A Simple and Efficient Near-lossless Compression Algorithm for Multichannel EEG Systems

Giuseppe Campobello
Department of Engineering, University of Messina
Messina, Italy
giuseppe.campobello@unime.it

Giovanni Gugliandolo
Department of MIFT, University of Messina
Messina, Italy
giovanni.gugliandolo@unime.it

Nicola Donato
Department of Engineering, University of Messina
Messina, Italy
nicola.donato@unime.it

***Abstract*—The research work presented in this paper focuses on a novel near-lossless compression algorithm which can be efficiently used for compression of electroencephalograph (EEG) signals. In particular, our proposed scheme aims to achieve a very low complexity solution suitable for wearable EEG monitoring systems. The algorithm is mostly based on a simple and efficient encoding scheme which can be easily implemented even in resources constrained microcontrollers. Despite its simplicity, comparison results provided for several real-world datasets, show that the proposed algorithm achieves compression ratios comparable with more complex state-of-the-art solutions.**

***Index Terms*—Electroencephalograph (EEG), near-lossless compression, biomedical signal processing, RAKE**

I. Introduction

Electroencephalography (EEG) is the recording of the brain electrical activity by means of multiple electrodes commonly placed on top of the scalp. Currently, its applications spans from the investigation of human cognitive processes and neurological disorders [1] to the implementation of brain computer interfaces (BCI) [2].

In most applications high-density EEG systems with 64 or even 256 electrodes are used to record the brain activity and the recording time spans from few days to several weeks [3], so that a large amount of data is generated for each EEG recording. For instance, one gigabyte (GB) of data per day is generated in the case of 64-electrodes EEG where signals are sampled at 100 Hz and represented with 16-bit resolution.

In this context, considering the large volume of data, compression algorithms can be profitably used with the aim of reducing storage resources and power consumptions [4]–[7].

Basically, it is possible to classify compression algorithms as lossless and lossy [8], [9]. In both cases, considering that the main aim of compression algorithms is to reduce the amount of data, their performance are usually measured in terms of the compression ratio (CR), which is here defined as the ratio between the number of bits needed to represent EEG data before and after compression.

In general, lossy compression algorithms allow to achieve much higher compression ratios but lossless compression schemes are usually preferred in biomedical applications to ensure that waveform details are not lost causing errors in medical diagnosis [10], [11]. Nevertheless, lossless compression techniques have limited impact on storage requirements of EEG applications because typical compression ratios that can be achieved with state-of-the-art lossless algorithms are in the order of 2 or 3 [12]–[17]. Even exploiting neural networks and an adaptive error modeling with context-based bias cancelation a maximum compression ratio of 3.23 has been obtained [18].

To solve this problem, lossy and near-lossless compression algorithms have been proposed that can achieve substantial higher compression ratios than lossless compression techniques by introducing a distortion that is tolerable for the specific application. Differently from lossy techniques, where the quality of the reconstructed signal is expressed considering the percent root-mean-square distortion (PRD), in the case of near-lossless algorithms distortion is measured considering both PRD and the maximum absolute error δ incurred on individual samples.

In order to remove spatial redundancy or for simultaneous spatio-temporal decorrelation, several near-lossless compression algorithms rely on wavelet transforms [19]–[21] and/or matrix decomposition [22]–[25]. For instance, in [22] near-lossless EEG compression schemes based on Singular Value Decomposition (SVD) and its generalization, i.e., Parallel Factor decomposition (PARAFAC), have been investigated and compared with wavelet-based compression techniques. In most cases, PARAFAC yields better compression performance but the maximum compression ratio achieved with a PRD lower than 2% was 4.96. In [23] the authors proposed a near-lossless algorithm able to achieve a CR of 4.58 with a PRD in the range between 0.27% and 7.28%, depending on the specific dataset investigated. More recently, in [24] the authors proposed a SVD-based compression scheme able to achieve 80% data compression (i.e., CR=5) with a PRD of 5%.

