Energy-Efficient Tree-Based EEG Artifact Detection

Thorir Mar Ingolfsson*, Andrea Cossettini*, Simone Benatti[†‡], Luca Benini[*†]

*Integrated Systems Laboratory, ETH Zürich, Zürich, Switzerland
[†]DEI, University of Bologna, Bologna, Italy
[‡]DIEF, University of Modena and Reggio Emilia, Reggio Emilia, Italy

Abstract—In the context of epilepsy monitoring, EEG artifacts are often mistaken for seizures due to their morphological similarity in both amplitude and frequency, making seizure detection systems susceptible to higher false alarm rates. In this work we present the implementation of an artifact detection algorithm based on a minimal number of EEG channels on a parallel ultra-low-power (PULP) embedded platform. The analyses are based on the TUH EEG Artifact Corpus dataset and focus on the temporal electrodes. First, we extract optimal feature models in the frequency domain using an automated machine learning framework, achieving a 93.95% accuracy, with a 0.838 F1 score for a 4 temporal EEG channel setup. The achieved accuracy levels surpass state-of-the-art by nearly 20%. Then, these algorithms are parallelized and optimized for a PULP platform, achieving a 5.21× improvement of energy-efficient compared to state-of-the-art low-power implementations of artifact detection frameworks. Combining this model with a low-power seizure detection algorithm would allow for 300h of continuous monitoring on a 300 mAh battery in a wearable form factor and power budget. These results pave the way for implementing affordable, wearable, long-term epilepsy monitoring solutions with low false-positive rates and high sensitivity, meeting both patients' and caregivers' requirements.

Clinical relevance— The proposed EEG artifact detection framework can be employed on wearable EEG recording devices, in combination with EEG-based epilepsy detection algorithms, for improved robustness in epileptic seizure detection scenarios.

Index Terms—healthcare, time series classification, smart edge computing, machine learning, deep learning

I. INTRODUCTION

Electroencephalography (EEG) is the golden standard medical diagnostic tool to analyze brain activity, with applications that include developing brain-machine interfaces or monitoring epilepsy [1]. Conventional EEG systems are, however, bulky and uncomfortable, causing patients to commonly refer perceived stigmatization. Furthermore, the long wires used to connect multiple electrodes are a significant cause of motion artifacts on the EEG traces [2]. Thus, patients and caregivers firmly push for wearable EEG solutions based on a minimal number of electrodes, allowing for continuous monitoring during the day in an inconspicuous way [3].

Several groups are involved in the development of such minimal EEG systems that can take the form of glasses, behind-the-ear, or even in-ear solutions [4], [5], [6], [7], [8], [9], [10], [11]. However, the collection of EEG signals is not trivial, and signals can become contaminated with artifacts even in the absence of long cables. In fact, artifacts can be caused by external sources (e.g., electrode movement, external noise) or by physiological factors (e.g., chewing, muscle movements, electromyographic or electrooculogram signals coupled to EEG) that are not of cerebral origin. In the context of epilepsy monitoring, these EEG artifacts are often mistaken for seizures due to their morphological similarity in both amplitude and frequency [12], hampering the epilepsy detection algorithms. It is therefore of paramount importance that EEG artifacts are properly detected so as not to induce false alarms in epilepsy detection systems. After detecting an artifact successfully, the epilepsy detection system could then intelligently decide whether to adaptively correct the EEG signal (using methods such as Artifact Subspace Reconstruction [13]), or to extract features that can help the system classify upcoming seizures or disregard the data. To this end, there is a need for estimations of the generic presence of artifacts (binary classification, BC), if not a per-channel estimation of the presence of artifacts (multi-label classification, MC), or even better, an estimation of the specific source cause of artifacts on each channel (multi-class multi-output classification, MMC).

