genetic algorithm.

[24] Jha, C.K., Kolekar, M.H., 2021. Tunable Q-wavelet based ECG data compression with validation using cardiac arrhythmia patterns. Biomedical Signal Processing & Control. 66, 102464.

[25] Mukhopadhyay, S.K., Ahmad, M.O., Swamy, M.N.S., 2018. An ECG compression algorithm with guaranteed reconstruction quality based on optimum truncation of singular values and ASCII character encoding. Biomedical Signal Processing & Control. 44, 288-306.

[26] Jha, C.K., Kolekar, M.H., 2021. Empirical mode decomposition and wavelet transform based ECG data compression scheme. IRBM. 42(1), 65-72.

[27] Kolekar, M.H., Jha, C.K., Kumar, P., 2021. ECG Data Compression Using Modified Run Length Encoding of Wavelet Coefficients for Holter Monitoring. IRBM.

[28] Shinde, A.A., Kanjalkar, P., 2011. The comparison of different transform based methods for ECG data compression. 2011 International Conference on Signal Processing, Communication, Computing and Networking Technologies. pp. 332-335.
DOI: https://doi.org/10.1109/ICSCCN.2011.6024570

[29] Vaswani, A., Shazeer, N., Parmar, N., et al., 2017. Attention is all you need. International Conference on Neural Information Processing Systems. 30.