However, as observed in [26], transform-based and SVD-based algorithms have computational issues due to the fact that the number of operations scales superlinearly with the number of channels and samples to be processed, which translates directly into higher latency and higher power consumptions. Therefore the above algorithms are not suitable for real-time compression on wearable EEG systems.

With this aim, in [26] authors proposed two near-lossless compression algorithms based on an adaptive predictor and able to achieve, for a maximum error $\delta = 10$, a compression ratio between 5.8 and 10, depending on the specific dataset investigated. Slightly better results have been obtained in [27] combining signal processing and information theory tools such as universal coding, universal prediction and multivariate recursive least squares.

Several other lossy and near-lossless compression algorithms exist which are not mentioned here for sake of space; however, most of them are not suitable when only limited computational resources are available, such as, for instance, in EEG monitoring systems that, in order to reduce power consumptions, rely on microcontrollers without floating point units [26], [28].

In this paper, we present a new near-lossless compression algorithm that relies mostly on a simple and efficient encoding scheme. The proposed algorithm is able to achieve a compression ratio and PRD comparable with other state-of-the-art solutions but without using complex transforms. Therefore, it can be easily implemented even in resources constrained microcontrollers as those commonly used in Internet of Things platforms and low-cost medical instruments and thus, differently from other compression schemes, can be performed directly in whatever EEG monitoring system.

II. Proposed Algorithm

The proposed algorithm exploits the RAKE encoding originally proposed in [29] and that we briefly review here for sake of readability.

A. RAKE algorithm

Basically, the RAKE algorithm provides an efficient manner to encode sparse binary strings, i.e. strings where the number of non-zero bits, also referred to as set bits, is smaller than the overall number of bits.

The algorithm can be explained by considering a sliding window of length T that moves forward over the binary sequence to be compressed, S_{in}, and that is able to catch substrings of T bits at a time (originally, authors presented their idea considering the rake commonly used in agriculture instead of a sliding window, thus the name of the algorithm).

For each substring of T bits, an output codeword, C_{out}, is generated accordingly to the following two possible cases:

1) In the substring there is at at least one set bit: in this case a codeword of $L = 1 + \lceil \log_2 T \rceil$ bits is generated where the first bit is set to 1 and the other $\lceil \log_2 T \rceil$ bits are used to encode the position $p \in [0, ..., T-1]$ of the first non-zero bit within the window; then the window moves forward by $p+1$ positions (i.e. immediately after the set bit that has been already encoded);
2) All T bits are zeros: in this case a single zero bit codeword is used and the sliding window is moved forward by T positions.

The above operations are repeated until the end of the sequence S_{in} is reached. Finally, the compressed string S_{out} is obtained by simply concatenating all previous codewords.

In Figure 1 a simple example is reported showing how the sequence $S_{in} = [010000001010000]$ of $n = 15$ bits is compressed by the RAKE algorithm to produce a compressed sequence $S_{out} = RAKE(S_{in}) = [10101101010]$ of 11 bits.

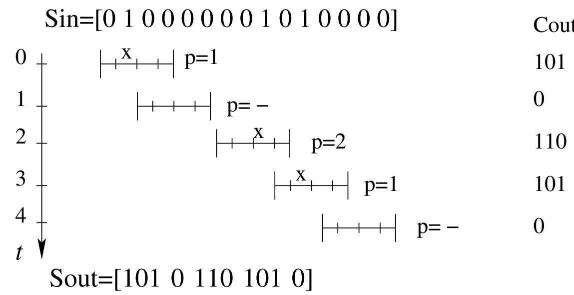

Fig. 1. Example of RAKE compression algorithm ($T = 4$).