Many different artifact detection/rejection algorithms have been proposed [14], [15], [16], [17], relying on both supervised and unsupervised methods. In [16], an energy-efficient convolutional neural network (CNN) is used to process raw EEG signals without the need for feature extraction, achieving around 74% accuracy on an in-house EEG artifact dataset. In [18], which focuses on the Temple University Hospital (TUH) EEG Artifact Corpus (TUAR) [12] dataset, a 71.43% accuracy score and a 0.8021 weighted F1 score was obtained, utilizing a *LDA* model and frequency features. In another work [19], also focused on the TUAR dataset, a deep learning framework was used, achieving a 0.838 F1 score. Finally, also in [20] the focus was on the TUAR dataset, and deep learning models allowed to classify artifacts with a 75% accuracy.

However, such approaches present significant limitations in the context of the deployment on minimal, wearable, low-power epilepsy monitoring devices. In fact, most of these methods and algorithms have not been designed with wearable long-term systems in mind. For example, the energy-efficient CNN introduced in [16] has an energy consumption of around 31 mJ per inference, which is not suitable for long-term usage. The Low-Power LSTM Processor introduced in [17] has an energy consumption of around 21 µJ per inference, which is a step in the right direction. Further optimizations are, however, needed, also to increase the accuracy of the detection.

Corresponding email: {thoriri}@iis.ee.ethz.ch

Within this context, we present a framework to automatically detect EEG artifacts, designed to aid and support more robust epilepsy detection systems [21] for wearable unobtrusive epilepsy detection with a reduced montage. Specifically, the main contributions are the following:

- Optimal model selection using a publicly available automated machine learning framework;
- A comprehensive evaluation of models on 15 datasets extracted from the TUH EEG Artifact Corpus;
- Achieved state-of-the-art $\approx 94\%$ artifact detection accuracy considering a 4 temporal channel EEG setup;
- Implementation and performance optimization of the above framework on real PULP chip target, namely Mr. Wolf [22], designed in 40nm FDSOI technology and widely used in ultra-low-power wearable nodes for biomedical applications [23], achieving state-of-the-art ≈ 4 μJ per inference.

The experimental results demonstrate that a high-accuracy framework for continuous automatic artifact detection can be designed for wearable long-lifetime systems with a low electrodes count.

II. BACKGROUND

A. Artifact Dataset

The TUH EEG Corpus is the world's most extensive open-source corpus of EEG data [12]. A subset of the corpus is the Temple University Artifact Corpus (TUAR), containing 310 annotated EEG files from 213 patients with annotations of every artifact. The dataset includes labels of five different individual artifacts, seven combinations of two different artifacts, and background EEG (no artifact), with a total of 13 different labels. Five different sampling frequencies are used in the dataset, with around 71.51% of the data sampled at 250 Hz.

B. Related Work

Not much prior work has been done on the TUAR dataset concerning artifact detection. One of the first works [18] was set up as a benchmark for further explorations on the dataset: it dates back to 2019 and used an earlier release of the TUAR dataset, where not every artifact had been fully annotated. Furthermore, while all the data have been considered, a clear indication of how the different sampling frequencies are processed was missing. Fast Fourier Transform (FFT) was utilized for feature pre-processing, and the algorithms considered are the following: *AdaBoost, Gaussian Naive Bayes, k-nearest neighbors, Linear discriminant analysis (LDA), multilayer perceptron, Random Forest, stochastic gradient descent Classifier, XGBoost*. The primary outcome of the work was that the best performing algorithm was the *LDA* with a 71.43% accuracy score and a 0.8021 weighted F1 score.

In [24], a 1D-CNN architecture takes in the raw EEG data with no additional feature pre-processing. The primary purpose of the work was not to classify artifacts with as much accuracy as possible but instead to design a framework that works when electrodes are missing or shifted. As such, the baseline
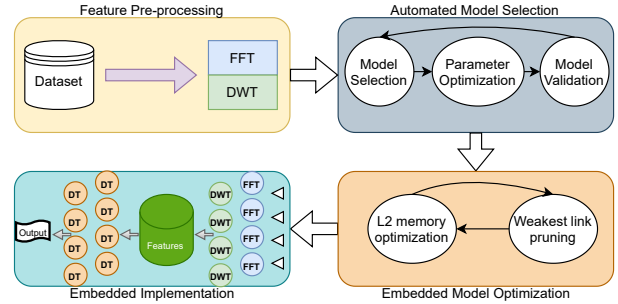

Fig. 1. Schematic representation of the workflow. Data is pre-processed using DWT and FFT, and model selection is made with TPOT. Embedded optimization and pruning are then done before classification is performed with an ensemble of decision trees.

accuracy 71.1% does not improve upon the results offered by *LDA*.

