An Ultra Low Power Feature Extraction and Classification System for Wearable Seizure Detection

Adam Page, Siddharth Pramod Tim Oates and Tinoosh Mohsenin
University of Maryland, Baltimore County

***Abstract*— In this paper we explore the use of a variety of machine learning algorithms for designing a reliable and low-power, multi-channel EEG feature extractor and classifier for predicting seizures from electroencephalographic data (scalp EEG). Different machine learning classifiers including k-nearest neighbor, support vector machines, naïve Bayes, logistic regression, and neural networks are explored with the goal of maximizing detection accuracy while minimizing power, area, and latency. The input to each machine learning classifier is a 198 feature vector containing 9 features for each of the 22 EEG channels obtained over 1-second windows. All classifiers were able to obtain F1 scores over 80% and onset sensitivity of 100% when tested on 10 patients. Among five different classifiers that were explored, logistic regression (LR) proved to have minimum hardware complexity while providing average F-1 score of 91%. Both ASIC and FPGA implementations of logistic regression are presented and show the smallest area, power consumption, and the lowest latency when compared to the previous work.**

I. Introduction

Collecting non-invasive health and vital signs data on a regular basis is becoming more common with advancement in bio-sensor and wearable device technology. With the right models, using such data for real-time detection of ongoing health-related events may soon become reality, enabling swift remedial action to be taken when possible.

One such application is the detection of epileptic seizures using electroencephalography (EEG). In a clinical setting, electroencephalography combined with video monitoring is the de facto gold standard for the detection and diagnosis of various neurological conditions including seizures [1]. Although detecting seizures that occur in daily life is important for the safety and well-being of those with seizures and those around them, clinical systems are far too resource intensive for ambulatory settings. To be of practical use, models must detect seizure events quickly, with high *sensitivity* (minimal false negatives) as well as high *specificity* (minimal false positives or false alarms). This requires training models that generalize well to previously unseen data.

In this paper, we explore various machine learning techniques such as logistic regression, naïve Bayes, *k*-nearest neighbor, support vector machines and neural networks, that can be trained to detect seizures in EEG data. The goal is to understand the trade-offs between detection accuracy and area, power, and latency when implemented in hardware suitable for use in daily living.

II. Classifier and Feature Extraction Analysis

A. Seizure Data and Approach

Data from the CHB-MIT database [2] is used to train and validate the models. This database contains 23 multi-channel scalp EEG records sampled at 256 Hz, collected from 22 patients over several days. Each patient's data, consists of several consecutive, 1-hour long records. For our experiments, we performed personalized per-patient training using a leave one record out cross validation approach [3]. We classify 1-second segments of this data as being in either a normal state or a seizure state. Classification of longer segments with durations greater than 23-seconds has been studied well, and existing techniques achieve high classification performance [4]. However, given the length of these segments, the proposed approaches would require a significant delay for detection after seizure onset. Using short segments allows for faster detection of seizure events which would then enable swift remedial action.

B. Feature Extraction and Optimization

Rather than using the raw EEG signal as input, the following 9 features described by [5] are extracted from each segment in each channel: Area, Normalized Decay, Line Length, Mean Energy, Average Peak Amplitude, Average Valley Amplitude, Normalized Peak Number, Peak Variation, and Root Mean Square. Hence each instance comprises 9 features x 22 channels = 198 multi-channel features. These features have previously been used successfully in other applications [5], [6], [7]. The label for each instance is either 0 (normal) or 1 (seizure) where 1 corresponds to an ongoing seizure event within the 1-second segment.

C. Machine Learning Algorithms

We evaluate five types of binary classifiers, k-nearest neighbor classifiers (KNN) with k = 3, 5, and 7 neighbors, support vector machines (SVM) with linear and polynomial kernels, logistic regression (LR), naïve Bayes (NB) and neural networks (NN).