As proved in [29], the optimal value of T for a n-bit sequence with k set bits can be obtained by the following equation

$$T^* = (\frac{n}{k} - 1) \cdot \ln(2) \qquad (1)$$

Note that the value of T is necessary to re-obtain the original sequence. Therefore, in order to reduce the related overhead, T is constrained to be a power of two, i.e. $T = 2^b$ where $b = \lceil \log_2(T^*) \rceil$. In this case T can be derived from b which in turns can be encoded with only $\lceil \log_2(\log_2(\frac{n}{k})) \rceil$ bits, i.e., a negligible number of bits in comparison to the length of the compressed string. Finally, it is worth observing that b can be obtained from the maximum codeword length L as $b = L - 1$.

B. Using the RAKE algorithm for compression of EEG signals

Henceforward, we assume that EEG signals are sampled at f_s Hz and that samples are represented with integer numbers of w bits each. For most commercial EEG systems, w is either 12, 16 or 24 bits. We further assume that EEG signals are processed in blocks of $N \times M$ samples, where N is equal to the number of EEG channels and $M = T f_s$ is the number of samples acquired, for each channel, in a time interval T. Note that M can be fixed according to available storage resources.

We indicate the generic block as a matrix $X = [x_{ij}]$, where $i \in \{1, ..., N\}$ and $j \in \{1, ..., M\}$.

In order to apply the RAKE algorithm, we have to transform the set of acquired EEG samples x_{ij} into sparse binary strings. For this purpose we proceed according to the following steps:

1) As first step, starting from X, we obtain the matrix $\tilde{X} = [\tilde{x}_{ij}]$ whose elements are

$$\tilde{x}_{ij} = F \cdot round(\frac{x_{ij}}{F}) \qquad (2)$$

where $round()$ is the integer rounding operator and F is an integer factor fixed according to the desired distortion. We derive the relation between F and PRD in the next subsection.
2) In the second step, we obtain a matrix $D = [d_{ij}]$ whose elements are differences of adjacent samples.
More precisely, we indicate with $diff(A)$ and A' respectively the matrix of column differences and the

transpose of a generic matrix A and define the matrix of differences D as

$$D = \begin{bmatrix} \tilde{x}_{11} & diff(\tilde{X}'_{1:})' \\ diff(\tilde{X}_{:1}) & diff(diff(\tilde{X}')') \end{bmatrix} \quad (3)$$

where $\tilde{X}_{:1}$ and $\tilde{X}_{1:}$ are, respectively, the first column and the first row of $\tilde{X}$.
Note that $\tilde{X}$ can be exactly recovered from D as

$$\tilde{X} = cumsum(cumsum(D)')' \quad (4)$$

where $cumsum()$ represents the cumulative sum of column elements.
Basically, D reduces both temporal and spatial redundacy of EEG signals and thus reduces the number of bits needed to represent $\tilde{X}$.

3) Starting from D, we obtain the matrix $Z = [z_{ij}]$ whose elements are

$$z_{ij} = \begin{cases} 2 \cdot |d_{ij}| & \texttt{if } d_{ij} \geq 0 \\ 2 \cdot |d_{ij}| - 1 & \texttt{if } d_{ij} < 0 \end{cases} \quad (5)$$

It is worth noting that all elements of Z are non-negative numbers and that eq.(5) is invertible, i.e. D can be reobtained from Z. In fact, by representing z_{ij} as a binary word, i.e., $[z_{ij}^{(w_z-1)}, ..., z_{ij}^{(0)}]$ where $z_{ij}^{(l)}$ are either 0 or 1, we have

$$d_{ij} = \begin{cases} z_{ij}/2 & \texttt{if } z_{ij}^{(0)} = 0 \\ -(z_{ij}+1)/2 & \texttt{if } z_{ij}^{(0)} = 1 \end{cases} \quad (6)$$

Henceforward, we indicate with w_Z the maximum number of bits needed to represent the elements of Z and with $Z^{(l)} = [z_{ij}^{(l)}]$ thet l-th bit plane of Z, i.e., the binary matrix obtained considering only the l-th bits of the elements z_{ij}. Note that $l \in [0, w_Z - 1]$.

