An EEG Artifact Identification Embedded System using ICA and Multi-Instance Learning

Ali Jafari[1], Sunil Gandhi[1], Sri Harsha Konuru[1], W. David Hairston[2], Tim Oates[1], and Tinoosh Mohsenin[1]

[1]Department of Computer Science & Electrical Engineering, University of Maryland, Baltimore County
[2]Human Research and Engineering Directorate, US Army Research Lab

***Abstract*—Electroencephalogram (EEG) data is used for a variety of purposes, including brain-computer interfaces, disease diagnosis, and determining cognitive states. Yet EEG signals are susceptible to noise from many sources, such as muscle and eye movements, and motion of electrodes and cables. Traditional approaches to this problem involve supervised training to identify signal components corresponding to noise so that they can be removed. However these approaches are artifact specific. In this paper, we present a novel software-hardware system that uses a weak supervisory signal to indicate that some noise is occurring, but not what the source of the noise is or how it is manifested in the EEG signal. The EEG data is decomposed into independent components using ICA, and these components form bags that are labeled and classified by a multi-instance learning algorithm that can identify the noise components for removal to reconstruct a clean EEG signal. We also performed extensive hyperparameter optimization for the model with the goal of improving accuracy without increasing execution time. This resulted the execution time to be reduced from 282 s to 8.8 s when running the model on an embedded ARM CPU processor at 1.6 GHz clock frequency. In this paper, we present the overall system which includes ICA, SAX and MIL, along with preliminary results for software and hardware implementation when using real EEG data from 64 electrodes. The proposed system consumes 909 mW power during processing above a baseline of 2.32 W idle, while achieving 91.2% artifact identification accuracy.**

I. Introduction

Electroencephalogram (EEG) data is used in a variety of applications because of its non-invasive nature, high resolution and comparatively low cost. These applications include brain-computer interfaces, seizure detection [1]- [2], and determining cognitive states. Unfortunately, EEG data is highly susceptible to artifacts from many sources like muscle movement, eye movement and environmental artifacts. Consequently, many methods and heuristics exist to identify tainted EEG segments, such as correlation with reference signals (EOG and ECG) [3], characteristic morphology (the steep rise and more shallow fall of electrode pop) [4], spatial localization (eye movement artifacts tend to occur in electrodes near the temporal and frontal muscles) and unusually large amplitude [4]. Unfortunately in order to operate, these methods need to first identify characteristics of all potential artifacts in order to properly classify and/or remove them, which can be tedious considering the large number of possible artifacts.

Another class of algorithms that is effective in separating EEG signal and artifacts is independent component analysis (ICA) [5]. ICA has been used to remove artifacts when the components corresponding to them are manually identified, if there is a reference signal, or if supervisory information is available to train a classifier offline to identify offending components online. Traditional approaches to this problem involve supervised training to identify signal components corresponding to noise so that they can be removed. However these approaches are artifact specific.

In this paper, we propose a software-hardware solution for receiving multi-channel EEG signals and performing artifact identification through the use of weak supervisory information. We formulate artifact identification as a multi-instance learning problem [6] [7] where we provide a bag of components from ICA as input and whether the signal contains artifact or not as a weak supervisory signal. We do not provide any information about artifact type, component correspondence to artifact or any other reference signal to the classifier. This allows us to deploy our solution for different kinds of artifacts and saves the effort of tedious annotations of artifact type, demonstrated here based on eyebrow movements.

II. System Architecture

The goal of this paper is to identify the artifacts in multi-channel EEG data using a weak supervisory signal. This allows our solution to work in non-stationary, incompletely controllable environments as it is significantly easier to generate highly accurate information about the existence of artifacts in multi-channel EEG data than to directly identify and characterize the artifacts themselves. Fig. 1 shows the architecture for our system for artifact identification. Where we use three algorithms: (1) ICA, (2) featurization of components using symbolic aggregate approximations (SAX) [8] and (3) multi instance learning (MIL).

ICA separates multivariate signals into its independent additive components. Previous work has shown that ICA successfully separates EEG data into components such that certain components correspond to artifacts [5] [9]. Our goal is to identify if such components can be identified without providing artifact type. We first use ICA to generate components from multi-channel EEG data. Components are selected corresponding to the window of length 1000 as a "bag" of components, and labeled positive if the corresponding EEG data contains artifact, otherwise it is labelled negative. We featurize each component in a bag using symbolic aggregate approximation (SAX) and use multi instance learning to train a classifier to predict if a component bag contains artifact or not.