A deep learning framework for an early exit upon artifact detection was proposed by [19]. Since the dataset has multiple sampling frequencies, the authors first re-sample the data that do not adhere to the 250 Hz sampling frequency. Then, they band-pass filter the data between 0.3 and 40 Hz and apply a notch filter at 60Hz. Windows of 10 seconds length are considered, and each channel is treated independently. The dataset was then split into 80-10-10 for train-validation-test, patient-independent between the datasets. The reported results are 0.838 F1 score, 0.853 precision, and 0.829 recall.

The most extensive research on this dataset for artifact detection is done in [20]. The same procedure as of [19] is followed, with the re-sampling of the data to 250 Hz. Then, three different deep learning methods and an ensemble of their outputs are tested. The best accuracy score for the binary case was obtained with a CNN network achieving 75% accuracy, and the best result for the ensemble resulted in a 67.59% accuracy.

III. MATERIALS & METHODS

In this section, we present the main steps followed in this work, starting from the extraction and re-sampling of the dataset, followed by the pre-processing and model selection, and concluding with the implementation on the PULP embedded platform. Fig. 1 shows a sketch of the workflow.

A. Dataset extraction

The TUAR has 22 channels separately annotated with 13 different labels (12 different artifacts labels, 1 non-artifact label). Starting from this dataset, we consider three different types of classification approaches: BC, MC, MMC. In the following, we indicate as ΔT a generic time window, and as $\Theta(\Delta T)$ the label assigned to that time window.

*Binary Classification (**BC**).* For a time window ΔT, if there is an artifact on any one of the channels, we label that window as an artifact ($\Theta(\Delta T) = 1$). Differently, if no channel was initially labeled with an artifact in that window, the window gets labeled as standard background EEG ($\Theta(\Delta T) = 0$).

*Multilabel Classification (**MC**).* In this scenario, each channel is analyzed independently. Therefore, we have an independent binary classification on each channel. If the i-th channel was initially labeled with any artifact label, we assign a label to that window for that channel as $\Theta_i(\Delta T) = 1$. Similarly, if channel i does not have an artifact in the considered time window, it is labeled as normal background EEG: $\Theta_i(\Delta T) = 0$.

*Multiclass-Multioutput Classification (**MMC**).* In this scenario, we expand the MC, discriminating between the specific cause of the artifact (12 alternative possibilities) for each label of each channel. Therefore, if the i-th channel was originally labeled with the artifact type k (with $k \in [1, 12]$), we assign a label to that window for that channel as $\Theta_i(\Delta T) = k$. Similarly, if channel i does not have an artifact in the considered time window, it is labeled as normal background EEG: $\Theta_i(\Delta T) = 0$.

The TUAR dataset also has 5 different sampling frequencies (250 Hz, 256 Hz, 400 Hz, 512 Hz, and 1000 Hz). Since the features extraction requires having a model with a constant sampling frequency, we also extract 5 different datasets depending on the sampling frequencies.

- 250 Hz (**A**)
- 250 + 1000 Hz (**B**)
- 256Hz (**C**)
- 256 + 512 Hz (**D**)
- All frequencies (**E**)

Dataset (**A**) contains only the data of the original dataset that was sampled at 250 Hz. Datasets **B** and **C** are generated by decimating the higher frequency to match the lower one. Dataset **E** is generated by re-sampling the data to match a 250 Hz sampling frequency, using a one-dimensional linear interpolation.

To summarize, we extract 15 different datasets from the TUAR data repository, considering 5 different frequency configurations and 3 different labeling approaches.