4) In this step we apply the RAKE algorithm to each of the binary matrices $Z^{(l)}$. More precisely, we consider each matrix $Z^{(l)}$ as a binary string of $N \times M$ bits and apply the RAKE algorithm to obtain the compressed string $Z_c^{(l)} = RAKE(Z^{(l)})$. In practice, we observed that the RAKE algorithm provides significant compression only if the number of set bits k satisfies $k \leq 0.25 \cdot N \cdot M$; therefore, every time that $k > 0.25 \cdot N \cdot M$ holds, we avoid compression by simply considering $Z_c^{(l)} = Z^{(l)}$.

5) Finally, the compressed representation X_c of the original block X is obtained by concatenating all the compressed strings $Z_c^{(l)}$ derived in the previous step, preceded by an header H needed for reconstruction. More precisely, the header H contains the value of w_Z and dimensions L_l of the codewords used by the RAKE algorithm to encode $Z^{(l)}$, so that at the end the compressed block is $X_c = [H, Z_c^{(w_Z-1)}, ..., Z_c^{(0)}]$ with $H = [w_Z, L_{w_Z-1}, ..., L_0]$.
It is worth mentioning that the codeword length L_i is set to 0 when the RAKE algorithm is not applied, i.e., when $Z_c^{(i)}$ coincides with the binary matrix $Z^{(i)}$.

Note that, when $F = 1$ holds, the proposed algorithm is a lossless compression algorithm, i.e., X can be exactly recovered from X_c. In fact, considering that RAKE is a lossless compression algorithm and that matrix transformations introduced in steps 2 and 3 are invertible, it follows that $\tilde{X}$ can be always exactly recovered from X_c; moreover, when $F = 1$ holds, accordingly to (2) we have $\tilde{x}_{ij} = x_{ij}$ and thus $X = \tilde{X}$.

C. On the choice of the rounding factor F

In this subsection we derive the relation between the rounding factor F and the distortion PRD defined by the following equation

$$PRD\% = 100 \cdot \sqrt{\frac{\sum_{i=1}^{N}\sum_{j=1}^{M}(x_{ij} - \tilde{x}_{ij})^2}{\sum_{i=1}^{N}\sum_{j=1}^{M} x_{ij}^2}} \quad (7)$$

where x_{ij} are original EEG samples and $\tilde{x}_{ij}$ are given by eq.(2).

Let us observe that the PRD can be rewritten as

$$PRD\% = 100 \cdot \frac{\sqrt{\sum_{i=1}^{N}\sum_{j=1}^{M}\epsilon_{ij}^2}}{||X||} \quad (8)$$

where $\epsilon_{ij} = x_{ij} - \tilde{x}_{ij}$ and $||X|| = \sum_{i=1}^{N}\sum_{j=1}^{M} x_{ij}^2$ is the Frobenius norm of the matrix X.

According to (2), the error $\epsilon_{ij} = x_{ij} - \tilde{x}_{ij}$ lies in the range $[-\delta, +\delta]$, where $\delta = \lfloor \frac{F}{2} \rfloor$.

We assume that errors ϵ_{ij} are independent and uniformly distributed random variables so that $E(\epsilon_{ij}) = 0$ and $E(\epsilon_{ij}^2) = \frac{F^2}{12}$, where we have indicated with $E()$ the expectation operator. Moreover, under the same hypothesis, we have $E(\sum_{i=1}^{N}\sum_{j=1}^{M}\epsilon_{ij}^2) = NM\frac{F^2}{12}$ and thus $E(PRD\%) = 100 \cdot \sqrt{\frac{NM}{12}} \cdot \frac{F}{||X||}$.