KNN classifiers classify test instances based on the most common label among it's k nearest neighbors, based on a specified distance metric. We use Euclidean distance. These distances are calculated between the test instance and each instance in the training set, while simultaneously tracking the k training instances nearest to the test instance.

SVMs learn a maximum-margin hyperplane, i.e. a linear separator that maximally separates the two classes. Kernel functions may be used to perform non-linear transformations of the data into a high-dimensional space. A linear classifier in this high-dimensional space is then equivalent to a non-linear classifier in the input space. Hence SVMs may be used to learn non-linear classifiers using kernel functions.

Logistic regression maps a weighted linear combination of the feature values to a real value between 0 and 1 using the logistic (sigmoid) function. These real values may be

978-1-4244-9270-1/15/$31.00 ©2015 IEEE

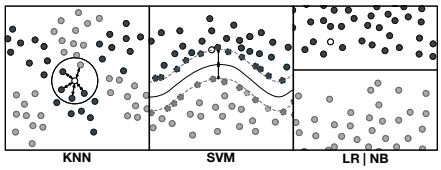


Fig. 1. K-nearest neighbor (KNN) classifier labels an instance by a majority vote of its k nearest neighbors according to some distance function. Support vector machines (SVMs) learn linear separators in high-dimensional spaces. Kernel functions can be used to separate non-linear feature space. Naïve Bayes (NB) and logistic regression (LR) are both linear classifiers based on conditional probabilities and log-likelihood ratios respectively. Neural Networks (NN) can learn non-linear decision boundaries similar to SVMs that use non-linear kernel functions.

interpreted as probabilities. Binary classification is performed by setting a threshold, and determining whether the output of logistic regression is greater than or less than the threshold. The threshold is typically set at 0.5, but may be tuned based on classification performance. Such classifiers form a linear decision boundary.

Naïve Bayes classifiers are probabilistic classifiers that use Bayes' theorem in probability to calculate the posterior probabilities of each class given the value of each feature in a test instance. The training set is used to estimate the prior and likelihood. Naïve Bayes classifiers make an assumption that features are independent of each other given the class label. Since the input is continuous, we use Gaussian Naïve Bayes, which assumes that data belonging to a specific class is normally distributed. Classification is performed by selecting the class label with the highest posterior probability. As with logistic regression, such classifiers also form a linear decision boundary.

Neural Networks are non-linear classifiers that work by stacking multiple layers of non-linear transformations on top of the preceding layer. The features in the given dataset form the first layer. The final layer may be any classifier, we use the softmax classifier. We use the rectified linear unit (ReLU) [8] as the non-linear transformation, which is commonly used to speed up computation over the sigmoid functions that may be used alternatively. The use of *dropout* [9] during training is known to reduce overfitting and improve generalization, which we employ in our experiments.

A schematic description of these classifiers is shown in Figure 1.

III. Classifier Algorithms Comparison

The bar plots in Fig. 2 show the comparison between the best classifiers found when single patient data is used for training and testing based on the F_1 score. All classifiers achieved an F_1 score over 80% with LR performing the best overall. Classification using data from the same patient for training and testing is generally an easier task than drawing the training and test data from different patients, so the differences between the neural network and the logistic regression are not significant on single patient training. The neural network algorithm is very effective at detection, with two perfect F_1 measures, and only one F_1 measure below 0.9. These same tests were also run against the same implementation of logistic regression that is used in the output layer of the neural network, with F_1 score comparisons shown in Figure 3.

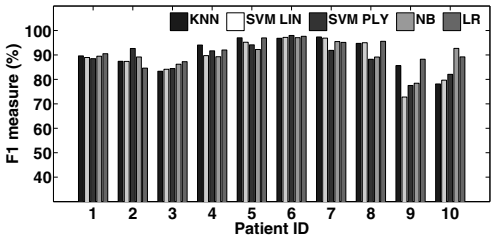


Fig. 2. Comparison of F1 measures for four classifiers with the 9 features when single patient data is used for both training and testing. All classifiers are able to get over 80% for almost all patients.