A. Featurizing time series

We featurize components generated from ICA using SAX, which takes time series (ICA components) as input and pro-

978-1-4673-6853-7/17/$31.00 ©2017 IEEE

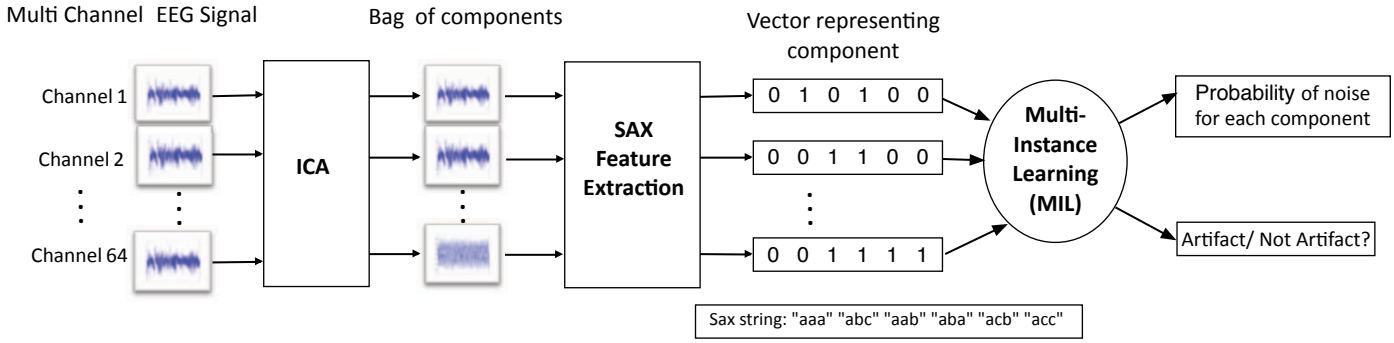

Fig. 1. System Architecture for recognizing Artifacts in EEG data including ICA for generating components, SAX feature extractor to create some features and Multi-instance classifier to find the probability of noise for each component.

duces a *SAX word* as output, a sequence of letters that characterize the time series. The algorithm requires two parameters: a word length w and an alphabet size a. The first step in the algorithm reduces the dimensionality of the component of length m by dividing it into $\lfloor m/w \rfloor$ adjacent, non-overlapping windows. The means of the raw data in each window are computed to create a reduced time series of length w containing just the means. This piecewise-aggregate approximation (PAA) reduces both the volume of data and smooths noise. Next, the mean and standard deviation of the PAA values are computed and the resulting normal distribution is divided into a equiprobable segments. Each successive segment is assigned a letter of the alphabet starting with `A` and each value in the PAA is assigned the letter corresponding to the normal segment into which it falls. Note that many raw time series with roughly the same shape will map to the same SAX word. Also, the word conveys the overall structure of the time series. [8] gives a more detailed explanation of using SAX to featurize time series data.

A list of SAX words is then created by sliding a window over all the ICA components in the entire training set. The most frequently occurring words are selected and given fixed integer IDs. The number of words selected is given by hyperparameter $num\_features$. Then we create $num\_feature$ dimensional vectors for each component, where i^{th} value of this vector contains the frequency of occurrence of i^{th} word. Fig. 1 shows an example of vector generated for each ICA component. The bag of these vectors are provided to multi instance learning for classification.

B. Multi instance learning

In machine learning, multiple-instance learning (MIL) is a form of learning with weak supervisory signals. In multi instance learning instead of receiving labels for each individual instances, the classifier receives labels for bags of instances. Individual bags can have many instances whose class labels are unknown. A bag is labeled positive if it contains at least one instance in it which is positive. Otherwise the bag is labeled negative. There are many MIL algorithms but all of them seek regions in the instance space that are more likely to occur in positive bags than negative bags.

In this case, bags of items correspond to a set of components for EEG data and the goal is to identify whether EEG data corresponding to the components contains artifact or not. We apply MIL algorithms to learn to identify bags (EEG segments converted to ICA components) that contain artifacts and those that do not. We examine several MIL algorithms that are based on Support Vector Machines(SVM) with different kernels and discuss their results in Section III-B.

