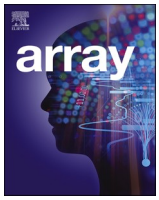
Array 12 (2021) 100076

Contents lists available at ScienceDirect

Array

journal homepage: www.sciencedirect.com/journal/array



Adaptive resampling for data compression

Pesenti Daniel, Morin Lucas, Dias André, Gilles Courret *

Department of Industrial Technologies, University of Applied Sciences and Arts Western Switzerland (HES-SO), Yverdon-les-Bains, 1401, Switzerland

ARTICLE INFO

Keywords:
Big data
Data compression
Local standard deviation
Adaptive sampling frequency
Sparse signals
Multiscale analysis
Green IT

ABSTRACT

With the advent of the digital age, data storage continues to grow rapidly, especially with the development of internet data centers. The environmental impact of this technological revolution has become a problem. As the cost of digital recordings decreases, the amount of unnecessary data stored increases. This paper presents a new algorithm for compressing digital data series, which uses a local measure of relevance based on statistical characteristics. This compression produces non-uniform sampling with a density dependent on the relevance of the data, hence the adaptive feature of the algorithm. It works without any additional input and allows to build a data tree with progressive compression. Such a structure can feed multiscale analysis tools as well as selective memory release solutions for efficient archive management. Tests were carried out on two ideal noise-free signals as well as two real-world applications, namely compression of electrocardiograms retrieved from the PhysioNet database and compression of remote measurements provided by the constellation of ESA's Swarm satellites. Non-sparse type signals have been chosen in order to investigate compression performances in unfavorable conditions. Despite this, the number of samples has been reduced by more than half while maintaining the relevant characteristics of the signals. By reconstructing uniform samplings of the ideal noise-free signals, a measure of the compression error is obtained. Comparing the Fourier transforms of the original and the reconstructed signals, we further allow for future comparative analysis taking into account the ratio between the bandwidth and the sampling frequency of the original signal.

1. Introduction

The ability to record information has always been a key factor in the development of civilizations, as it makes the transfer of knowledge more reliable. The appearance of writing in the 4th millennium BC [1] was a big breakthrough, but data sharing was still hampered by the lengthy and burdensome nature of the reproduction of handwritten data. Thus, the invention of the printing press by Gutenberg during the Renaissance is recognized as a major technological advance that accelerated the development of humanity [2]. Nowadays, some authors consider that the digital turn of society will have a comparable or even greater impact [3,4]. The transition to the digital age has indeed put an end to the information sharing issue by establishing an ease of recording, exchange and processing of data in quantities that seem unlimited, in particular due to the fall in the prices of electronic mass memories in the recent decades [5]. However, this technological revolution has a major impact on global warming [6]. Data centers already account for more than 2 % of global energy consumption [7] and this will increase considerably in the coming years [8]. Today, the climate change demands more efficient technologies in the IT market. To meet the ever-growing need for data, a field of computational theories and tools such as Knowledge Discovery in Databases (KDD) has aroused primary interest in recent years. Data compression algorithms techniques have been developed to reduce the original data without losing the meaning of the information. These techniques allow to boost the productivity of analysts by facilitating search and visualization in databases. Large investments have been made in Big Data, but most of the data collected remains unused. Today's data analysis tools fail to provide fluidity when it comes to Big Data, and user productivity decreases. The environmental issue requires to reduce data waste at source.

In the glaring example of current data acquisition systems (DAQ), the classical sampling techniques usually use a constant frequency, which results in a huge waste of data because the signals in real applications are often irregular. For example, monitoring systems that in the majority of cases measure sparse or chaotic signals consume a great deal of energy due to the need to adjust a constant sample rate in the acquisition chain. Indeed, in order not to miss any event, this constant sampling frequency must be set to its maximum. This generates a large data stream, but of

* Corresponding author.
E-mail address: gilles.courret@heig-vd.ch (G. Courret).

https://doi.org/10.1016/j.array.2021.100076
Received 11 November 2020; Received in revised form 29 June 2021; Accepted 3 July 2021
Available online 31 July 2021

little useful value because nothing special happens most of the time. One of the strategies is the initially specified detection criteria with the problem transposed to an event detection algorithm. Artificial intelligence (AI) offers automatic learning functions for this purpose. However, the cost of maintenance is a major downside related to AI that arises for the customer. Through machine learning, each case becomes indeed unique and the search for the causes of a malfunction can become extremely long.

In this work, we propose to overcome this problem by developing a self-adapting frequency sampler that records data in a tree structure allowing rapid exploration and analysis of Big Data. This study contributes to improving the efficiency of processes in the field of serial data management by addressing challenges related to the cost of storage and their access in memory. This research framework integrates the wasting issue regarding the monitoring systems set at constant sampling frequencies.

1.1. Objective of the work

The purpose of this study is to provide an algorithmic solution for a responsible consumption and production of serial digital data. We propose an implementation of a new digital data compression algorithm, in order to establish an architecture of a data acquisition system with an adaptative frequency for the recording of data in a multiscale tree. We offer, in the future, rapid analysis of very large amounts of data as well as a tool for visualization and exploration in the context of Data Analytics. In a perspective to reduce costs and overconsumption of energy, this work guarantees a first step to converge towards a global and optimal solution in the evolutionary search for robust and reliable solutions for data compression and self-adaptive sampling.

1.2. Organization of the paper

The rest of this work is organized as follows. The next section 2 presents an overview of the state of the art regarding the development of compression techniques in the context of Big Data, positioning our contribution in the broader field of Data Analytics. A detailed description of our algorithm's working principles is provided in section 3. Section 4 proposes the formulation of the algorithmic characteristics as well as the evaluation of the metrics obtained. Section 5 analyzes the results of the tests performed, first on two ideal signals, then on two case studies. Finally, the whole work is concluded in section 6, together with the potential future research directions.

2. Literature review

2.1. Related work

Among the various aspects of the problematic, the most considerable challenges are related to data analysis. In their article [9], Espinosa et al. listed the new issues and challenges for the future of Big Data in the field of Data Analytics. Thus, a valuable skill to develop is the ability to facilitate the research within databases. To achieve this, computer tools like Knowledge Discovery in Databases (KDD) has emerged, whose principles and techniques practices were recently introduced by Bhatia in his work [10]. It encompasses the methods that map Big Data into more compact, more abstract, or more useful forms to enhance analysis [11,12]. At the heart of KDD is the data mining process, including the application of data analysis and the discovery of algorithms mentioned for example by Ganasan in his article [13] whose role is to provide a definition of models from the data. Recently, Menaga and Saravanan [14] have targeted the major disciplines involved in the procedure as being machine learning, AI and statistics. Many applications are emerging in various fields: in the field of healthcare where it improves the prediction of many diseases and helps physicians in the diagnosis [15–17], in climate change studies [18], in education systems [19,20], in management [21], in market analysis [22,23], in sports data analysis [24,25], in scientific research [26,27] and many more [28–30].

In this context, compression techniques have appeared at least since Claude Shannon established in 1948 the foundations of information theory [31]. By defining the extent to which information can be removed from the original data without losing its core meaning, data compression algorithms can then be developed. In his book [32], Sayood explains that compression algorithms can be categorized as either lossless or lossy. The trade-off is that generally a lossy compression will be able to compress more than a lossless one. In their article [33], Khan et al. evaluated lossy algorithms as a better alternative if losing some information is acceptable in order to enhance the compression. The body of work in Ref. [34] explored several types of algorithms. They are often based on operators such as the fast Fourier transform [35,36], the discrete cosine transform [37] or the discrete wavelet transform [38]. In their work [39], Sharma et al. provide near-lossless compression techniques to remove data redundancy, in which the difference between the reconstructed and original signals is guaranteed not to exceed a user-defined value.

Another aspect of the problem concerns exploration in a database. A multi-scale or multi-level approach can be an effective modeling methodology. A tree structure is a powerful tool for organizing multiple data objects in terms of hierarchical relationships. For example, the Ref. [40,41] propose hierarchical data applications models. This type of structure features notably a quick and efficient gathering of data. It can feed a multiscale graphical tool, helping to find patterns of different characteristic scales, hence leading to coarse-grained modeling like for example in molecular biology [42–44] or in biomedical engineering [45,46].

To date, some systems generate a very large amount data, for instance monitoring applications that require continuous acquisition in order to spot unpredictable events, in the field of radar signal processing [47,48] or in the field of medical surveillance [49–51]. The problem addressed here is the management of the sampling frequency. The authors of the body of work in Refs. [48,52] have explored some applications where the activity is irregular, and where it is possible to observe a sparse representation of the monitoring signals where all segments of the signal do not necessarily contain the same amount of information. As Wang et al. point out in their article [53], this random and non-periodic component in time is a central problem in the research for an effective way to save the costs for data storage. In the example of sparse signals containing sudden bursts of oscillations, they show long and flat parts while waiting for the next event to occur. With a constant sampling rate, a large amount of data is generated on this flat part whose only meaning is that nothing in the sampling bandwidth has happened. In classical acquisition systems, while a constant and high sampling rate is required for rapidly varying pulses at the time of all new events, no sampling would be necessary between them if the waiting time was known. Moreover, the bandwidth of the signal must be known to apply the Nyquist-Shannon sampling theorem [54,55], thus ensuring a correct reconstruction. But this may require restrictive assumptions in the case of aperiodic or sparse signals, as mentioned by Jiao et al. in their work [56]. To overcome this problem, time-frequency domain analysis can be used to show how the frequency content of the signal changes over time. The most common tool used for this is probably the wavelet transform [57]. However, there are several functional decomposition techniques, some of which are highlighted and compared by the authors of Ref. [58]. Thus, the user in this methodology must make an appropriate choice of scales and basis of functional decomposition among a large number of possibilities. The method proposed in this publication removes these constraints, facilitating the applications.