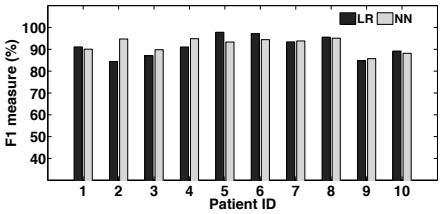


Fig. 3. Comparison of F1 measures for logistic regression (LR) alone and a neural network coupled with logistic regression using 9 features per channel. Addition of neural network provides little accuracy gain over just LR.

IV. Feature Extraction and Classifier Hardware

Figure 4 depicts the top-level block diagram of the proposed feature extraction and classifier system. Although this is not the focus of this paper, we envision that external to the processor there will be peripheral boards each consisting of multi-channel AFEs, such as TIs ADS1299. The AFEs will be connected via a shared bus to a discrete SPI controller with each channel being sampled at 256Hz. A state machine implemented in the FPGA would use the SPI controller to read one second windows (256 samples/channel) from the AFEs and buffer the data in an internal FIFO. Buffering a channel of samples allows the system to fetch the next window's samples while processing the current window, reducing the effective latency. The "Feature Extractor" block calculates and outputs the features in parallel to the "Feature Serializer". The serializer buffers the features and outputs them serially to the classifier block. The "Feature FSM" block is responsible for managing both the extractor and serializer in addition to reading samples from the FIFO.

For comparison and testing purposes, a single classifier is selected and instantiated at compile time to avoid using unneeded hardware resources. The classifier outputs one classification per channel to the "Multi-Channel Vote" block, which employs a simple voting methodology. The final decision is reported as a seizure if the number of channels that are classified as a seizure exceeds a predefined threshold.

A detailed architecture of the feature extraction process is shown in Fig. 5(a). The block receives 256 samples serially from which it calculates the features in parallel. Details of the classifiers used for the proposed system are shown in Fig. 5 b, c, d, e and f.

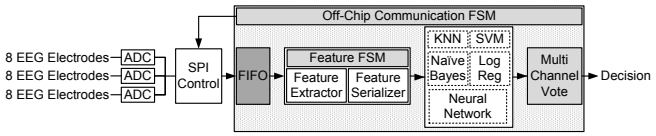
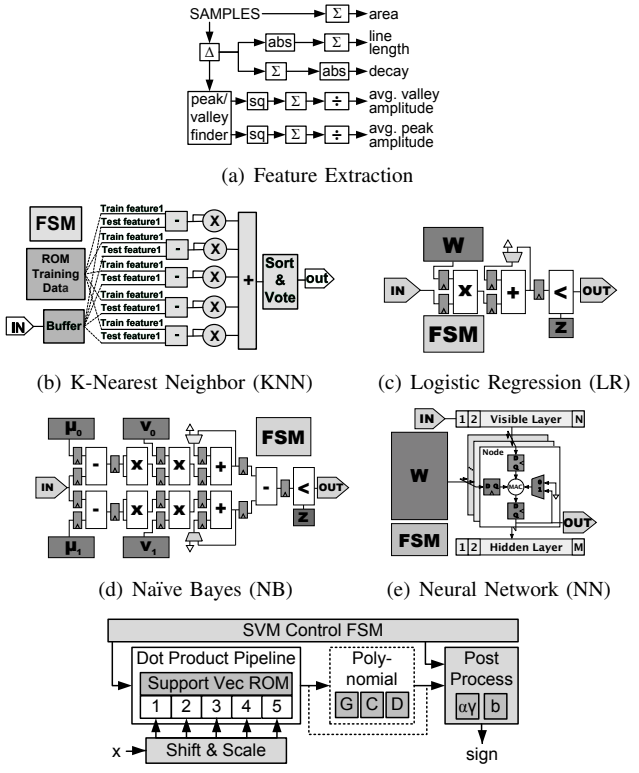


Fig. 4. Block diagram of the flexible seizure detection processor containing feature extraction, classifier, multi-channel vote, and IO interface. Note: SPI and AFE blocks are not implemented.