III. Software Analysis

A. Dataset

Example movement data came from a dataset generated by the US Army Research Laboratory that has been previously discussed [10]. Briefly, participants performed a block of artifact-inducing facial and head movements, collected as part of a larger study. The exact details of each movement were not controlled by the experimenter, but rather left up to the participant to perform in their most natural manner. The seven movements included: clenching the jaw; moving the jaw vertically (like chewing gum); blinking both eyes (but without squinting); moving eyes leftward, then back to center; moving eyes upwards, then back to center; raising and lowering eyebrows; and rotating head side-to-side (as in looking leftward). Each type of movement was performed in a separate run consisting of 20 repetitions; at the beginning of each run, instructions appeared on the screen reminding the participant of which movement should be made. For each run, a male voice initially counted down from 3 at a rate of every 2 s, followed by a tone every 2 s, and participants made the movement in time with the tone. The participants were told to make the movement for the first second of the 2 second period, and then to return to a relaxing state for the remaining 1 s. Analyses here focus on on eyebrow movements, with intention of extrapolating results to other more complicated movements in the future. Fig 2 shows a sample window of EEG data when a participant is not performing any activity versus raising and lowering eyebrows.

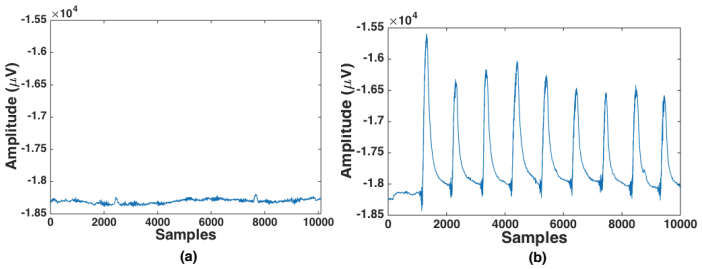

Fig. 2. EEG (single electrode) when participant is (a) not performing any activity and (b) raising and lowering eyebrows.

TABLE I. MIL CLASSIFIER ACCURACY FOR PREDICTING EEG ARTIFACT

Algorithm	Kernel	Accuracy
Normalized set kernel (NSK)	Linear	95.2%
MISVM	Linear	69.60%
miSVM	Linear	60.80%
MISVM	Quadratic	48.40%
miSVM	Quadratic	47.60%

B. Multi instance learning Evaluation

We evaluated our method by identifying artifacts corresponding to the actions of raising and lowering eyebrows. This includes 500 EEG signals, half of which contain artifact, with 64 channels each, yielding 64 components per ICA computation. We label bag of these 64 components as positive if corresponding signal has artifact, otherwise we label it as negative.

We then featurize all the components and use 250 randomly selected signals for training and remaining 250 for testing MIL algorithms. We experiment with different window size, alphabet size and word size for generating SAX parameters, and test our solution with 3 multi instance learning algorithms: miSVM, MISVM and Normalized Set Kernel (NSK). miSVM and MISVM methods modify the standard SVM formulation so that the constraints on instance labels correspond to the MI assumption that at least one instance in each bag is positive. The NSK uses kernels to map entire bags into features, then use the standard SVM for training classifier. Table I shows the accuracy of different classifiers. The normalized set kernel consistently performs better than other MIL algorithms for identifying artifacts.

C. Hyperparameter optimization for accuracy and speed

We did extensive hyperparameter optimization with goal of improving accuracy without increasing execution time. Fig. 3 shows the accuracy of MIL algorithms as the number of sub-sequences generated by SAX is varied. We observe that changes in number of sub sequences has considerable effect on the accuracy of artifact identification. In our system there are several hyper-parameters for ICA, SAX and MIL. ICA parameters include number of components and number of iterations. SAX parameters are word size, alphabet size, window overlap and number of features for SAX. MIL parameters are type of kernel, maximum number of iterations, slack variable and kernel specific parameters. Because of this reason we would use hyperparameter optimization [11] for selecting parameters in the future. We observe that with increase in $num\_features$, accuracy increases but execution time also increases. To speedup the algorithm we optimize to reduce $num\_features$, without significant decrease in accuracy. The most optimized model has parameters $num\_features = 200$, $word\_size = 5$, $alphabet\_size = 4$, $overlapping\_fraction = 0.8$, quadratic kernel and achieves accuracy of 91.2%. We use this optimized model in Section V to get on board power and execution time measurements. Note that, in this experiment we use entire patients data for generating components using ICA. This is because ICA needs large amount of data to get reasonable components. In this paper, we focus on optimizing SAX and SVM for artifact identification. We leave optimizing amount of data needed for ICA to get reasonable components for future work. In Section V, we use subset of data for generating components for measuring on board power and execution time.