2.2. Contributions

This paper introduces a new digital data compression algorithm. Our approach is hybrid and versatile, because its software level can be

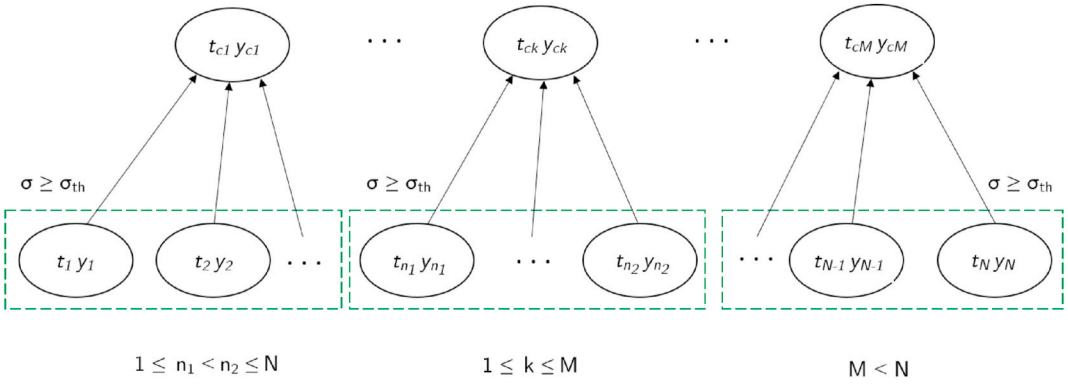


Fig. 1. Diagram of the compression process, showing the affiliation between nodes (the samples of current segment) and their children (the nodes of the lower level that are merged). The current segment is extended until its standard deviation exceeds the cutoff threshold σ_{th}.

classified in the event-based sampling category but no special data acquisition hardware is needed. Compression is lossy, although it shares some similarities with near-lossless compression since the user can control the degree of compression. Also, as mentioned earlier, regular sampling of sparse signal generates a lot of unnecessary data. Uniform sampling is then seen a posteriori as a penalizing constraint. In contrast, our algorithm essentially performs non-uniform downsampling which results in a uniform distribution of relevance.

Thereby, our first contribution is the implementation of a structured data tree with a level-by-level progression of compression, allowing data management directly at several scales and the progressive reduction of archive files. We present its process principle in section 3.1. At the current stage of development, only one-dimensional signals can be processed. The second contribution is a statistical approach based on a local standard deviation for the implementation of the non-uniform sampling during compression, meaning no machine learning or AI techniques are used. Non-uniform or self-adaptive sampling is implemented thanks to the central parameter of the local standard deviation which measures the relevance of the sample and which is very resilient to noise. This adaptive compression technique also exploits the quality index parameter calculated from the data. This functionality is exposed in section 3.2.

To our knowledge, the work presented in this research is a further attempt at near-lossless hybrid compression through non-uniform sampling, maximizing energy efficiency as well as data analyst productivity and avoiding the previously mentioned AI drawbacks. Several keys of the characteristics and metrics to measure the algorithm performances are evaluated by simulation experiments in section 4, including compression ratio, relative mean error of a compressed level, space-saving, compression gain, signal-to-noise ratio, maximum absolute distortion, signal segmentation, local sample rate and mean number of children per node. Finally, we present the results of the tests of our compression algorithm on the signals provided by uniform sampling. Two tests with real world signals were carried out using a threshold determined with the heuristic method: a normal ECG for a human at rest extracted from the Physionet database and a signal measurement provided by the satellites of a European Space Agency test mission.

3. Algorithm breakdown

3.1. Tree structure data compression

The compression of digital data series aims not only to save memory space, but also to filter out less relevant or unnecessary information such as the measurement noise. Hence, the algorithm acts as a low pass filter aiming to facilitate the exploitation of Big Data.

Fig. 1 schematizes the proposed compression tree process. This tree structure reduces data storage when the original sampling is uniform. Arguably, non-uniform sampling introduces additional data storage because timestamps are no longer computable from just two real numbers: the sampling rate and the first timestamp. These two data are usually stored in double-precision floating-point format so that the resulting series has sufficient resolution. In our case, storing all timestamps is also not mandatory if the tree is saved, storing the number of children of each node. Each timestamp can indeed be calculated from the tree with the same precision since the constant sampling rate of the original signal and its first timestamp are recorded. The number of children can be stored in integer format, which uses only a small amount of memory.

Therefore, we consider the set of points $S = \{(t_1; y_1), \ldots, (t_N; y_N)\}$ be the N samples of the digital input signal, and $S_c = \{(t_{c1}; y_{c1}), \ldots, (t_{cM}; y_{cM})\}$ be the M samples of the output signal, with $M < N$. To initiate the compression, a first segment of the input signal is taken, containing only the first two samples $S_n = \{(t_1; y_1), (t_2; y_2)\}$ Its standard deviation σ is compared to a predefined cutoff threshold σ_{th}. If $\sigma < \sigma_{th}$, then the next sample, $(t_3; y_3)$, is appended to S_n and the process of comparison is repeated as well as the appending (cf. Equation (1)) until the threshold crossing. In this case, the segment is cut off from the signal and its centroid $(t_{c1}; y_{c1})$ (cf. Equations (2) and (3)) is appended to the compressed sampling, which is its only point for the moment. Then a new segment is opened that initially contains only the next two points from the original sampling. The whole process is repeated until the end, as in Fig. 1 showing an arbitrary item $(t_{ck}; y_{ck})$ of the compressed sampling.

$$S_n = \{(t_{n1}; y_{n1}), \ldots, (t_{n2}; y_{n2})\} \tag{1}$$

$$\bar{y}_n = \frac{1}{n_2 - n_1 + 1} \sum_{i=n_1}^{n_2} y_i \tag{2}$$

$$\bar{t}_n = \frac{1}{n_2 - n_1 + 1} \sum_{i=n_1}^{n_2} t_i \tag{3}$$

$$\sigma_n = \sqrt{\frac{1}{n_2 - n_1 + 1} \sum_{i=n_1}^{n_2} \left(y_i - \bar{y}_n\right)^2} \tag{4}$$

The comparison with the threshold as well as the resulting actions can be implemented in a simple standard logic operation. (assuming $n_2 \neq N$ *or* $N - 1$):

$$\sigma_n \geq \sigma_{th} \Rightarrow \begin{cases} t_{ck} = \bar{t}_n \\ y_{ck} = \bar{y}_n \\ S_n = \left\{\left(t_{n_2+1}; y_{n_2+1}\right), \left(t_{n_2+2}; y_{n_2+2}\right)\right\} \end{cases} \tag{5}$$

$$\sigma_n < \sigma_{th} \Rightarrow \mathrm{S_n} = \mathrm{S_n} \cup \left\{\left(t_{n_2+1}; y_{n_2+1}\right)\right\} \tag{6}$$

However, the standard deviation does not take into account a

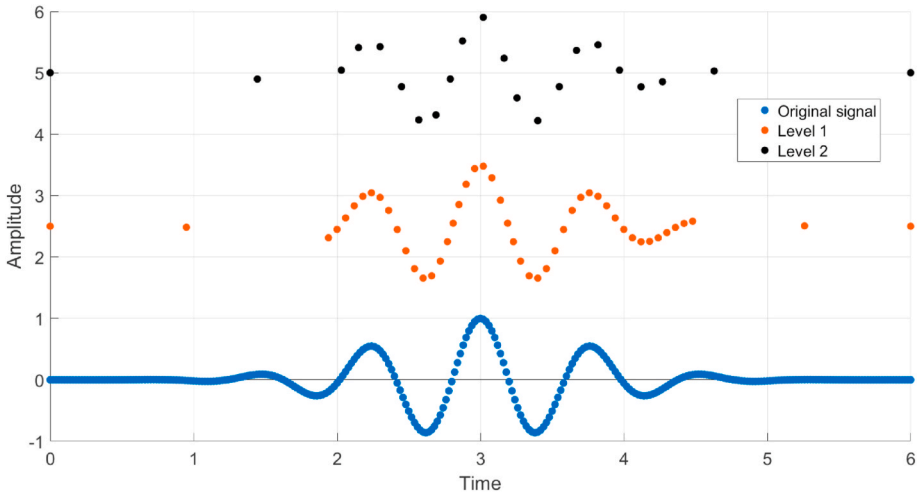


Fig. 2. A uniform sample (below) with 2 compressions shifted upward (center and top).

possible ramp in the segment considered, S_n. A refinement is therefore obtained by considering the signal rate of change in average value over the segment. This is achieved by replacing the mean value $\overline{y}_n$ in Equation (4) by the linear regression model $\widehat{y}_{ni}$ calculated as follows:

$$\widehat{y}_{ni} = a_n t_i + b_n \quad n_1 \leq i \leq n_2 \tag{7}$$

$$a_n = \frac{\sum_{i=n_1}^{n_2} \left(t_i y_i - \overline{y}_n \overline{t}_n \right)}{\sum_{i=n_1}^{n_2} \left(t_i^2 - \overline{t}_n^2 \right)} \tag{8}$$

$$b_n = \overline{y}_n - a_n \overline{t}_n \tag{9}$$

So equation (4) becomes:

$$\sigma_n = \sqrt{\frac{1}{n_2 - n_1 + 1} \sum_{i=n_1}^{n_2} \left(y_i - \widehat{y}_{ni} \right)^2} \tag{10}$$

Some computational time can be saved by excluding segments of only two samples from the refinement, as there would then be no deviation from linear regression.

3.2. Adaptive resampling

The compression algorithm is applicable on non-uniform sampling (variable sampling rate), so that iteration can be performed to build a tree data structure where each level further compacts the initial signal.

Fig. 2 shows two iterations applied on a wave packet, that spaces the sampling points according to the signal variation. They are distributed so that the sampling focuses on the most curved parts. In this regard, the measure of relevance is related to the deviation from a straight line. Signal derivatives have been proposed as a suitable attribute, but have the considerable drawback of amplifying measurement noise, as Algabroun explains in his article [59]. In the design of our algorithm, we apply an alternative based on the standard deviation (cf. Equation (10)), therefore very noise-resilient, as mentioned before.

Thenceforth, an important question to discuss is how to set the threshold value σ_{th}. As mentioned earlier, it can be freely defined by the user. This is a useful feature if he can access the application specifications. For instance, a threshold proportional to the noise level at the output of the measurement chain would certainly be a good approach in the case of measurement data. In cases where no sufficient specification is available or cannot be obtained, a default solution is formulated now, allowing a value to be assigned to σ_{th} only from the input signal. This method is therefore qualified as adaptive.

Taking advantage of some metrics of the tree compression algorithm

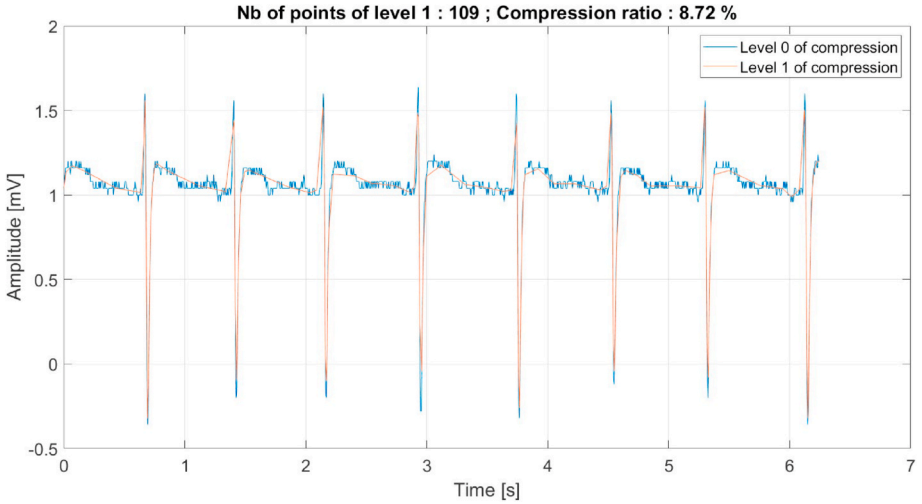


Fig. 3. ECG taken from the PhysioNet database [60]. The original number of samples is 1250 (sampling rate is 200 Hz).