Therefore, given a desired distortion PRD_{thr}, the rounding factor F needed for the first step of the proposed algorithm can be fixed according to the following equation

$$F = \frac{\sqrt{12} \cdot ||X||}{100 \cdot \sqrt{NM}} \cdot PRD_{thr}\% \quad (9)$$

III. Experimental results

We applied the proposed algorithm to different datasets representing real-world EEG signals. All datasets investigated are open access and can be freely downloaded from the official website of the BNCI Horizon 2020 project [31], an european project focused on brain-computer interfaces. Among the 28 available datasets, we considered 4 datasets (DS14,DS15,DS17 and DS19) obtained with 16-bit EEG recording systems and other 2 datasets (DS22 and DS27) with higher resolution, i.e., 24 bit. We reported a short description and the number of EEG channels available for each dataset in Tab. I.

For all datasets investigated, we applied the proposed algorithm considering blocks of $M = 1000$ samples for each channel and evaluated the compression ratio (CR) according to the relation

$$CR = \frac{\texttt{\# bits BEFORE compression}}{\texttt{\# bits AFTER compression}} \quad (10)$$

TABLE I
INFORMATION OF INVESTIGATED DATASETS.

Data Set ID	Short description (and file name)	Number of EEG Channels (N)	Samples per channel	Resolution (w)
DS14	Covert shifts of visual attention (VPiac.mat)	60	657660	16
DS16	Motion VEP Speller (MVEP_VPfat.mat)	63	205132	16
DS17	Center Speller (VPiac.mat)	63	777820	16
DS19	RSVP speller (RSVP_VPfat.mat)	63	492552	16
DS22	Monitoring error-related potentials (Subject01_s1.mat)	64	91648	24
DS27	Reach and Grasp movement decoding (V01.mat)	32	797514	24

TABLE II
A FEW REPRESENTATIVE EEG COMPRESSION ALGORITHMS

Reference	Techniques used	CR	Distortion (PRD%)
[12]	Vector quantization and Huffman encoding	2.63	lossless
[13]	Percepton-based predictor and Arithmetic Coding	2.62	lossless
[14]	Karhunen-Loeve transform	2.84	lossless
[16]	Correlation dimension based predictor	3.20	lossless
[18]	Neural network adaptive error modeling	3.23	lossless
[22]	Matrix and tensor decomposition	4.96/12.13	2.0/5.4
[23]	Independent Component Analysis and wavelet transform (SPIHT)	4.58/9.11	0.27-7.28/0.92-7.66
[24]	Singular Value Decomposition	5	5
[30]	Neural network predictor and Arithmetic Coding	6.5	7.0
[19]	Wavelet transform	9.13	5.25
[27]	Universal prediction and multivariate recursive least squares	6.67/8.56	0.39/0.75

TABLE III
COMPRESSION RATIO (CR), ACTUAL DISTORTION (PRD) AND NORMALIZED MAXIMUM ABSOLUTE ERROR ($NMAE$) ACHIEVED FOR DIFFERENT DATASETS IN THE CASE OF $PRD_{thr} = 2\%$.

Dataset	w	CR	Actual PRD (%)	$NMAE$
DS14	16	7.83	2.00	0.0057
DS16		5.69	2.00	0.0067
DS17		4.61	2.00	0.0047
DS19		4.53	2.00	0.0039
DS22	24	185.59	2.02	0.0009
DS27		149.62	2.01	0.0085

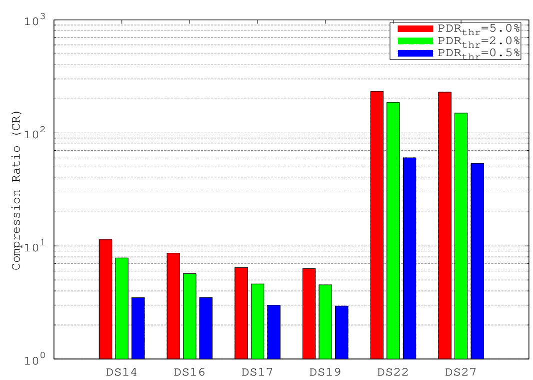

Fig. 2. Compression ratios for different datasets and different values of PRD_{thr}

In Tab. III we reported the compression ratio achieved by the proposed algorithm for $PRD_{thr} = 2\%$. In the same table we reported also the actual PRD, i.e. the PRD achieved after reconstruction and evaluated according to (7), and the normalized maximum absolute error, i.e. $NMAE = \frac{\delta}{2^{w-1}}$.