(a) Feature Extraction

(b) K-Nearest Neighbor (KNN)

(c) Logistic Regression (LR)

(d) Naïve Bayes (NB)

(e) Neural Network (NN)

(f) Support Vector Machine (SVM)

Fig. 5. Block diagrams of the Feature Extraction and reconfigurable classifiers including Logistic Regression, Naïve Bayes, Support Vector Machine, K-Nearest Neighbor and Neural Network. The parameters for the classifier algorithms are explained in section II.

1) Computational and memory complexity requirements: Besides the ability for the classifiers to accurately predict seizures, it is also essential to minimize complexity in order to meet the strict area and power limitations of a long-term wearable system. Since the device can be trained offline, the complexity comes in the form of memory required to store the classifier's model and the computation required to classify real-time data.

The plots in Fig. 6 summarize the experimental memory and computational complexity for each of the classifiers, respectively. These figures also include the feature extraction stage requirements. Also included in the comparison is condensed Nearest Neighbor (CNN), an optimization applied to KNN that attempts to remove low-content model data while maintaining nearly the same accuracy.

As seen in the figures, KNN did by far the worst for both cases. This is to be expected since KNN requires storing all of the unique training data and labels. For the experimental values, KNN required 100,000x more memory and over

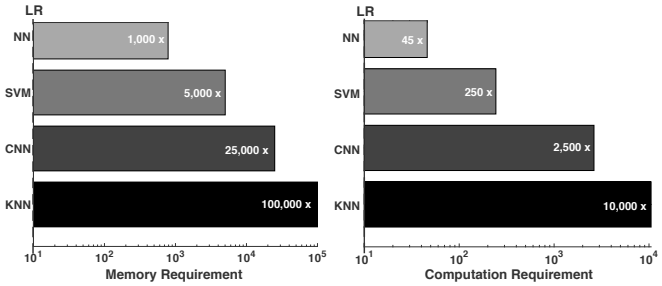


Fig. 6. Comparison of classifiers memory and computation complexity relative to logistic regression.

10,000x more computations than logistic regression. The CNN variant of KNN managed to reduce the memory and computation by roughly 75%. However, CNN still required much more memory and computation compared to LR. Both neural networks and SVM performed on the same level with SVM needing 5x more memory and computation than NN per classification. The larger requirements for SVM was due to having a high number of support vectors (roughly 5% of the training data). Logistic regression did the best in terms of hardware efficiency while still achieving high seizure accuracy. Thus for hardware implementation, we selected LR as the classifier.

V. FPGA and ASIC Implementation and Comparison

A. Architecture and Fixed-Point Word Width Optimization

A key consideration for the proposed system was to balance the performance and area. The decision of whether or not a seizure occurred should be output quickly, but the design's area and power should be kept at a minimum. Given the half-second interval allotted for computation, both the feature extraction and classification of each channel could be done serially. Furthermore, the classifiers' algorithms could likewise be implemented serially. More details on hardware implementation are discussed in [10].

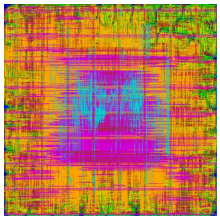
B. Resource and Power Results

Table I compares the performance of fully placed and routed classifiers with feature extraction implemented in Virtex-5 FPGA. All delay, power, and area results are for a complete system processing 256 samples per window. The results show that the LR classifier ranks best in terms of FPGA resource utilization, dynamic power consumption and computation latency. Additionally, having the highest F1 score and lowest variation for the studied patients, this classifier is the best candidate for the seizure detection application. NB classifier ranks second. Although the KNN classifier performs better than the linear SVM in terms of F1, it consumes dramatically more FPGA resources and power. The particular results are based on 50% overlapping windows that provide a half-second interval for featurization and classification. If instead completely overlapping windows are used, then there is only a 3.9 ms interval. In this case, KNN would not have sufficient time to classify. Increasing the overlap may prove useful to reduce onset detection latency from 0.5 sec to 0.004 sec. Regardless, for the proposed system, the latencies are all relatively small for a wearable