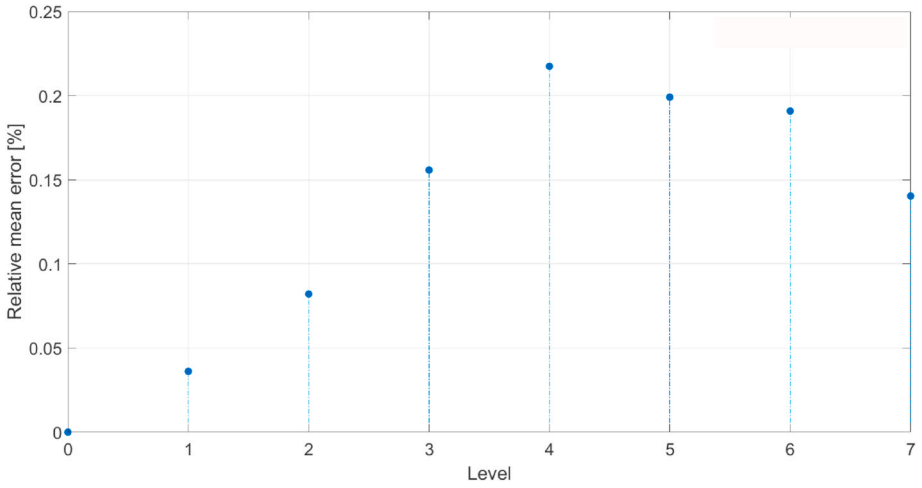


Fig. 4. Relative mean error e_R^k of the ECG signal of Fig. 3, cf. Equation (16).

presented in the next section 4.1 – namely CR^k and ε_r^k - an arbitrary function $Q^k(\sigma_{th})$ is defined and will serve as a quality factor for a given level k and a given σ_{th} (referring to Equations (14) and (16)):

$$Q^k(\sigma_{th}) = \frac{1}{CR^k(\sigma_{th})\varepsilon_r^k(\sigma_{th})} \tag{11}$$

Ideally, the algorithm should grant a low compression ratio as well as a low error. This leads to a high value of Q^k, which allows to find the value of σ_{th} inducing a maximum of Q^k. To this end, let $S^0 = \{(t_1^0; y_1^0), \ldots, (t_{N_0}^0; y_{N_0}^0)\}$ be the signal to be compressed. First, all possible values of the standard deviation are calculated as follows:

$$\begin{cases} Y_1^0 = \left[y_1^0\ y_2^0\right] \quad \sigma_1 = std\left(Y_1^0\right) \\ Y_2^0 = \left[y_1^0\ y_2^0\ y_3^0\right] \quad \sigma_2 = std\left(Y_2^0\right) \\ \ldots \\ Y_{N_0-1}^0 = \left[y_1^0\ y_2^0 \ldots y_{N_0}^0\right] \quad \sigma_{N_0-1} = std\left(Y_{N_0-1}^0\right) \end{cases} \tag{12}$$

By doing so, we can delimit the domain of variation of σ_{th}:

$$\sigma_{min} = \min\left(\sigma_1,\ \sigma_2,\ \ldots,\ \sigma_{N_0}\right) \text{ and } \sigma_{max} = \max\left(\sigma_1,\ \sigma_2,\ \ldots,\ \sigma_{N_0}\right) \tag{13}$$

Then, the compression ratio $CR^k(\sigma_{th})$, relative mean error $\varepsilon_r^k(\sigma_{th})$ and quality index $Q^k(\sigma_{th})$ are calculated for a series of threshold values scanning the interval. $[\sigma_{min}, \sigma_{max}]$.

In addition, for monitoring systems which are predominantly inactive and where an event can occur after a long period of time, our algorithm provides a limit to the number of points per segment N. Indeed, without this delimitation, the number of points in the current segment could grow infinitely. When an event occurs, the algorithm could no longer react since the division by a too large number would impose a standard deviation threshold tending towards 0. The limit of N could be configured according to the hardware specifications, i.e. its memory buffer register. In this study, it is fixed to the cardinal of the input signal (no delimitation).

4. Metrics and characteristics evaluation

In this section, we visualize the metrics and characteristics performed by the algorithm using as an example the electrocardiogram (ECG) signal shown in Fig. 3. In section 4.1, Figs. 5 and 6 show an overview of metrics obtained with the normalized threshold σ_{th}/σ_{max} on the x-axis, facilitating the analysis of a compression level or of the tree as a whole.

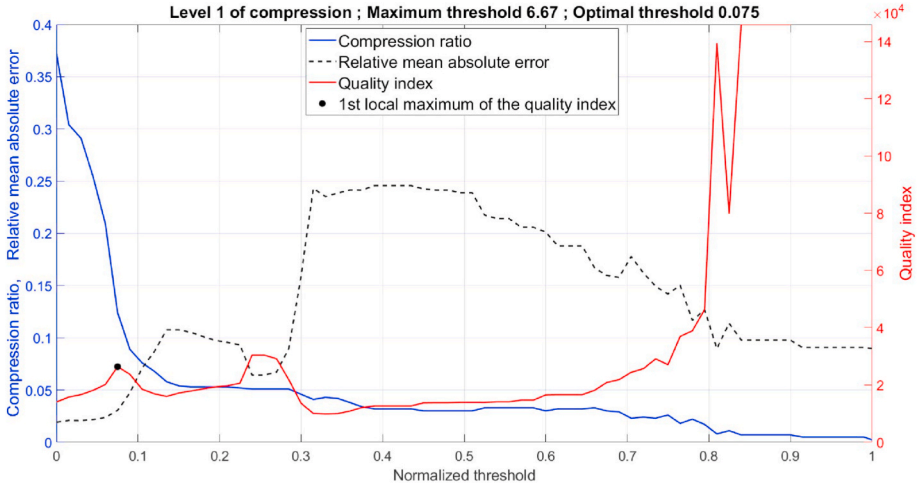


Fig. 5. Q^1, ε_R^1, and CR^1 of level 1 compression of the ECG signal of Fig. 3 versus the normalized standard deviation threshold $(\sigma_{th}/\sigma_{max})$. The black dot indicates the chosen local maximum of Q^1. (The curve Q^1 has been truncated for readability.).

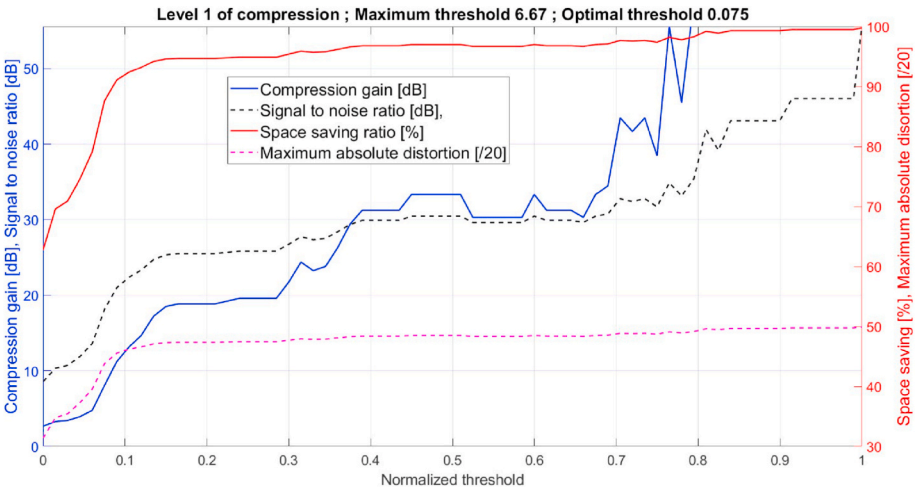


Fig. 6. SS^1, CG^1_{dB}, SNR^1_{dB}, and MAD^1 (cf. Equations (17)–(20)) of level 1 compression of the ECG signal of Fig. 3 versus the same normalized threshold as in Fig. 5 (The curve of CG^1_{dB} has been truncated for readability).

4.1. Algorithm metrics

The Figs. 5 and 6 show a number of key metrics for evaluating the performance of our data compression algorithm. The analysis of each measure were calculated from the definitions that follow.

(i) Compression ratio:

For a level k, the compression ratio plotted above in Fig. 5 is defined as the following ratio

$$CR^k = \frac{S^k}{S^0} \tag{14}$$

with

$S^k =$ *Number of samples at the current level*

$S^0 =$ *Number of original samples* (*level* 0)

(ii) Relative mean error of a compressed level:

Using linear interpolation of the compressed level k, an error measure ε^k with respect to the original signal (level 0) is defined:

$$\varepsilon^k_i = y^k_i - y^0_i\,; \quad 1 \le i \le N. \tag{15}$$

Then taking the average of the absolute value of ε^k and dividing by the mean absolute value of level 0, a relative mean error ε^k_R illustrated in Fig. 5 is furthermore defined:

$$\varepsilon^k_R = \frac{\sum_{i=1}^{N} |\varepsilon^k_i|}{\sum_{i=1}^{N} |y^0_i|} \tag{16}$$

Note that the case of a zero denominator is excluded because the whole signal would then be zero. It is also noteworthy that the relative mean error does not always increase from one compression level to the upper next, as can be seen in Fig. 4.

In Fig. 5, it is noteworthy that Q^1 rises significantly when $\sigma_{th}/\,\sigma_{max}$ approaches 1. CR^1 converges towards low values on the right end of the x-axis and ε^1_R decreases, which leads to the highest value of Q^1 overall. Hence, following the absolute maximum value of Q^1 is not a suitable option as the number of levels in the tree would be too low. Instead, we chose the first local maximum found by gradually increasing the threshold from σ_{min}, making a heuristic trade-off between the error and the ratio of compression. The threshold value corresponding to Q^1 is hence used for compression.

For all the signals tested so far (cf. section 5), Q^1 is an increasing function of the threshold at the lower bound of the variation domain with at least one local maximum. We therefore believe that this heuristic has wide applicability. In the case of the ECG signal of Fig. 3, it results in a normalized threshold value of 0.075, which leads to a compression ratio of 12.4 % and a value of 0.0305 for ε^1_R which are decent results for such data.