B. Pre-processing and feature extraction

We use Discrete Wavelet Transform (DWT) and FFT for feature extraction on each dataset. We consider a 1 second time window as longer windows did not result in better classification accuracy of artifacts; furthermore, longer windows introduce a longer latency of artifact detection.

FFT is a fast way of calculating the frequency representation of time-domain signal values. However, the time information of the signal is lost after the transformation. FFT features have been shown as a good feature for artifact detection [25], and we use the FFT to calculate the energy of the high-frequency parts of the signal (frequencies above 80 Hz).

DWT is a signal decomposition technique for time-frequency analyses [26] that has been reported to work well for seizure detection [21]. DWT offers a good trade-off between signal-to-noise ratio and performance [27], and it allows to obtain both frequency and temporal features of a signal by applying a cascade series *Mother Wavelets* filters.

We use FFT and DWT with a 1 s window length. The multi-core platform (Sec. III-E) allows the process of computations

in a fast, energy-efficient, and accurate manner, splitting the computation of the FFT and DWT using 4 cores for each. Selection of variables and decomposition is performed following the same approach as in [28]. For the DWT, we use the Haar wavelet, and once the detail coefficients for each level are computed, we calculate the energy to be used as a feature. The Haar wavelet was chosen through empirical exploration. Even though the FFT and DWT features can be used alone and give good metrics, relying on a combination of FFT and DWT features yielded the best results.
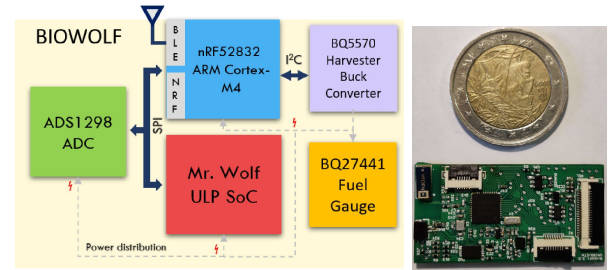

Fig. 2. Left: main building blocks of the BioWolf [23] embedded platform. Right: wearable form factor of BioWolf, compared to a 2 euro coin.

C. Model Selection

Following the feature extraction based on FFT and DWT, for model selection and optimization, we utilize the Tree-based Pipeline Optimization Tool (TPOT) [29], which is an automated machine learning system that takes in features and labels and uses genetic programming to output the best model with cross-validated classification accuracy. Using TPOT allows for a comprehensive search over a wide field of different machine learning models. Among the different available AutoML frameworks [30], [31], [32], we relied on TPOT because of its simplicity and short development time. The reported methodology can be easily adapted and extended to the other frameworks.

D. Performance Metrics

We evaluate the models according to the classification accuracy, the ratio between correctly classified trials, and the total number of trials in the test set. Since the TUAR dataset we use in this paper is imbalanced, we also consider the F1 score, which is defined as:

$$F_1 = 2 \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{1}$$

For the **MMC** case, we report weighted F1 scores considering each class's support.

E. Embedded platform

Following the approach of [21], we implement the artifact detection framework on the BioWolf wearable ExG device [23] (Fig. 2). BioWolf is a sub-10-mW 8-channels brain-computer interface platform equipped with the PULP Mr. Wolf multi-core processor and offering wireless Bluetooth low-energy (BLE) connectivity. The BioWolf system is based on a multi-core fully programmable processor, combining the versatility

TABLE I

OPTIMIZED ALGORITHM OUTPUT FROM TPOT

Sampling rate	250 Hz			250+1000 Hz			256 Hz			256+512 Hz			All		
Dataset	BC	MC	MMC	BC	MC	MMC	BC	MC	MMC	BC	MC	MMC	BC	MC	MMC
Accuracy [%]	93.95	90.05	89.23	93.43	90.14	89.25	87.34	86.75	86.92	86.88	86.52	85.69	92.23	90.72	87.94
F1 score	0.838	0.600	0.867	0.825	0.608	0.868	0.644	0.607	0.830	0.671	0.647	0.826	0.798	0.749	0.853

of a programmable system with the efficiency of a computational 8-core cluster with two shared floating-point units. Mr. Wolf outperforms the performance achievable with commercial single-core low-power processors with a comparable power budget by at least one order of magnitude. Mr. Wolf has an L2 memory of 512 kB and a shared L1 memory of 64 kB in terms of memory footprint.