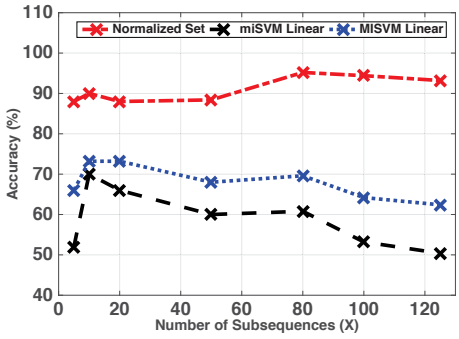

Fig. 3. Effect of number of SAX sub-sequences on accuracy of artifact identification where $num\_features = 8000$, $word\_size = 8$, $alphabet\_size = 4$, $overlapping\_fraction = 0.9$. The optimal number of subsequences is 10 to achieve 92% accuracy, without optimizing for $num_feature$ optimization.

IV. HARDWARE DESIGN OBJECTIVES

Fig. 4 shows the hardware architecture of the proposed ICA-SAX-MIL processor that can perform artifact detection in realtime. It consists of 1) ICA Processor which can separate out artifacts that are existing in time series from multi-channel EEG data. ICA includes one sub-module for pre whitening the data and another one for extracting multiple components. 2) A SAX feature extractor which generates different features to be used as inputs to the machine learning classifier. 3) MIL processor which uses machine learning kernels such as SVM to detect artifact.

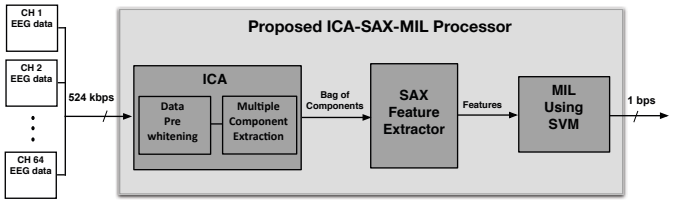

Fig. 4. Hardware architecture used in the proposed system which includes ICA, SAX and MIL kernels. The architecture is implemented on an embedded off-the-shelf ARM A15 processor in NVIDIA Jetson TK1 platform.

V. HARDWARE IMPLEMENTATION AND RESULTS

The complete proposed system including ICA, SAX and MIL kernels was implemented and tested on an embedded off-the-shelf NVIDIA TK1 board. NVIDIA Jetson TK1 is a System-on-Chip (SoC) combining the Kepler graphics processing unit (GPU) and a 4-Plus-1 ARM processor arrangement. For this implementation, we only used one Cortex A15 ARM CPU which can run up to 2.3 GHz, and has 32KB L1 data and instruction cache supporting 128-bit NEONTM general-purpose single instruction and SIMD instructions.

Fig. 5 shows the experimental setup for power measurement of the TK1 board. Power consumption is measured by collecting the current consumed at the CPU core using a

current monitor IC (TI INA219) and sampled by an Arduino Uno board.

Fig. 5. Experimental setup for power measurement of TK1 board including, INA219 power monitor sensor, Arduino Uno microcontroller and TK1 CPU-GPU boards.

Fig. 6 shows the execution time and energy consumption of SAX and SVM algorithm together versus the processor frequency using the parameters that were discussed in Section III-C. The execution time and energy consumption are reduced from 2.6 s and 9.4 J to 700 ms and 3.64 J, respectively when the frequency increases from 584 MHz to 1.8 GHz. Note that these results are for the system running on a single ARM A15 CPU without any parallelization. The total execution time when we initially implemented the model on the ARM CPU was 282 s and we could reduce it down to total of 8.86 s at 1.6 GHz clock frequency after hyperparameter optimization.

TABLE II. ON-BOARD POWER MEASUREMENT RESULTS OF THE PROPOSED SYSTEM. THE MEASURED POWER CONSUMPTION OF THE BOARD WHILE PROCESSOR IS IDLE IS 2.32 W.