In Fig. 6, we analyze the correlation between metrics like the space saved by the algorithm, its compression gain, the signal-to-noise ratio as well as the maximum absolute distortion between the original signal and the compressed signal of level 1. By still considering the same local maximum of Q^1 0.075 obtained in Fig. 5, the algorithm performs as follows. The space-saving SS^1 increases significantly as it approaches the local maximum of Q^1 that we have chosen, before quickly reaching a plateau close to 100 %. The maximum absolute distortion shows also a plateau over the same interval. As this space-saving index is just the complement to 1 of the compression ratio CR^1 measured previously, the result is again satisfactory. Also, as is the case with space-saving, the compression gain CG^1_{dB} and the signal-to-noise ratio SNR^1_{dB} both show quite similar overall behavior, with a greater sensitivity for the first metric. Both seem to be correlated with each other. This result is relevant since the algorithm, as it compresses the data, reduces the noise level at the output of the measurement chain.

An important observation to make, in the case of this ECG signal, concerns the second phase of the test, on the right of the chosen local maximum of Q^1, 0.075 (cf. Fig. 5). All metrics present there a stationary phase. Hence, we believe that it would be quite advisable to stop the adaptive process after finding the first local maximum of Q^1. Of course, all the data are presented for general analysis. These metrics are calculated with the following definitions:

(i) Space-saving:

Conventionally, the space-saving is given by the relation with the compression [61]. That's why, in our case, we choose to define it, for a compression level k, as:

$$SS^k = 1 - CR^k = 1 - \frac{S^k}{S^0} \tag{17}$$

Naturally, this measure of our algorithm could be the subject of further study. At this stage, we do not go into details on this aspect because it would require an overly ambitious extension of this work.

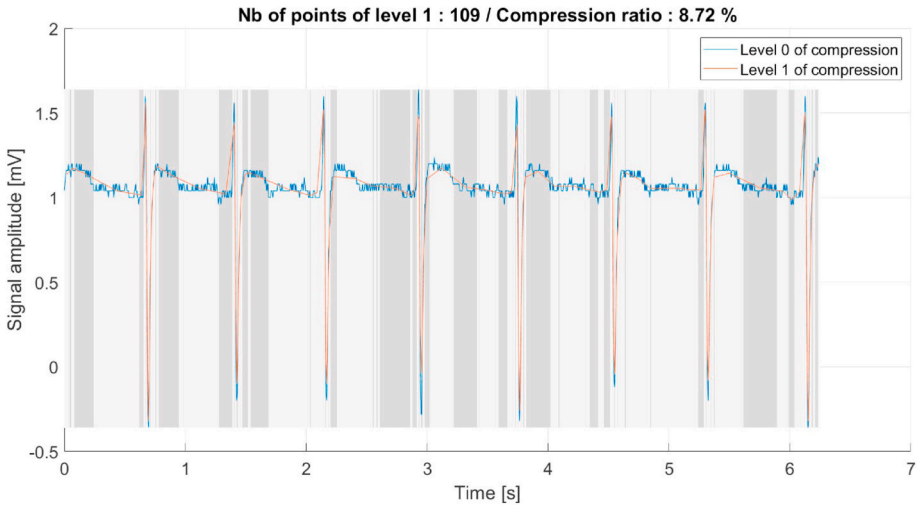


Fig. 7. Segmentation of the ECG signal of Fig. 3. Each vertical band (two tone grey background) contains only one sample of the downsampling, plotted in red. The level below is also plotted in blue. One can see the width of the segments is larger between the pulses, meaning a lot more points are condensed in these time intervals.

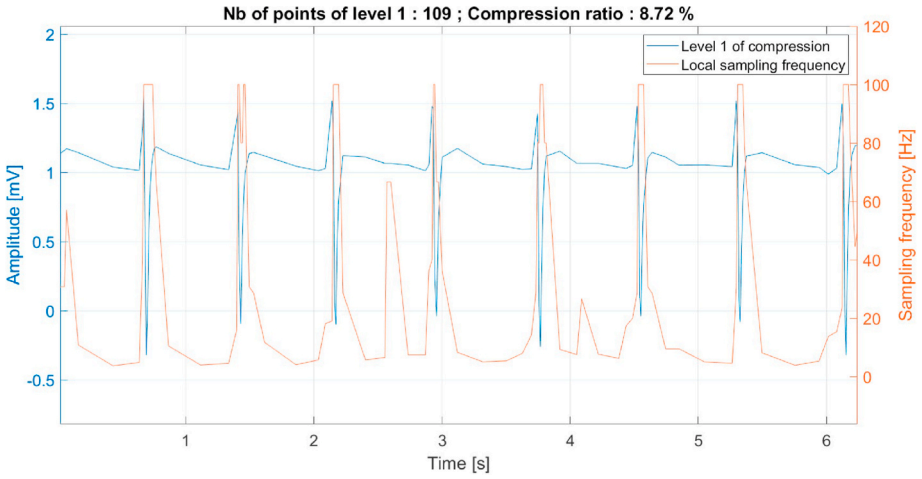


Fig. 8. Local sampling rate of level 1 compression of the ECG signal of Fig. 3.

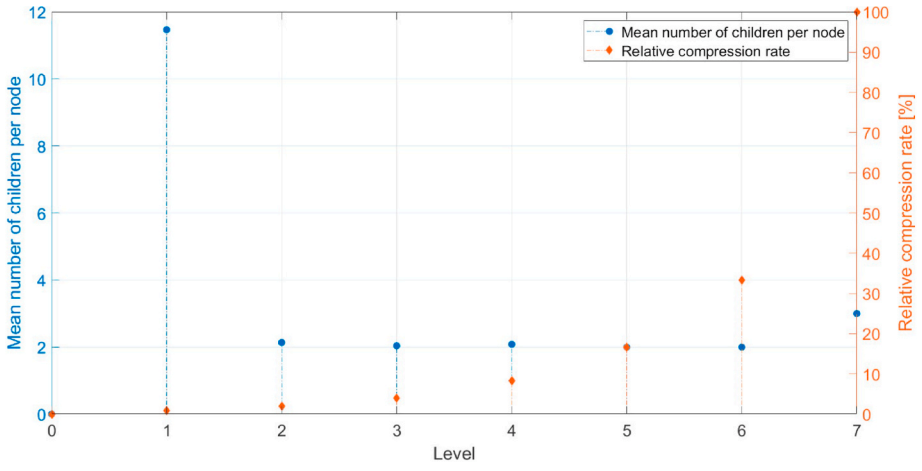


Fig. 9. Mean number of children per node $\overline{N}_c^k$ and relative compression rate CR_r^k of the ECG signal of Fig. 3 (cf. Equations (22) and (23)).

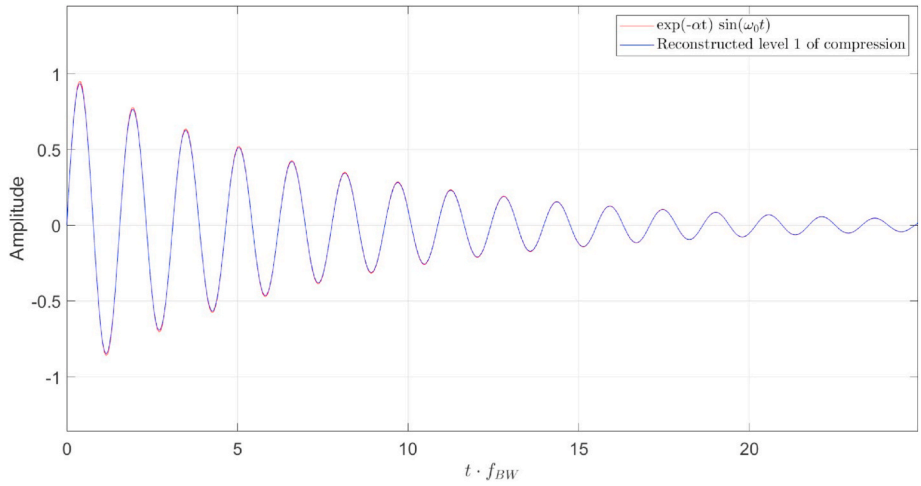


Fig. 10. Comparison of a damped sine wave with its reconstruction from level 1 compression. $CR^1 = 36\%$.

(ii) Compression gain:

Still for a level k, the compression gain in decibels is defined as the following:

$$CG^k_{dB} = 20\log_{10}\frac{S^k}{S^0} \tag{18}$$

(iii) Signal-to-noise:

The signal-to-noise ratio standard definition is conventionally indicated by the relativeness of denoised signal corresponding to original signal [62]. Here, we use the relative average error established above as a measure of the signal background noise. Thus, the signal to-noise ratio in decibels, for a current level k, is defined as:

$$SN^k_{dB} = 10\log_{10}\frac{S^0}{\varepsilon^k_R} \tag{19}$$

(iv) Maximum absolute distortion:

The local distortion measure is frequently used to quantify the error between the original signal and the reconstructed signal [63,64]. In this way, for a level k, the local maximum absolute distortion (or peak distortion) is defined by (cf. Equation (15)):

$$MAD^k = \max\left(\left|\varepsilon^k_i\right|\right) \tag{20}$$

4.2. Algorithm characteristics

The graphics below (cf. Figs. 7–9) illustrate some characteristics performed by the algorithm still in the case of the electrocardiogram (ECG) signal example shown in Fig. 3.

(i) Segmentation of the signal:

The plot function of the Fig. 7 presents the segmentation throughout the ECG signal by colored vertical bands. To avoid gaps, the smaller boundary is placed in the middle between the first child of the current segment and the last child of the previous segment, and the greater boundary in the middle between the last child of the current segment and the first child of the next segment.

(ii) Local sampling frequency

The local sampling frequency parameter is suitable for monitoring signal segmentation. For an arbitrary level k composed of the time vector $t^k = [t^k_1, t^k_2, \ldots, t^k_N]$, the vector $f^k = [f^k_2, f^k_3, \ldots, f^k_N]$ is constructed:

$$f^k_{i+1} = \frac{1}{\Delta t^k} = \frac{1}{t^k_{i+1} - t^k_i}\ [Hz]\ 1 \le i \le N-1 \tag{21}$$

In order to match vector lengths, an additional element f^k_1 equal to f^k_2 is arbitrarily inserted at the beginning of the frequency vector. As expected, the sampling rate periodically peaks with the heart pulses, reaching an upper limit of 100 Hz, which is half the original sampling rate (cf. Fig. 8). This limit matches the minimum compression ratio of 50 % since each node has at least two children, for it takes at least two values to calculate a standard deviation.