Design	KNN3	SVM LIN	SVM PLY3	NB	LR
Onset Sensitivity[1]	100.00	100.00	100.00	100.00	100.00
Window Sensitivity[1]	94.44	95.00	95.39	93.65	95.24
False-alarms/hr	0.50 ± 0.96	0.50 ± 1.16	0.47 ± 1.09	0.46 ± 0.53	0.45 ± 0.56
F1 Accuracy (%)	90.05	88.69	90.50	90.34	91.16
Logic Slices	3788	4281	3629	2987	2583
Memory (KB)	2916.4	828.4	792.4	0.26	0.30
Max Freq (MHz)	131	152	150	101	134
Latency (cycle)	644,622	108,665	91,928	264	242
Latency (ms)	4.9	0.7	0.6	0.002	0.0018
Nominal Freq (kHz)[2]	1286.01	217.33	183.86	0.53	0.48
Dynamic Power (μW)[3]	2697.76	192.96	131.83	0.171	0.098
Leakage Power (mW)	1046.41	1044.36	1043.96	1042.88	1042.65
Energy (μJ)[2]	1348.9	96.5	65.9	0.09	0.05

TABLE I

COMPARISON OF 22-CHANNEL SEIZURE DETECTION HARDWARE IMPLEMENTATION (CLASSIFIER + FEATURE EXTRACTION) FOR DIFFERENT CLASSIFIERS ON VIRTEX-5 FPGA. 1. ONSET AND WINDOW SENSITIVITY ARE DEFINED AS % OF CORRECTLY IDENTIFIED SEIZURE ONSETS AND WINDOWS, RESPECTIVELY. 2. THE POWER RESULTS ARE FOR NOMINAL FREQUENCY TO MEET HALF-SECOND WINDOW INTERVALS. 3. SINCE FPGA HAS VERY LARGE LEAKAGE POWER (DOMINANT COMPARED TO DYNAMIC POWER), THE ENERGY RESULTS ARE BASED ON DYNAMIC POWER ONLY.



Implementation Results	
Technology	65 nm, 1 V
Logic Utilization	95%
Area	0.008 mm^2
Max Freq.	1.0 GHz
Nominal Freq.	484 Hz
Total Power*	37 nW
Energy*	19 nJ

Fig. 7. Layout view and post-layout simulation results of the proposed seizure detection with optimized feature extraction and logistic regression classifier. *The power and energy are reported for the nominal frequency where the computation is done in half-second interval window.

seizure detection problem. The more important metrics are the power, area, and accuracy.

VI. ASIC IMPLEMENTATION AND RESULTS

The seizure detection processor with the LR classifier configuration is also synthesized and placed and routed in the 65 nm TSMC CMOS technology. Figure 7 shows the layout of the proposed seizure detection hardware (feature extraction + LR classifier) and post-layout simulation results. The processor occupies 0.008 mm^2 and dissipates approximately 77 mW when simulated at its maximum frequency of 1.0 GHz. When running at the nominal frequency of 484 Hz required to meet the half-second window, the design dissipates approximately 37 nW (linearly scaled with frequency) which results in 19 nJ at 1 V to classify 256 sample input.