F. Embedded implementation

1) Feature Extraction Implementation: The FFT implementation on Mr. Wolf exploits the conjugate symmetry of real-valued FFT to optimize the computation. A complex FFT is computed over half of the signal, and the real-valued part is then extracted from the output. Therefore, the computation is significantly reduced compared to doing a full real-valued FFT on all the signal. Furthermore, the FFT also uses a mixed-radix complex FFT approach to save processor computation time. As for the DWT implementation, it iteratively passes through low-pass and high-pass filters.

2) Decision Tree Implementation: The optimized temporal models generated by TPOT are mainly Extra Trees, i.e., an ensemble of Decision Trees [33] (DT). The obtained DTs are pretty large, with millions of threshold values, not fitting on an embedded platform. Therefore, we applied pruning on the DTs with a Minimal cost-complexity pruning (MCCP) algorithm [34]. While MCCP is most often used to prune DTs to avoid overfitting, in this case, we use it to prune the DTs down such that the tree will fit on the L2 memory of the Mr. Wolf multicore processor. We, therefore, take the best-performing model that TPOT outputs and prune it such that the total number of nodes in the tree fits on the 512 kB L2 memory.

During inference, we start from the tree's root node and look in the current feature array to see which feature we must compare with the current node threshold value. Depending on the outcome of the comparison between the threshold value and the feature value, we either go into the tree's left or right branches. The behavior of a decision tree can then be implemented using four arrays:

- Current feature array
- Threshold value array
- Left child node index array
- Right child node index array

Usually, a decision tree implementation needs another array: a value array that corresponds to the final classification output of the tree. Our four array approach is, therefore, an optimized representation of the decision tree since we inject the corresponding class value prediction into the right child node at the current index for leaf nodes. By doing so, we compress

the required memory footprint to implement a decision tree by $10-30\%$ (depending on how the value array is implemented).

We represent the threshold values with 32 bits. Since the feature array is relatively small (5 features per channel), we can represent it with 8 bits, while the index of the left and right child nodes can be represented with 16 bits. In total, each node in the decision tree requires 9 bytes. These arrays are then stored on the L2 memory of Mr. Wolf.

We do not use double-buffered direct memory transfer to transfer tree contents from L2 to L1. This decision is based on the sparse inference property most decision trees have. The maximum number of nodes in a decision tree depends on the maximum depth of the tree, with the number of nodes (binary decision trees) growing exponentially with regards to the depth ($2^{d+1}-1$). However, only a tiny percentage of these nodes are traversed during inference. As an example, we can consider a decision tree of depth 20. Such tree has a theoretical maximum number of nodes of $2\,097\,151$. In practice, such a decision tree with a maximum depth of 20 would have far fewer nodes, as some branches are terminated early. During inference of this example DT, we would at most traverse through 20 nodes, which is only around 0.00095% of the total nodes. Consequently, doing double-buffered direct memory access would require the microcontroller to move $2\,097\,151$ nodes (around 18.9 MB) from the L2, while the inference only requires 20 nodes (around 180B). This explains why blind direct memory access (DMA) of all nodes to L1 is an unnecessary and energy-wasting solution.

G. Power Measurements

We operate the cluster of Mr. Wolf at 100 MHz (frequency of maximal energy efficiency [22]). A Keysight N6705C power analyzer, with 0.2048 ms sampling interval, is used to perform the power measurements.