(iii) Mean number of children per node:

This parameter is suitable for measuring the compression of one level. It is defined as $\overline{N}^k_c$ in Equation (22) for an arbitrary level k composed of N nodes, with N^k_{ci} being the number of children of the node i.

$$\overline{N}^k_c = \frac{1}{N}\sum_{i=1}^{N} N^k_{c_i} \tag{22}$$

Moreover, by dividing by the total number of nodes of the level just below, therefore the level k-1, we obtain a general indicator of compression, allowing for example to compare the compression between different levels. This parameter is called the relative compression rate CR^k_r (cf. Equation (23)).

$$CR^k_r = \frac{\overline{N}^k_c}{Number\ of\ samples\ of\ the\ previous\ level} \tag{23}$$

In Fig. 9, we can notice $\overline{N}^k_c$ drops to about 2 after level 1, reducing interest in the tree. This may be due to taking the same threshold value (σ_{th}) to build each level. The issue can be overcome by varying the threshold from one level to another, for example by reapplying the adaptive method described in section 3.2.

5. Tests, simulation analysis and results

We perform compression tests with adaptive resampling on signals provided by uniform sampling. This section starts by testing with ideal signals, i.e. whose continuous-time Fourier transform have an analytical expression, hence allowing to set its bandwidth (denoted f_{BW}). By reconstructing the compressed signal in uniform sampling, the Fast Fourier Transform (FFT) can be calculated and compared to the original

Fourier transform. The sampling frequency, denoted f_s, is fixed at 20 times f_{BW} in reference to a practical engineering rule [65]. Two tests with real world signals are also performed: a normal ECG for a human at rest extracted from the Physionet database [60] and a signal measurement provided by the satellites of the European Space Agency's Swarm mission [66]. All these signals have been chosen so as not to contain any straight portion because we want to test the compression under unfavorable conditions, in order to obtain the low limits of its performance.

5.1. Tests with ideal signals

The two ideal noise-free signals used for testing are chosen from the most common in engineering and physics. The compression is limited to level 1. The signal is reconstructed by cubic spline interpolation in a uniform sampling with the level 0 rate (f_s). The amplitude of the FFT is then calculated to make a comparison between the two levels 0 and 1. The continuous time Fourier transforms are used to set the signal bandwidth f_{BW}. The number of samples is fixed at 500.

(1) The first ideal signal is a sine wave with exponential damping: $y_1(t) = e^{(-\alpha t)}\sin(\omega_0 t)$; $\alpha = 2u^{-1}$; $\omega_0 = 20\pi\ rad/u$; $f_{BW} = 15.53\ u^{-1}$; $f_s = 310.6\ u^{-1}$, with u an arbitrary unit of time. The Laplace transform of $y_1(t)$ is found in Ref. [67]. f_{BW} is computed with the software Matlab ('bandwidth' command). The peak value in the spectrum of the oscillations after reconstruction is slightly lower (cf. Fig. 11), but the result is very satisfactory. The error is even better bounded than in the case of a pure sine wave (2% against 3.5%).

The reconstruction is very close to the original sampling, so much that they can hardly be distinguished in Fig. 10. The difference remains less than 0.5 % over the whole sampling, in absolute value.

(2) The second ideal signal is a Gaussian wave packet $y_2(t) = \frac{1}{\sigma\sqrt{2\pi}}\exp\left(-\frac{(t-t_0)^2}{2\sigma^2}\right) sin(\omega_0(t-t_0))$; $\omega_0 = 20\,\pi\ rad/u$; $\sigma = 0.3$; $t_0 = 1.0396\ u$; $f_{BW} = 12\ u^{-1}$; $f_s = 240\ u^{-1}$

An analytical expression for the Fourier transform of Gaussian waves is given in Ref. [68]. Its modulus is computed for $y_2(t)$ to find f_{BW} (cf. Fig. 12). f_{BW} is around $12\ u^{-1}$, which brings the sampling rate to $240\ u^{-1}$.

After reconstruction, only a small disturbance can be observed, in the interval [0–3] of the adimensional time (cf. Fig. 13). This segment of signal was condensed into a single sample during compression, and reconstruction by the cubic spline technique produced this artifact

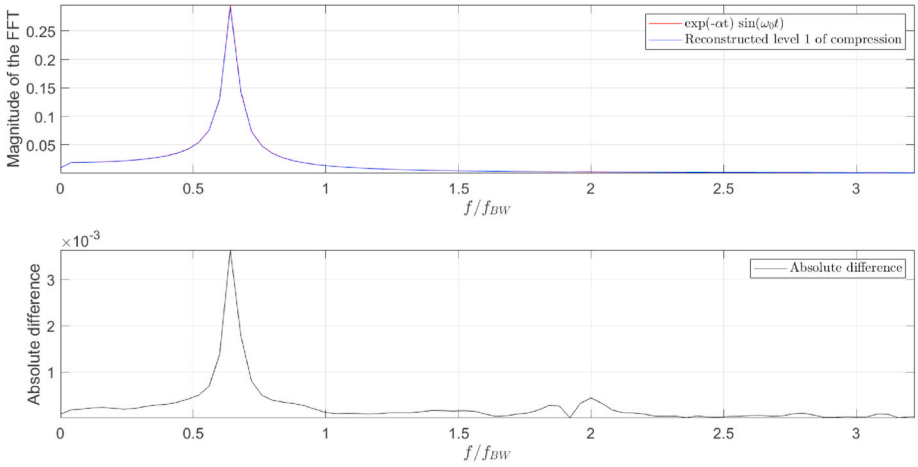


Fig. 11. Above: magnitude of the frequency spectrum of the damped sine wave of Fig. 10. Below: difference in FFT magnitudes between levels 1 and 0.

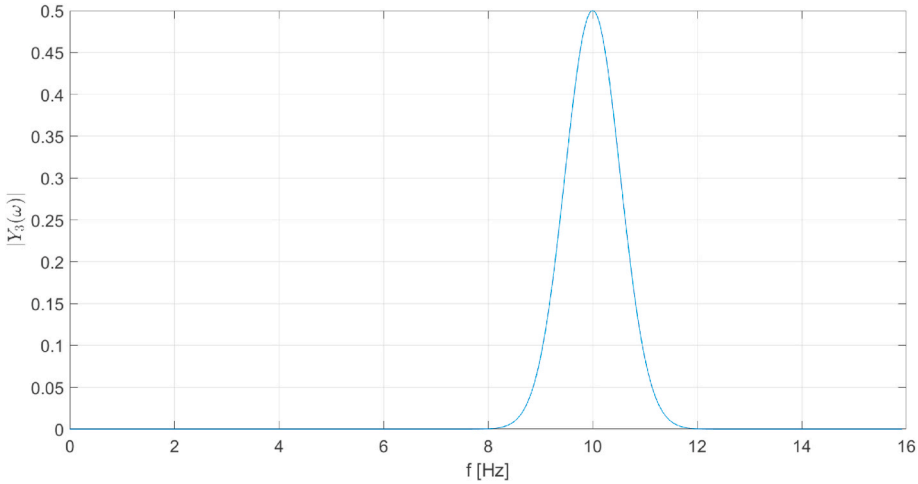


Fig. 12. Magnitude of the analytical Fourier transform of a Gaussian pulse [68].

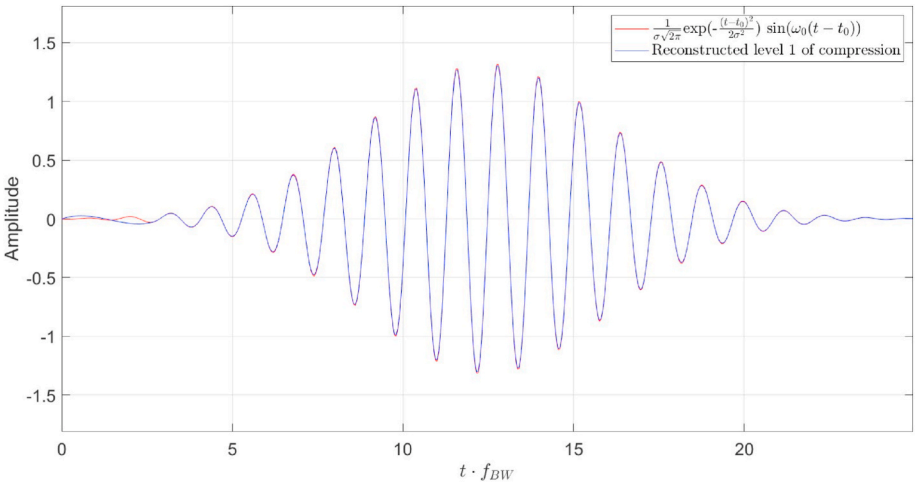


Fig. 13. Comparison of a Gaussian pulse with its reconstruction from level 1 compression. $CR^1 = 40$ %.

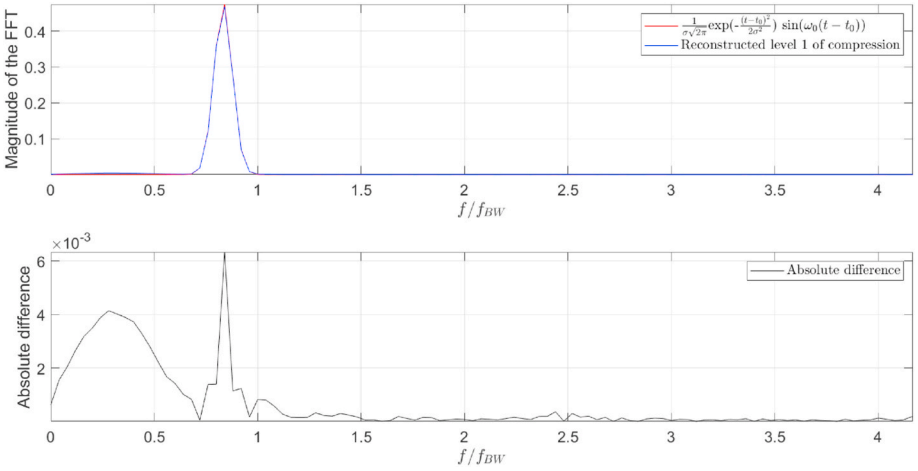


Fig. 14. Above: magnitude of the frequency spectrum of the Gaussian pulse of Fig. 13. Below: difference in FFT magnitudes between the level 1 and 0.

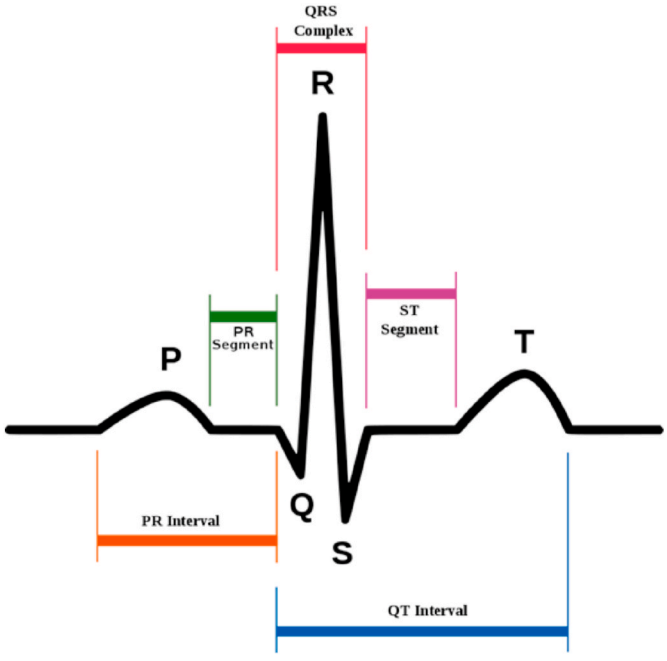


Fig. 15. Main features looked for in an ECG [69].