Design	ICA	SAX+SVM	Total
Operating Freq. (MHz)	828	828	828
Processing Power Consumption (W)	0.94	0.87	0.9
Total Power Consumption (W)	3.26	3.19	3.22

The performance results of the proposed system, including ICA, SAX feature extractor and MIL implementation on ARM CPU core is provided in Table II. To have a more accurate current measurement, the experiments were performed over 45 iterations. At 800 MHz clock frequency with a 12 V power supply, the total power consumption of the proposed system is 3.22 W. However, it should be noted that the vast majority, >90%, of the power consumption is while the processor is in idle, suggesting that the actual processing is consuming very little. This is critical because when scaled down to a discrete dedicated IC design for this purpose the total consumption will be dramatically less.

VI. CONCLUSION

This paper presents an EEG artifact identification software-hardware embedded system, using ICA, SAX feature extraction and Multi-Instance Learning algorithms. The proposed system uses a weak supervisory signal to indicate that some noise is occurring, but not what the source of the noise is or how it is manifested in the EEG signal. The proposed system has been implemented on an embedded off-the-shelf ARM A15 CPU processor in NVIDIA Jetson TK1 board. While the overall power consumption is 3.22 W, most of the power is consumed during idle, demonstrating strong efficiency

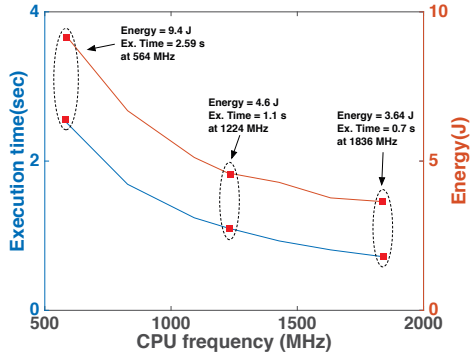

Fig. 6. Execution time and energy consumption of SAX+SVM processor versus CPU frequency

while also maintaining 91.2% accuracy. In future, we intend to optimize ICA in order to run smaller data reducing execution time even further.

ACKNOWLEDGMENT

This research was sponsored by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-10-2-0022.

REFERENCES

[1] A. Jafari, A. Page, C. Sagedy, E. Smith, and T. Mohsenin, "A low power seizure detection processor based on direct use of compressively-sensed data and employing a deterministic random matrix," in *Biomedical Circuits and Systems Conference (BioCAS), 2015 IEEE*. IEEE, 2015, pp. 1–4.

[2] A. Page, C. Shea, and T. Mohsenin, "Wearable seizure detection using convolutional neural networks with transfer learning," in *Circuits and Systems (ISCAS), 2016 IEEE International Symposium on*, 2016.

[3] A. Page, A. Kulkarni, and T. Mohsenin, "Utilizing deep neural nets for an embedded ecg-based biometric authentication system," in *Biomedical Circuits and Systems Conference (BioCAS), 2015 IEEE*, Oct 2015, pp. 1–4.

[4] D. M. White and C. A. Van Cott, "Eeg artifacts in the intensive care unit setting," *American journal of electroneurodiagnostic technology*, vol. 50, no. 1, pp. 8–25, 2010.

[5] A. Delorme, T. Sejnowski, and S. Makeig, "Enhanced detection of artifacts in eeg data using higher-order statistics and independent component analysis," *Neuroimage*, vol. 34, no. 4, pp. 1443–1449, 2007.

[6] S. Andrews, I. Tsochantaridis, and T. Hofmann, "Support vector machines for multiple-instance learning," in *Advances in neural information processing systems*, 2002, pp. 561–568.

[7] T. Gärtner, P. A. Flach, A. Kowalczyk, and A. J. Smola, "Multi-instance kernels." in *ICML*, vol. 2, 2002, pp. 179–186.

[8] J. Lin, E. Keogh, L. Wei, and S. Lonardi, "Experiencing sax: a novel symbolic representation of time series," *Data Mining and knowledge discovery*, vol. 15, no. 2, pp. 107–144, 2007.

[9] S. P. Fitzgibbon, D. M. Powers, K. J. Pope, and C. R. Clark, "Removal of eeg noise and artifact using blind source separation," *Journal of Clinical Neurophysiology*, vol. 24, no. 3, pp. 232–243, 2007.

[10] V. Lawhern, W. D. Hairston, K. McDowell, M. Westerfield, and K. Robbins, "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.

[11] J. Bergstra, D. Yamins, and D. D. Cox, "Hyperopt: A python library for optimizing the hyperparameters of machine learning algorithms," in *Proceedings of the 12th Python in Science Conference*. Citeseer, 2013, pp. 13–20.