IV. EXPERIMENTAL RESULTS

A. Classification results

Table I summarizes the accuracy metrics reported for all the datasets and their sampling frequencies considered in this paper, applied to the TUAR data. We consider the TCP average referenced montage [35], focusing on the 4 temporal channels (F7-T3, T3-T5, F8-T4, T4-T6, according to the 10-20 international system notation). Comparing the best models which TPOT found for the **MMC** and all frequencies case, we see that the model outperforms all other models proposed in the literature (which were limited to maximum 75% accuracy and maximum 0.838 F1 score). These results demonstrate the feasibility of accurate artifact classification based on minimal-montage EEG setups.
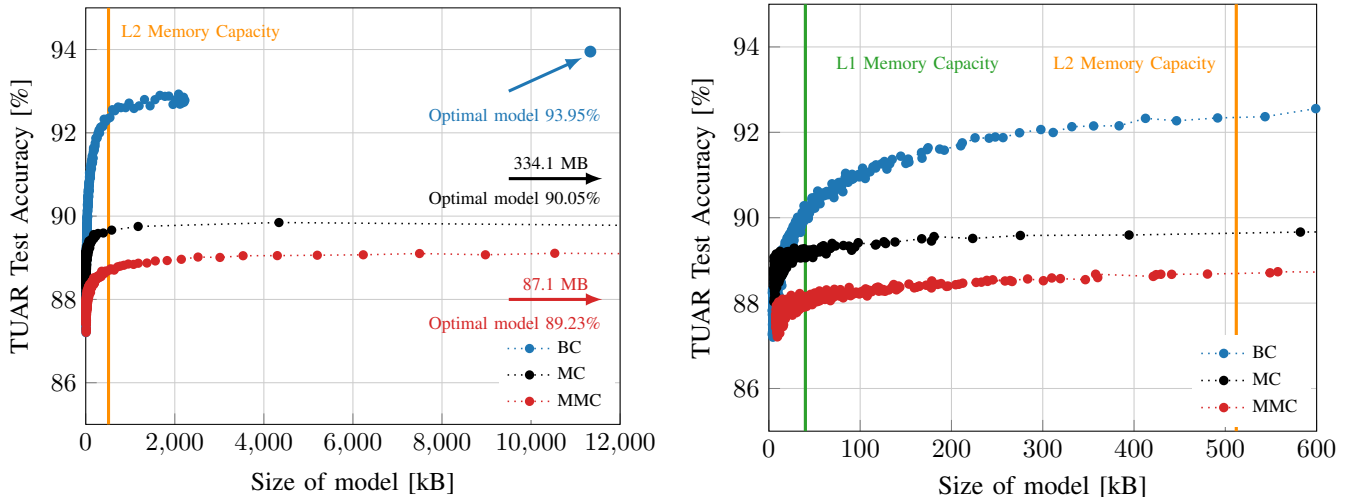
Fig. 3. **Left**: Accuracy vs size of Extra Trees model when implemented on the Mr. Wolf microprocessor for all three labelling methods considered. Regions where L1 and L2 are filled are marked with green and orange lines respectively. **Right**: zoom of the left plot over the 0-600 kB range.

B. Embedded Implementation

After inspecting the optimal DT ensemble that the automated TPOT framework outputs, we optimized it further to fit Mr. Wolf. First, we limit the number of decision trees in the ensemble such that they are a multiple of 8, to parallelize it on the 8-cores cluster. Then, we use the MCCP algorithm to prune away the weakest links in the decision trees. Figure 3 shows the result of such optimization. We perform these limiting and pruning operations for all three cases (BC, MC, MMC) of the 250 Hz dataset. In the **BC** case, by limiting the number of decision trees to be a multiple of 8, we can decrease the size of the model very drastically, from 11000 kB to around 2300 kB, at the cost of only 1% of accuracy reduction. Further optimization is done using the MCCP algorithm, slowly increasing the complexity parameter that controls how much pruning is done. We can then see that we reach a 92.4% accuracy when the model's size matches the size of the L2 memory (512 kB), which corresponds to a decrease of only 1.55% from the optimal model found by using TPOT. For the **MC** and **MMC** cases, we see that we can prune the decision trees more aggressively, with almost no loss in accuracy, until fitting the size of the L1 memory. This low loss of accuracy with pruning validates that embedded implementations of the models with low memory usage can offer similar accuracy levels as state-of-the-art models.