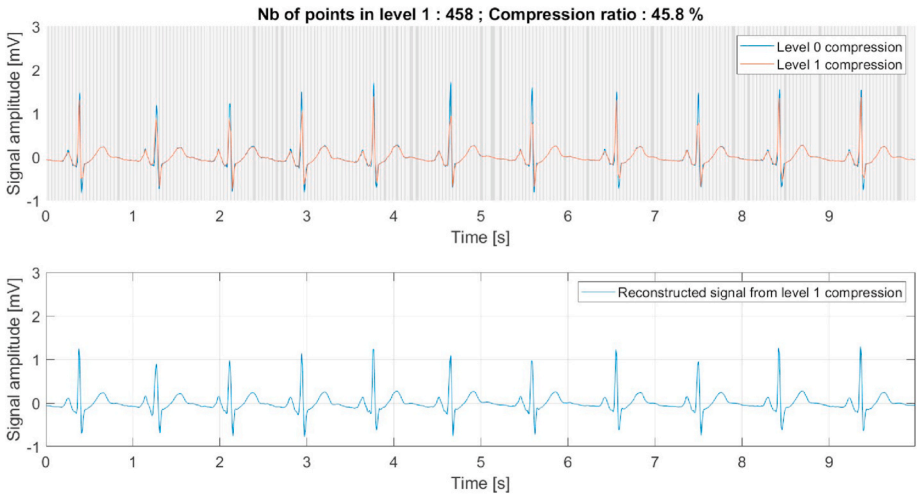


Fig. 16. ECG example 1, large scale view.

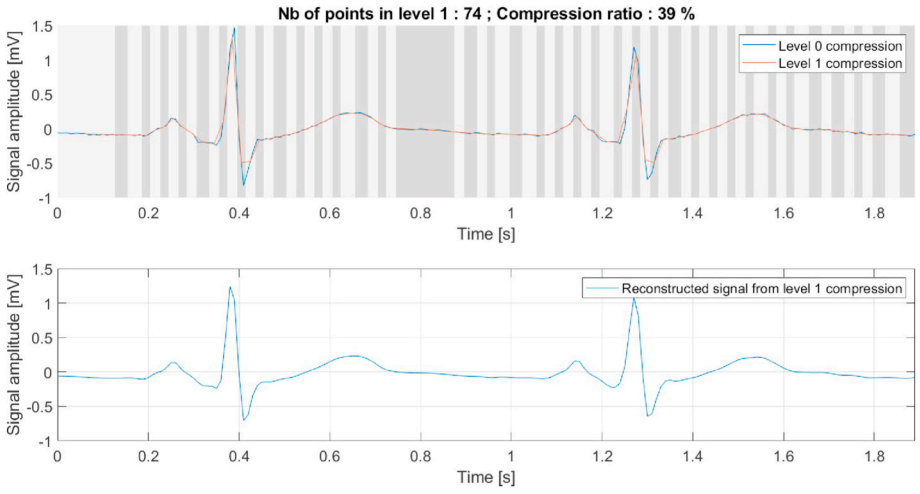


Fig. 17. Two-beat portion of ECG example 1.

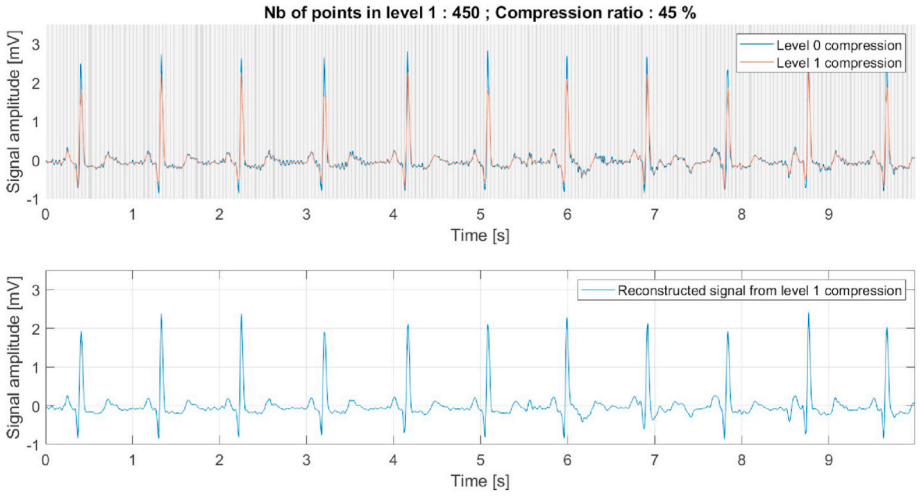


Fig. 18. ECG example 2, large scale view.

wave.

The spectrum of the reconstructed signal is very close to the original (cf. Fig. 14, above). We magnified by a thousand to observe the difference between the two spectra (cf. Fig. 14, below). The hill at the adimensional peak frequency of about 0.3 is due to the above mentioned artifact of reconstruction. In the whole spectrum, the error is less than 1

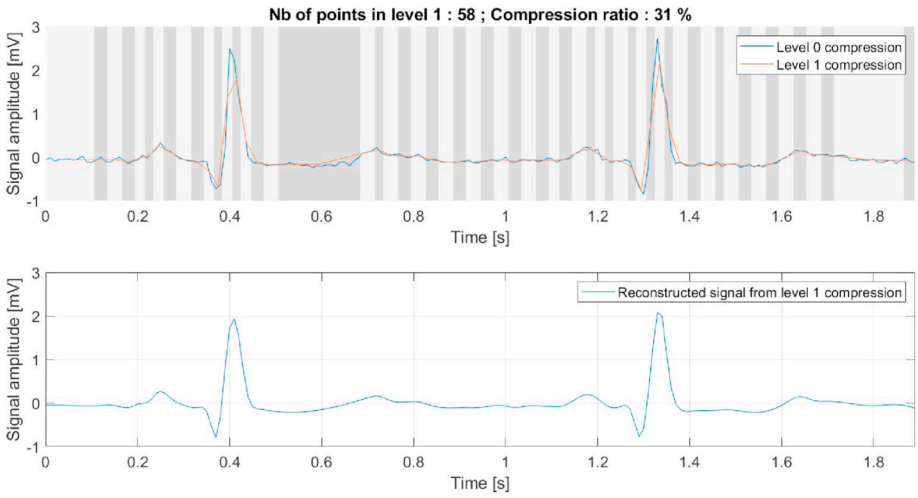


Fig. 19. Two-beat portion of ECG example 2.

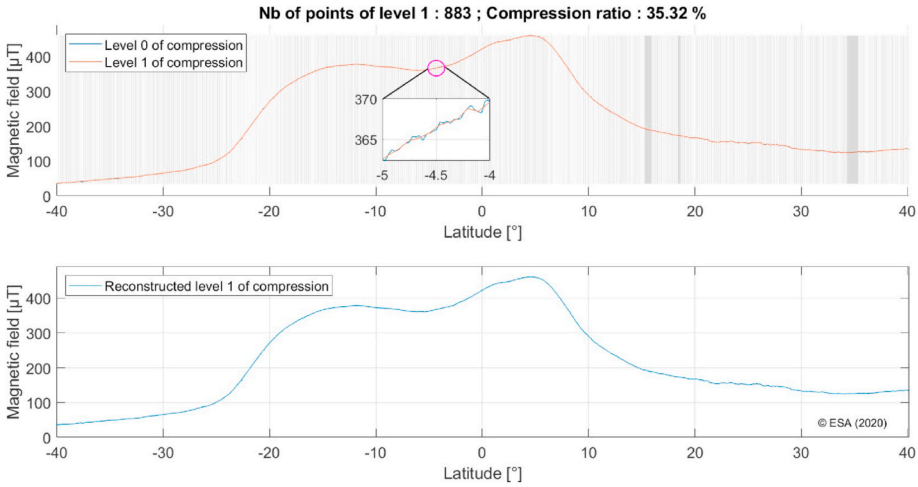


Fig. 20. ESA Swarm mission measurement data on Earth's magnetic field as a function of latitude. The [-5°, −4°] interval is zoomed in to show small fluctuations in the original signal, that the compression smooths.

%, in absolute value. Therefore, we can say that the compression gives a solid result.

Considering the fidelity of the reconstructed signals, the compression ratios (CR^1) reached are quite good (35, 36 and 40%), considering the absence of straight portions. However, as shown with $y_2(t)$ (cf. Fig. 13), the reconstruction method can be improved. The spline reconstruction technique was chosen because it generates a continuous and differentiable function, avoiding the high frequency noise that would appear in the case of linear interpolation for example. The search for a reconstruction technique better suited to our compression algorithm is part of our future prospects.

5.2. Electrocardiogram (ECG) data from PhysioNet

ECG samples retrieved from the PhysioNet [60] database have been compressed. When faced with such ECG signals, the most important is to recognize the main medical features on it to make a diagnosis: the P and T waves, and the QRS complex (cf. Fig. 15). These will be the first to be compromised or to disappear if the compression is too strong. We consider two signals from normal human heart beats at rest, testing whether compressed level 1 exhibits the main characteristics of a normal ECG. The figures from Figs. (16)–(19) show the results.

As these figures show, compression ratios of less than 46 % are obtained at level 1 and the conservation of the QRS complex as well as P and T waves in the reconstructed signal is indisputable. However, one drawback to note is the reduction in R-peaks. This is because the resampling rate cannot exceed half of the original one. The consequence is however limited thanks to the reconstruction. The mean relative error in the peak values between levels 0 and 1 lies in the ranges [11,50] % for the example 1 and [5,40] % for the example 2. The peak R values in the reconstructed signal are significantly closer to the original than in the compressed signal (cf. Figs. 16 and 18): the error is halved.

5.3. Data from swarm satellites (ESA)

The Swarm mission was initiated by the European Space Agency (ESA) in November 2013 with the launching of three identical satellites capturing the fluctuations of the Earth's magnetic field. An extract of this data is shown in Fig. 20 with the signal after compression and after reconstruction. Level 1 compression ratio (CR^1) is better than 36 %. The reconstruction shows a solid result.