As it is possible to observe, compression ratios in the order of 150 can be achieved with the proposed algorthm in the case of 24-bit datasets and $PRD_{thr} = 2\%$.

Obviously, the CR decreases for lower values of PRD_{thr} and increases for higher values of PRD_{thr}, as it is possible to observe in Fig. 2 where we reported compression ratios obtained for different values of PRD_{thr}, i.e. $0.5\%, 2\%$ and 5%.

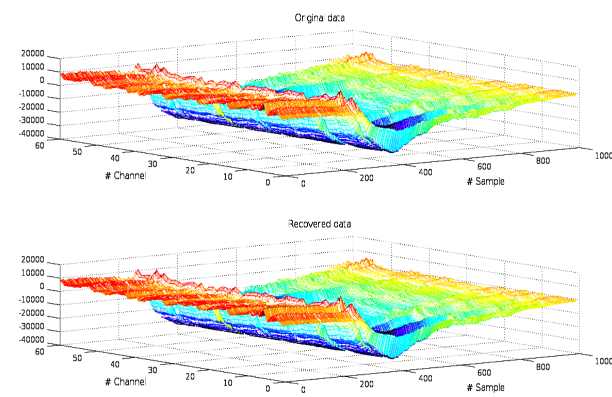

Fig. 3. Example of original and recovered EEG signals ($PRD_{thr} = 2\%$)

In the case of 16-bit datasets and PRD_{thr} between 2% and 5%, a compression ratio between 4.53 and 11.34 is achieved, depending on the specific dataset. For sake of completeness, in Fig. 3 we show a pair of original and recovered EEG signals for one block of DS14 compressed with $PRD_{thr} = 2\%$.

Note that several state-of-the-art compression algorithms achieve similar compression ratios for the same distortion level (see Tab. II). Nevertheless, in the case of the proposed algorithm, neither wavelet transforms nor singular value decomposition are used. Therefore, the proposed algorithm can be implemented even in simple processing units commonly used for wearable EEG monitoring systems.

A few better near-lossless compression algorithms exist in literature able to achieve similar CR but with lower distortion. For instance in [27], a CR of 8.56 with a distorsion lower that 1% has been achieved using multivariate recursive least squares (RLS). However, floating-point RLS is not feasible

TABLE IV
ACTUAL PRD OBTAINED AFTER RECONSTRUCTION

$PDR_{thr}\%$	DS14	DS16	DS17	DS19	DS22	DS27
0.5	0.50	0.50	0.50	0.50	0.51	0.50
2.0	2.00	2.00	2.00	2.00	2.02	2.01
5.0	5.00	5.00	5.00	5.00	5.16	4.97

for wearable EEG systems due to energy consumption and memory constraints. Instead, the proposed solution, mostly based on a simple encoding scheme and arithmetic operations, is suitable even for low-cost microcontroller-based EEG monitoring systems.

Finally, it is also worth noting that, differently from several other lossy and near-lossless compression algorithms, the proposed algorithm allows to fix a priori the desired distortion. In fact, as shown in Tab. IV, the actual PRD evaluated after reconstruction almost coincides with the a-priori fixed threshold PRD_{thr}. Experimental results, omitted for sake of space, confirm that errors are uniformly distributed in $[-\frac{F}{2}, +\frac{F}{2}]$.