Table II reports the energy numbers of the power measurements performed on Mr. Wolf. Implementing an artifact detecting Extra Trees model on a multi-core edge platform (Mr. Wolf) requires a power envelope of only $\approx 22\,\mathrm{mW}$ with sub-$200\,\mu s$ processing time, thus with much lower energy re-

quirements than the analog front-end and BLE of the complete platform that enables multi-day functionality [23]. In combination with previous results for low energy implementation of seizure detection on Mr. Wolf [21], these pave the way to the design of robust and unobtrusive EEG wearable devices.

V. CONCLUSION

This work presented the analysis and implementation of an artifact detection framework with minimal EEG setups (4 temporal channels), considering different classification approaches (binary, multi-label, multi-class multi-output). We used a combination of FFT and DWT for signal pre-processing and an automated machine learning framework (TPOT) to search for the optimal model for each scenario. Analyses have been done on the TUAR dataset with 15 sub-datasets extracted, and we achieved state-of-the-art accuracy and F1 scores. Furthermore, we deployed and optimized the algorithms on a PULP platform, achieving minimal energy requirements ($\approx 4\,\mu J$ per inference), outperforming competing commercial devices. These results, combined with robust epilepsy detection models [21], show that a PULP system is indeed one of the best candidates for future wearable epilepsy monitoring systems based on minimal EEG setups. Future work will focus on integrating the proposed artifact detection model with a robust epilepsy detection framework and on expanding the proposed techniques to include sensor fusion from other data sources, to be realized in a wearable setting of a body area network and on testing the whole system in ambulatory and domestic environments.

TABLE II
ENERGY NUMBERS FOR IMPLEMENTATION ON MR WOLF.

Dataset	BC	MC	MMC
Time/inference [ms]	0.18	0.19	0.21
Power [mW]	22.41	22.43	22.44
Energy/inference [μJ]	4.03	4.26	4.71

REFERENCES

[1] W. H. Organization *et al.*, "Epilepsy: a public health imperative," 2019.

[2] W. O. Tatum *et al.*, "Artifact and recording concepts in eeg," *Journal of clinical neurophysiology*, vol. 28, no. 3, pp. 252–263, 2011.

[3] E. Bruno *et al.*, "Seizure detection at home: Do devices on the market match the needs of people living with epilepsy and their caregivers?" *Epilepsia*, vol. 61, pp. S11–S24, 2020.

[4] R. Asif *et al.*, "Epileptic seizure detection with a reduced montage: A way forward for ambulatory EEG devices," *IEEE Access*, vol. 8, pp. 65 880–65 890, 2020.

[5] N. Pham *et al.*, "WAKE: a behind-the-ear wearable system for microsleep detection," in *Proc. Int. MobiSys*, 2020, pp. 404–418.

[6] D. Sopic *et al.*, "e-glass: A wearable system for real-time detection of epileptic seizures," in *IEEE ISCAS*, 2018, pp. 1–5.

[7] M. Guermandi *et al.*, "A wearable device for minimally-invasive behind-the-ear eeg and evoked potentials," in *IEEE BioCAS*, 2018, pp. 1–4.

[8] B. BVBA, "Byteflies · Our Solutions."

[9] R. Kaveh *et al.*, "Wireless user-generic ear EEG," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 14, no. 4, pp. 727–737, 2020.

[10] Y. Gu *et al.*, "Comparison between scalp eeg and behind-the-ear eeg for development of a wearable seizure detection system for patients with focal epilepsy," *Sensors*, vol. 18, no. 1, p. 29, 2018.

[11] B. G. Do Valle *et al.*, "Wireless behind-the-ear eeg recording device with wireless interface to a mobile device (iphone/ipod touch)," in *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, 2014, pp. 5952–5955.

[12] A. Hamid *et al.*, "The temple university artifact corpus: An annotated corpus of eeg artifacts," in *2020 IEEE Signal Processing in Medicine and Biology Symposium (SPMB)*, 2020, pp. 1–4.