6. Conclusion and future research prospects

Assigning relevance to the samples of a record is certainly the key point in designing our progressive lossy data compression algorithm.

Considering one-dimensional digital signals, we have developed an algorithm that cuts the signal into segments and replaces them with their linear regressions. The segmentation is carried out with respect to the variance of the deviation from the local regression. The segments are dynamically delimited by comparing the standard deviation to a predefined threshold value. In assigning relevance this way, we therefore assert that the relevance is evenly distributed in the compressed signal, thus optimizing the compression for a given threshold. Obviously, the compression performance depends on the input signal. The more straight portions it contains, the stronger the compression. In setting the threshold value, the user can adjust the compression ratio. This is especially useful with recordings containing long latencies such as sparse type signals, which are typical of monitoring systems. For an appropriate filtering in the case of measurement data, this setting can be linked, for example, to the noise of the measurement chain or its uncertainty. In order to maximize the applicability of the algorithm, we have also introduced a heuristic for adaptive threshold determination, which does not require any input in addition to the signal. This work opens a unique approach, where the sampling rate adaptation is governed so to produce a sampling of uniform relevance, serving as base level of the tree data structure.

Tests using this heuristic are performed on two ideal noise-free signals as well as two signals from the real world extracted from two scientific databases of different fields (medicine and space). These signals are chosen not to be sparse type in order to investigate compression performance in unfavorable conditions. Despite this, we obtain compression ratios of less than 50 % at level 1 while maintaining the relevant characteristics of the signal (less than 46 % for the ECG signals and 36 % for the satellites measurements). By reconstructing uniform samplings of the ideal noise free signals from their compressions, a measure of the compression error is obtained. Comparing the Fourier transforms of the original and the reconstructed signals, we further allow for future comparative analysis with other compression methods taking into account the ratio between the bandwidth and the sampling frequency of the original signal. The compression can be applied to any sampling, uniform or not. It can be thus applied recursively, so to build a tree data structure. This optional output can feed multiscale analysis tools, helping to find models of different characteristic scales. Data tree can open a powerful avenue for data visualization and exploration. In archive management, when it is necessary to free memory space, it allows moreover a progressive memory release, where the less relevant components are removed first, unlike a sudden erasure file by file as is the case today.

Other developments of the algorithm are in perspective, including in particular the extension to multidimensional signals, for an application in the field of video broadcasting for example. An extension to data acquisition systems with auto-adaptative sampling rate is also in prospect since an adaptive sampling frequency would certainly be a major advance in the field of low energy embedded systems.

The authors would like to thank Dr Benvenuti Juan Francisco, who provided advice regarding compression tests performed on ECGs retrieved from the PhysioNet database. He introduced us to the main medical features a doctor would look for in an ECG to make a diagnosis. He examined the compressed signals we obtained, observing whether their main medical implications were retained.

Credit author statement

Pesenti Daniel: Methodology, Software, Validation, Formal analysis, Writing - Original, Visualization; Morin Lucas: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation; Dias André: Methodology, Software, Validation, Writing - Review & Editing, Visualization; Courret Gilles: Conceptualization, Supervision, Project administration, Funding acquisition.

References

[1] Valério M, Ferrara S. Numeracy at the dawn of writing: mesopotamia and beyond, historia mathematica. 2020. https://doi.org/10.1016/j.hm.2020.08.002. . [Accessed 12 April 2021].
[2] Fü ssel S. Gutenberg and the impact of printing. Taylor & Francis; 2020. https://books.google.ch/books?id=2TPNDwAAQBAJ. [Accessed 12 April 2021].
[3] Al-Sai ZA, Abdullah R, Husin MH. Big data impacts and challenges: a review. In: 2019 IEEE Jordan international joint conference on electrical engineering and information technology. JEEIT); 2019. p. 150–5.
[4] Iafrate F. In: ISTE ed, editor. Intelligence artificielle et Big Data: naissance d'une nouvelle intelligence. ISTE ed, editor. Systèmes d'information avancés, Vol. 7. ISTE ed.; 2018. p. 15.
[5] Ferraris A, Mazzoleni A, Devalle A, Couturier J. In: Big data analytics capabilities and knowledge management: impact on firm performance. Emerald publishing limited, vol. 77. Emerald publishing; 2019. p. 1923–36. 8.
[6] Jun Y, Xiaoming L, Shoujun H. Impacts on environmental quality and required environmental regulation adjustments: a perspective of directed technical change driven by big data. J Clean Prod 2020;275:124126. https://doi.org/10.1016/j.jclepro.2020.124126. . [Accessed 12 April 2021].
[7] Pettiford O. How is data-hungry AI affecting the environment?. https://techhq.com/2019/10/how-is-data-hungry-ai-affecting-the-environment/. [Accessed 12 April 2021].
[8] Masanet E, Shehabi A, Lei N, Smith S, Koomey J. Recalibrating global data center energy-use estimates: impact on firm performance. 2020. https://doi.org/10.1126/science.aba3758. . [Accessed 12 April 2021]. February.
[9] Espinosa J, Kaisler S, Armour F, Money W. Big data redux: new issues and challenges moving forward. 2019. https://doi.org/10.24251/HICSS.2019.131.
[10] Bhatia P. Chapiter 2 : introduction to data mining. Cambridge University Press; 2019. p. 17–27. https://doi.org/10.1017/9781108635592.003.
[11] Fayyad U, Piatetsky-Shapiro G, Smyth P. From data mining to knowledge discovery in databases. AI Mag 1996;17(3):37. https://doi.org/10.1609/aimag.v17i3.1230. . [Accessed 12 April 2021].
[12] Azevedo A. Data mining and knowledge discovery in databases. In: Advanced methodologies and technologies in network architecture, mobile computing, and data Analytics, mehdi khosrow-pour, d.b.a. Edition, vol. 1. Portugal: IGI Global, Polytechnic Institute of Porto; 2019. p. 502–14. https://doi.org/10.4018/978-1-5225-7598-6.ch037.
[13] Ganasan JR. Big data mining: managing the costs of data mining. In: 2019 17th international conference on ICT and knowledge engineer- ing. ICT KE); 2019. p. 1–4. https://doi.org/10.1109/ICTKE47035.2019.8966806.
[14] Menaga D, Saravanan S. Chapiter 7 : application of artificial intelligence in the perspective of data mining. In: Binu D, Rajakumar B, editors. Artificial intelligence in data mining. Academic Press; 2021. p. 133–54. https://doi.org/10.1016/B978-0-12-820601-0.00006-9.
[15] Lin AL, Cen WC, Hong JC. Chapter 8 : electronic health record data mining for artificial intelligence healthcare. In: Xing L, Giger ML, Min JK, editors. Artificial intelligence in medicine. Academic Press; 2021. p. 133–50. https://doi.org/10.1016/B978-0-12-821259-2.00008-9.
[16] Song C-W, Jung H, Chung K. Development of a medical big- data mining process using topic modeling. Cluster Comput 2019;22(1):1949–58. https://doi.org/10.1007/s10586-017-0942-0. . [Accessed 12 April 2021].
[17] Nalinipriya G, Geetha M, Cristin R, Maram B. Chapter 8 : biomed- ical data mining for improved clinical diagnosis. In: Binu D, Ra- jakumar B, editors. Artificial intelligence in data mining. Academic Press; 2021. p. 155–76. https://doi.org/10.1016/B978-0-12-820601-0.00012-4.
[18] Zhang Z, Jianping L. Big data mining for climate change. Amsterdam, Netherlands: Elsevier; 2020.
[19] Majeed I, Naaz S. Current state of art of academic data mining and future vision. Indian J Computer Sci Eng 2018;9:49–56. https://doi.org/10.21817/indjcse/2018/v9i2/180902026. . [Accessed 12 April 2021].
[20] Lemay DJ, Baek C, Doleck T. Comparison of learning analytics and educational data mining: a topic modeling approach. Comput Educ: Artif Intell 2021;2:100016. https://doi.org/10.1016/j.caeai.2021.100016. . [Accessed 12 April 2021].
[21] Fan C, Song M, Xiao F, Xue X. Discovering complex knowledge in massive building operational data using graph mining for building energy management. Energy Procedia 2019;158:2481–7. https://doi.org/10.1016/j.egypro.2019.01.378. innovative Solutions for Energy Transitions, . [Accessed 12 April 2021].
[22] Tekin M, Etlioğ lu M, Koyuncuoğ lu Ö, Tekin E. Data mining in digital marketing. In: Durakbasa NM, Gencyilmaz MG, editors. Proceedings of the international symposium for production research 2018. Cham: Springer International Publishing; 2019. p. 44–61.
[23] Mahmood Z. Data mining and knowledge management application to enhance business operations: an exploratory study. In: Arai K, Kapoor S, Bhatia R, editors. Advances in information and communication networks. Cham: Springer International Publishing; 2019. p. 570–83.
[24] Sarode R, Muley A, Bhalchandra P, Singh SK, Joshi M. Discovery of variables affecting performance of athlete students using data mining. In: Behera HS, Nayak J, Naik B, Abraham A, editors. Computational intelligence in data mining. Singapore: Springer Singapore; 2019. p. 449–58.
[25] Sarlis V, Chatziilias V, Tjortjis C, Mandalidis D. A data science approach analysing the impact of injuries on basketball player and team performance. Inf Syst 2021;99:101750. https://doi.org/10.1016/j.is.2021.101750. . [Accessed 12 April 2021].
[26] Kotawadekar R. 9 - satellite data: big data extraction and analysis. In: Binu D, Rajakumar B, editors. Artificial intelligence in data mining. Academic Press; 2021. p. 177–97. https://doi.org/10.1016/B978-0-12-820601-0.00008-2.

[27] Wang M, Qiu L, Wang X. Gdms: a geospatial data mining system for abnormal event detection and visualization. In: 2019 20th IEEE international conference on mobile data management (MDM); 2019. p. 355–6. https://doi.org/10.1109/MDM.2019.00-34.

[28] Jadhav PP. 11 - advanced data mining for defense and security applications. In: Binu D, Rajakumar B, editors. Artificial intelligence in data mining. Academic Press; 2021. p. 223–41. https://doi.org/10.1016/B978-0-12-820601-0.00009-4.

[29] Iwin Thanakumar JS. Department of Computer Science and Engineering, Survey of data mining algorithm's for intelligent computing system. IRO J 2019;1(1):1–10. https://irojournals.com/tcsst/V1/I1/02.pdf. [Accessed 12 April 2021].

[30] Prabhakaran A, Chithra Lekshmi K, Janarthanan G. Chapiter 10 : advancement of data mining methods for improvement of agricultural methods and productivity. In: Binu D, Rajakumar B, editors. Artificial intelligence in data mining. Academic Press; 2021. p. 199–221. https://doi.org/10.1016/B978-0-12-820601-0.00010-0.