IV. CONCLUSIONS

In this paper we have presented a simple and effective near-lossless compression algorithm for multichannel EEG recording systems. Using only simple counting and arithmetic operations, the algorithm is able to achieve performance comparable with other more complex state-of-the-art solutions. Considering its inherent low complexity, the proposed algorithm is well suited for wearable EEG monitoring systems based on resource constrained microcontrollers. As future works, we will better investigate computational complexity of the proposed algorithm and we will evaluate its performance with other kind of biomedical signals.

REFERENCES

[1] D. A. Pizzagalli, *Electroencephalography and high-density electrophysiological source localization (Handbook of Psychophysiology, 3th ed.).* Cambridge University Press, 2007.

[2] R. Abiri, S. Borhani, E. Sellers, Y. Jiang, and X. Zhao, "A comprehensive review of eeg-based brain-computer interface paradigms." *Journal of Neural Engineering*, vol. 16, no. 1, pp. 011 001–011 001, 2018.

[3] C. Lustenberger and R. Huber, "High density electroencephalography in sleep research: potential, problems, future perspective," *Frontiers in neurology*, vol. 3, p. 77, 2012.

[4] A. Nait-Ali and C. Cavaro-Menard, *Compression of Biomedical Images and Signals*. John Wiley & Sons, 2008.

[5] G. Gugliandolo, G. Campobello, P. P. Capra, S. Marino, A. Bramanti, G. D. Lorenzo, and N. Donato, "A movement-tremors recorder for patients of neurodegenerative diseases," *IEEE Trans. Instrum. Meas.*, vol. 68, no. 5, pp. 1451–1457, 2019.

[6] R. Hussein, A. Mohamed, and M. Alghoniemy, "Adaptive compression and optimization for real-time energy-efficient wireless eeg monitoring systems," in *2013 6th Biomedical Engineering International Conference*, 2013, pp. 1–5.

[7] R. Hussein, A. Mohamed, and M. Alghoniemy, "Scalable real-time energy-efficient eeg compression scheme for wireless body area sensor network," *Biomedical Signal Processing and Control*, vol. 19, pp. 122–129, 2015.

[8] D. Salomon and G. Motta, *Handbook of Data Compression (5.ed.).* Springer, 2010.

[9] K. Sayood, *Introduction to data compression (4.ed.).* The Morgan Kaufmann series in multimedia information and systems, 2012.

[10] C. P. Antonopoulos and N. S. Voros, "Resource efficient data compression algorithms for demanding, wsn based biomedical applications," *Journal of biomedical informatics*, vol. 59, pp. 1–14, 2016.

[11] G. Campobello, O. Giordano, A. Segreto, and S. Serrano, "Comparison of local lossless compression algorithms for wireless sensor networks," *J. Network and Computer Appl.*, vol. 47, no. C, pp. 23–31, Jan. 2015.

[12] G. Antoniol and P. Tonella, "Eeg data compression techniques," *IEEE Trans. on Biomedical Engineering*, vol. 44, no. 2, pp. 105–114, 1997.

[13] N. Sriraam and C. Eswaran, "Lossless compression algorithms for eeg signals: a quantitative evaluation," in *Proceedings of the IEEE/EMBS 5th International Workshop on Biosignal Interpretation*, 2005, pp. 125–130.

[14] Y. Wongsawat, S. Oraintara, T. Tanaka, and K. R. Rao, "Lossless multichannel eeg compression," in *2006 IEEE International Symposium on Circuits and Systems*, 2006, pp. 4 pp.–1614.

[15] K. Srinivasan, J. Dauwels, and M. R. Reddy, "A two-dimensional approach for lossless eeg compression," *Biomedical Signal Processing and Control*, vol. 6, no. 4, pp. 387 – 394, 2011.

[16] N. Sriraam, "Correlation dimension based lossless compression of eeg signals," *Biomedical Signal Processing and Control*, vol. 7, no. 4, pp. 379 – 388, 2012.