[13] C.-Y. Chang *et al.*, "Evaluation of artifact subspace reconstruction for automatic artifact components removal in multi-channel eeg recordings," *IEEE Transactions on Biomedical Engineering*, vol. 67, no. 4, pp. 1114–1121, 2020.

[14] A. Mognon *et al.*, "Adjust: An automatic eeg artifact detector based on the joint use of spatial and temporal features," *Psychophysiology*, vol. 48, no. 2, pp. 229–240, 2011.

[15] V. Lawhern *et al.*, "Detection and classification of subject-generated artifacts in eeg signals using autoregressive models," *Journal of neuroscience methods*, vol. 208, no. 2, pp. 181–189, 2012.

[16] M. Khatwani *et al.*, "Energy efficient convolutional neural networks for eeg artifact detection," in *2018 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, 2018, pp. 1–4.

[17] Hasib-Al-Rashid *et al.*, "A low-power lstm processor for multi-channel brain eeg artifact detection," in *2020 21st International Symposium on Quality Electronic Design (ISQED)*, 2020, pp. 105–110.

[18] S. Roy, "Machine learning for removing eeg artifacts: Setting the benchmark," 2019.

[19] L. Qendro *et al.*, "High frequency eeg artifact detection with uncertainty via early exit paradigm," 2021.

[20] D. K. Kim *et al.*, *Fast Automatic Artifact Annotator for EEG Signals Using Deep Learning*. Cham: Springer International Publishing, 2021, pp. 195–221.

[21] T. M. Ingolfsson *et al.*, "Towards long-term non-invasive monitoring for epilepsy via wearable eeg devices," in *2021 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, 2021, pp. 01–04.

[22] A. Pullini *et al.*, "Mr. wolf: An energy-precision scalable parallel ultra low power soc for iot edge processing," *IEEE J. Solid-State Circuits*, vol. 54, no. 7, pp. 1970–1981, 2019.

[23] V. Kartsch *et al.*, "Biowolf: A sub-10-mw 8-channel advanced brain–computer interface platform with a nine-core processor and ble connectivity," *IEEE Trans. Biomed. Circuits Syst.*, vol. 13, pp. 893–906, 2019.

[24] A. Saeed *et al.*, "Learning from heterogeneous eeg signals with differentiable channel reordering," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 1255–1259.

[25] M. K. Islam *et al.*, "Methods for artifact detection and removal from scalp eeg: A review," *Neurophysiologie Clinique/Clinical Neurophysiology*, vol. 46, no. 4, pp. 287–305, 2016.

[26] S. G. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 11, no. 7, pp. 674–693, 1989.

[27] S. Benatti *et al.*, "Towards EMG control interface for smart garments," in *Proc. ACM ISWC*, 2014, pp. 163–170.

[28] ——, "Scalable eeg seizure detection on an ultra low power multi-core architecture," in *2016 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, 2016, pp. 86–89.

[29] T. T. Le *et al.*, "Scaling tree-based automated machine learning to biomedical big data with a feature set selector," *Bioinformatics*, vol. 36, no. 1, pp. 250–256, 2020.

[30] N. Erickson *et al.*, "Autogluon-tabular: Robust and accurate automl for structured data," *arXiv preprint arXiv:2003.06505*, 2020.

[31] E. LeDell *et al.*, "H2O AutoML: Scalable automatic machine learning," *7th ICML Workshop on Automated Machine Learning (AutoML)*, July 2020.

[32] M. Feurer *et al.*, "Efficient and robust automated machine learning," in *Advances in Neural Information Processing Systems 28 (2015)*, 2015, pp. 2962–2970.

[33] L. Rokach *et al.*, *Decision Trees*. Boston, MA: Springer US, 2005, pp. 165–192.

[34] L. Breiman *et al.*, "Classification and regression trees," 1983.

[35] V. Shah *et al.*, "Optimizing channel selection for seizure detection," in *2017 IEEE Signal Processing in Medicine and Biology Symposium (SPMB)*, 2017, pp. 1–5.