[31] Shannon CE. A mathematical theory of communication. Bell Sys Technical J 1948; 27(3):379–423.

[32] Sayood K. Introduction to data compression, the Morgan Kaufmann series in multimedia information and systems. Morgan Kaufmann; 2017.

[33] Khan MA, Pierre JW, Wold JI, Trudnowski DJ, Donnelly MK. Impacts of swinging door lossy compression of synchrophasor data. Int J Electr Power Energy Syst 2020;123:106182. https://doi.org/10.1016/j.ijepes.2020.106182.

[34] Uthayakumar J, Vengattaraman T, Dhavachelvan P. A survey on data compression techniques: from the perspective of data quality, coding schemes, data type and applications. J King Saud Univ Computer Information Sci 2018. https://doi.org/10.1016/j.jksuci.2018.05.006. . [Accessed 12 April 2021].

[35] Purnamasari R, Suksmono AB, Joseph Matheus Edward I, Zakia I. Fast fourier transform sparsity for high quality weather radar reconstruction. In: IGARSS 2019 - 2019 IEEE international geoscience and remote sensing symposium; 2019. p. 7748–51. https://doi.org/10.1109/IGARSS.2019.8899152.

[36] Thyagarajan KS. Fast fourier transform. Cham: Springer International Publishing; 2019. p. 385–426. https://doi.org/10.1007/978-3-319-76029-2_9.

[37] Tan L, Jiang J. Chapter 10 : waveform quantization and compression. In: Tan L, Jiang J, editors. Digital signal processing. third ed. Academic Press; 2019. p. 475–527. https://doi.org/10.1016/B978-0-12-815071-9.00010-5.

[38] Devi NK, Mahendran G, Murugeswari S, Washburn SPS, Devi DA, Saravanan B, Bharathi G, Begam NM. A new lossless compression method using direction adaptive-discrete wavelet transform and modified spiht coding. Mater Today: Proceedings 2021. https://doi.org/10.1016/j.matpr.2021.03.387. . [Accessed 12 April 2021].

[39] Sharma U, Sood M, Puthooran E. Predictor based block adaptive near-lossless coding technique for magnetic resonance image sequence, Procedia Computer Science. In: International conference on computational intelligence and data science, vol. 167; 2020. p. 696–705. https://doi.org/10.1016/j.procs.2020.03.335. . [Accessed 12 April 2021].

[40] Rani NS. Role of data structures in multiple disciplines of computer science-a review. Int J Sci Eng Res July 2013.

[41] Olech Łukasz P, Spytkowski M, Kwaśnicka H, Michalewicz Z. Hierarchical data generator based on tree-structured stick breaking process for benchmarking clustering methods. Inf Sci 2021;554:99–119. https://doi.org/10.1016/j.ins.2020.12.020. . [Accessed 12 April 2021].

[42] Habeck M. Bayesian structural modeling of large biomolecular systems. Biophys J 2019;116(3, Supplement 1):330a. https://doi.org/10.1016/j.bpj.2018.11.1793. . [Accessed 12 April 2021].

[43] Roel-Touris J, Bonvin AM. Coarse-grained (hybrid) integrative modeling of biomolecular interactions. Comput Struct Biotechnol J 2020;18:1182–90. https://doi.org/10.1016/j.csbj.2020.05.002. . [Accessed 12 April 2021].

[44] Singh N, Li W. Recent advances in coarse-grained models for biomolecules and their applications. Int J Mol Sci 2019;20(15). https://doi.org/10.3390/ijms20153774. . [Accessed 12 April 2021].

[45] Grundy JG, Barker RM, Anderson JA, Shedden JM. The relation between brain signal complexity and task difficulty on an executive function task. Neuroimage 2019;198:104–13. https://doi.org/10.1016/j.neuroimage.2019.05.045. . [Accessed 12 April 2021].

[46] Yuan R, Lv Y, Li H, Song G. Robust fault diagnosis of rolling bearings using multivariate intrinsic multiscale entropy analysis and neural network under varying operating conditions. IEEE Access 2019;7:130804–19. https://doi.org/10.1109/ACCESS.2019.2939546. . [Accessed 12 April 2021].

[47] Marques E, Maciel N, Naviner L, Cai H, Yang J. A review of sparse recovery algorithms. IEEE Access; 2018. https://doi.org/10.1109/ACCESS.2018.2886471. 1–1, . [Accessed 12 April 2021].

[48] Stanković L, Sejdić E, Stanković S, Daković M, Orović I. A tutorial on sparse signal reconstruction and its applications in signal processing. Circ Syst Signal Process 2019;38(3):1206–63. https://doi.org/10.1007/s00034-018-0909-2. . [Accessed 12 April 2021].

[49] Keerthana K, Aasha Nandhini S, Radha S. Chapter 8 : cyber physical systems for healthcare applications using compressive sensing. In: Khosravy M, Dey N, Duque CA, editors. Compressive Sensing in Healthcare, Advances in ubiquitous sensing applications for healthcare. Academic Press; 2020. p. 145–64. https://doi.org/10.1016/B978-0-12-821247-9.00013-5.

[50] Meneguitti Dias F, Khosravy M, Wulfert Cabral T, Moreira Monteiro H, Manhaes de Andrade Filho L, de Mello Honório L, Naji R, Duque CA. Chapter 9 : compressive sensing of electrocardiogram. In: Khosravy M, Dey N, Duque CA, editors. Compressive Sensing in Healthcare, Advances in ubiquitous sensing applications for healthcare. Academic Press; 2020. p. 165–84. https://doi.org/10.1016/B978-0-12-821247-9.00014-7.

[51] Kumar S, Deka B, Datta S. Chapter 10 : multichannel ecg reconstruction based on joint compressed sensing for healthcare applications. In: Khosravy M, Dey N, Duque CA, editors. Compressive Sensing in Healthcare, Advances in ubiquitous sensing applications for health- care. Academic Press; 2020. p. 185–200. https://doi.org/10.1016/B978-0-12-821247-9.00015-9.

[52] Wang Z, Huang S, Wang S, Wang Q, Zhao W. Sparse reconstruction based time-frequency representation for time-of-flight extraction of undersampled lamb wave signal. In: 2020 conference on precision electromagnetic measurements. CPEM); 2020. p. 1–2. https://doi.org/10.1109/CPEM49742.2020.9191705.

[53] Wang J, Qiao W, Qu L. Wind turbine bearing fault diagnosis based on sparse representation of condition monitoring signals. IEEE Trans Ind Appl 2019;55(2): 1844–52. https://doi.org/10.1109/TIA.2018.2873576. . [Accessed 12 April 2021].

[54] Shannon CE. Communication in the presence of noise. Proc IRE 1949;37(1):10–21. https://doi.org/10.1109/JRPROC.1949.232969. . [Accessed 12 April 2021].

[55] Romanov E, Ordentlich O. Above the nyquist rate, modulo folding does not hurt. IEEE Signal Process Lett 2019;26(8):1167–71. https://doi.org/10.1109/LSP.2019.2923835. . [Accessed 12 April 2021].

[56] Jiao L, Shang R, Liu F, Zhang W. Chapter 5 - theoretical basis of compressive sensing. In: Jiao L, Shang R, Liu F, Zhang W, editors. Brain and nature-inspired learning computation and recognition. Elsevier; 2020. p. 109–26. https://doi.org/10.1016/B978-0-12-819795-0.00005-0.

[57] Tan L, Jiang J. Chapter 12 : subband and wavelet-based coding. In: Tan L, Jiang J, editors. Digital signal processing. third ed. Academic Press; 2019. p. 591–648. https://doi.org/10.1016/B978-0-12-815071-9.00012-9.

[58] Kaur C, Bisht A, Singh P, Joshi G. Eeg signal denoising using hybrid approach of variational mode decomposition and wavelets for depression. Biomed Signal Process Contr 2021;65:102337. https://doi.org/10.1016/j.bspc.2020.102337. . [Accessed 12 April 2021].

[59] Algabroun H. Dynamic sampling rate algorithm (dsra) implemented in self-adaptive software architecture: a way to reduce the energy consumption of wireless sensors through event-based sampling. Microsystem Technologies; September 2019.

[60] Goldberger AL, Amaral LAN, Glass L, Hausdorff JM, Ivanov PC, Mark RG, Mietus JE, Moody GB, Peng C-K, Stanley HE, PhysioBank, PhysioToolkit, and PhysioNet. Components of a new research resource for complex physiologic signals. Circulation 2000 (June 13);101(23):e215–20. https://doi.org/10.1161/01.CIR.101.23.e215. circulation Electronic Pages, http://circ.ahajournals.org/content/101/23/e215.full. PMID:1085218.

[61] Wikipedia contributors, Data compression ratio — wikipedia, the free encyclopedia. 2021. https://en.wikipedia.org/w/index.php?title= Data_compression_ratio&oldid=1022012705. [Accessed 4 June 2021].

[62] Wikipedia contributors, Signal-to-noise ratio — wikipedia, the free encyclopedia. 2021. https://en.wikipedia.org/w/index.php?title= Signal-to-noise_ratio&oldid=1024797052. [Accessed 4 June 2021].

[63] Němcová A, Śmúsek R, Maršánová L, Smital L, Vítek M. A comparative analysis of methods for evaluation of ecg signal quality after compression. BioMed Res Int 2018 Jul 18. https://doi.org/10.1155/2018/1868519.

[64] Manikandan M, Dandapat S. Wavelet threshold based tdl and tdr algorithms for real-time ecg signal compression. Biomed Signal Process Contr 2008;3(1):44–66. https://doi.org/10.1016/j.bspc.2007.09.003. . [Accessed 9 June 2021].

[65] Longchamp R. Commande numérique de systèmes dyamiques: cours d'automatique. Switzerland: Presses polytechniques et universitaires romandes; 2010. https://books.google.ch/books?id=z2rFTpmfs4UC.

[66] European Space Agency. Swarm, data provided by the European space agency. 2020. https://swarm-diss.eo.esa.int/#.

[67] Chaparro LF, Akan A. Chapter 5 - frequency analysis: the fourier transform. In: Chaparro LF, Akan A, editors. Signals and systems using MATLAB. third ed. Academic Press; 2019. p. 305–62. https://doi.org/10.1016/B978-0-12-814204-2.00015-6.

[68] Derpanis KG. Fourier transform of the Gaussian. http://www.cse.yorku.ca/~kosta/CompVis_Notes/fourier_transform_Gaussian.pdf. [Accessed 12 April 2021].

[69] Atkielski A. Schematic diagram of normal sinus rhythm for a human heart as seen on ecg (with English labels). https://en.wikipedia.org/wiki/Electrocardiography#/media/File:SinusRhythmLabels.svg. [Accessed 12 April 2021].