[17] G. Campobello, A. Segreto, S. Zanafi, and S. Serrano, "An Efficient Lossless Compression Algorithm for Electrocardiogram Signals," in *26th European Signal Processing Conference (EUSIPCO)*, Sep. 2018, pp. 777–781.

[18] N. Sriraam and C. Eswaran, "Performance evaluation of neural network and linear predictors for near-lossless compression of eeg signals," *IEEE Trans. on Information Technology in Biomedicine*, vol. 12, no. 1, pp. 87–93, 2008.

[19] J. L. Cárdenas-Barrera, J. V. Lorenzo-Ginori, and E. Rodríguez-Valdivia, "A wavelet-packets based algorithm for eeg signal compression," *Medical informatics and the Internet in medicine*, vol. 29, no. 1, pp. 15–27, 2004.

[20] K. Srinivasan and M. R. Reddy, "Wavelet-domain l ∞ -constrained two-stage near-lossless EEG coder," in *Communications in Computer and Information Science*. Springer Berlin Heidelberg, 2010, pp. 76–80.

[21] K. Srinivasan, J. Dauwels, and M. R. Reddy, "Multichannel eeg compression: Wavelet-based image and volumetric coding approach," *IEEE Journal of Biomedical and Health Informatics*, vol. 17, no. 1, pp. 113–120, 2013.

[22] J. Dauwels, K. Srinivasan, M. R. Reddy, and A. Cichocki, "Near-lossless multichannel eeg compression based on matrix and tensor decompositions," *IEEE Journal of Biomedical and Health Informatics*, vol. 17, no. 3, pp. 708–714, 2013.

[23] L. Lin, Y. Meng, J. Chen, and Z. Li, "Multichannel eeg compression based on ica and spiht," *Biomedical Signal Processing and Control*, vol. 20, pp. 45 – 51, 2015.

[24] M. K. Alam, A. A. Aziz, S. A. Latif, and A. Awang, "Eeg data compression using truncated singular value decomposition for remote driver status monitoring," in *2019 IEEE Student Conference on Research and Development (SCOReD)*, 2019, pp. 323–327.

[25] G. Campobello, A. Quercia, G. Gugliandolo, et al., "An efficient near-lossless compression algorithm for multichannel eeg signals," in *2021 16th IEEE Int. Symp. on Medical Measurements and Applications (MeMeA)*, 2021.

[26] G. Dufort y lvarez, F. Favaro, F. Lecumberry, . Martn, J. P. Oliver, J. Oreggioni, I. Ramrez, G. Seroussi, and L. Steinfeld, "Wireless eeg system achieving high throughput and reduced energy consumption through lossless and near-lossless compression," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 12, no. 1, pp. 231–241, 2018.

[27] I. Capurro, F. Lecumberry, . Martn, I. Ramrez, E. Rovira, and G. Seroussi, "Efficient sequential compression of multichannel biomedical signals," *IEEE Journal of Biomedical and Health Informatics*, vol. 21, no. 4, pp. 904–916, 2017.

[28] A. M. Abdulghani, A. J. Casson, and E. Rodriguez-Villegas, "Quantifying the performance of compressive sensing on scalp eeg signals," in *2010 3rd International Symposium on Applied Sciences in Biomedical and Communication Technologies (ISABEL 2010)*, 2010, pp. 1–5.

[29] G. Campobello, A. Segreto, S. Zanafi, and S. Serrano, "Rake: A simple and efficient lossless compression algorithm for the internet of things," in *2017 25th European Signal Processing Conference (EUSIPCO)*, Aug 2017, pp. 2581–2585.

[30] N. Sriraam, "Neural network based near-lossless compression of eeg signals with non uniform quantization," in *2007 29th Int. Conf. of the IEEE Engineering in Medicine and Biology Society*, 2007, pp. 3236–3240.

[31] Bnci horizon 2020. [Online]. Available: http://bnci-horizon-2020.eu/database/data-sets