Design	[11]	[12]	This Work
Onset Sensitivity	100.00	100.00	100.00
Window Sensitivity	82.70	93.80	95.24
# Channels	8	18	22
# Feature computations	8× 7=56	18× 3 × 8=432	22× 5=110
Classifier	SVM	SVM	LR
Computation latency	< 2 sec	1 sec	0.5 sec
Operating Freq	512 KHz	2 MHz	484 Hz
Technology	180 nm, 1 V measured	130 nm, 0.85 V measured	65 nm, 1 V postlayout
Energy per classification	2.03μJ Analog+Digital	273 μJ Analog+Digital	19 nJ Digital

TABLE II

COMPARISON WITH PREVIOUS WORK ON SEIZURE DETECTION PROCESSORS

We used a standard-cell RTL to GDSII flow using synthesis Cadences RTL compiler and place and route SOC Encounter to implement the chip. Table II compares the proposed 22-channel feature extraction + LR processor with multi-channel seizure detection processors in [12], [11]. The SoC power consumption for both previous works include AFEs and ADCs.

VII. CONCLUSION

This paper presented a low-power, multi-channel architecture designed to perform personalized seizure detection. It contrasted several classifiers in terms of accuracy, area, and power to identify the best choice for this application. Ultimately, the optimal choice of classifier was logistic regression, which had the best average F1 measure of 91%, the smallest area and power footprint, and the lowest latency. When implemented on a Virtex-5 FPGA, the logistic regression based system occupies 33% fewer slice resources and consumes 26,960x less energy than a KNN-based processor. The ASIC implementation of the same combination in 65 nm CMOS demonstrated that the design fits in chip area of 0.008 mm^2 with a maximum clock frequency of 1.0 GHz, and power and energy consumption of 37 nW and 19 nJ respectively at 484 Hz.

REFERENCES

[1] S. Benbadis, W. LaFrance *et al.*, "Interrater reliability of eeg-video monitoring," *Neurology*, vol. 73, no. 11, pp. 843–846, 2009.

[2] A. L. Goldberger, L. A. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley, "Physiobank, physiotoolkit, and physionet components of a new research resource for complex physiologic signals," *Circulation*, vol. 101, no. 23, pp. e215–e220, 2000.

[3] N. Verma, A. Shoeb *et al.*, "A micro-power eeg acquisition soc with integrated feature extraction processor for a chronic seizure detection system," *Solid-State Circuits, IEEE Journal of*, vol. 45, no. 4, 2010.

[4] G. Chen, "Automatic eeg seizure detection using dual-tree complex wavelet-fourier features," *Expert Systems with Applications*, vol. 41, no. 5, pp. 2391–2394, 2014.

[5] D. Wulsin, J. Gupta, R. Mani, J. Blanco, and B. Litt, "Modeling electroencephalography waveforms with semi-supervised deep belief nets: fast classification and anomaly measurement," *Journal of neural engineering*, vol. 8, no. 3, p. 036015, 2011.

[6] A. Page, J. Turner, T. Mohsenin, and T. Oates, "Comparing raw data and feature extraction for seizure detection with deep learning methods," in *The Twenty-Seventh International Flairs Conference*, 2014.

[7] J. Turner, A. Page, T. Mohsenin, and T. Oates, "Deep belief networks used on high resolution multichannel electroencephalography data for seizure detection," in *2014 AAAI Spring Symposium Series*, 2014.

[8] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, 2010, pp. 807–814.

[9] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint arXiv:1207.0580*, 2012.

[10] A. Page, C. Sagedy, E. Smith, N. Attaran, T. Oates, and T. Mohsenin, "A flexible multichannel eeg feature extractor and classifier for seizure detection," *Circuits and Systems II: Express Briefs, IEEE Transactions on*, vol. 62, no. 2, pp. 109–113, Feb 2015.

[11] J. Yoo, L. Yan *et al.*, "An 8-channel scalable eeg acquisition soc with patient-specific seizure classification and recording processor," *Solid-State Circuits, IEEE Journal of*, vol. 48, no. 1, pp. 214–228, 2013.

[12] K. H. Lee and N. Verma, "A low-power processor with configurable embedded machine-learning accelerators for high-order and adaptive analysis of medical-sensor signals," *Solid-State Circuits, IEEE Journal of*, vol. 48, no. 7, pp. 1625–1637, July 2013.