A Flexible Multichannel EEG Feature Extractor and Classifier for Seizure Detection

Adam Page, Chris Sagedy, Emily Smith, Nasrin Attaran, Tim Oates, and Tinoosh Mohsenin

***Abstract*—This brief presents a low-power, flexible, and multichannel electroencephalography (EEG) feature extractor and classifier for the purpose of personalized seizure detection. Various features and classifiers were explored with the goal of maximizing detection accuracy while minimizing power, area, and latency. Additionally, algorithmic and hardware optimizations were identified to further improve performance. The classifiers studied include *k*-nearest neighbor, support vector machines, naïve Bayes, and logistic regression (LR). All feature and classifier pairs were able to obtain F1 scores over 80% and onset sensitivity of 100% when tested on ten patients. A fully flexible hardware system was implemented that offers parameters for the number of EEG channels, the number of features, the classifier type, and various word width resolutions. Five seizure detection processors with different classifiers have been fully placed and routed on a Virtex-5 field-programmable gate array and been compared. It was found that five features per channel with LR proved to be the best solution for the application of personalized seizure detection. LR had the best average F1 score of 91%, the smallest area and power footprint, and the lowest latency. The ASIC implementation of the same combination in 65-nm CMOS shows that the processor occupies 0.008 mm^2 and dissipates 19 nJ at 484 Hz.**

***Index Terms*—ASIC, electroencephalography (EEG), field-programmable gate array (FPGA), *k*-nearest neighbor (KNN), logistic regression (LR), low power, naïve Bayes (NB), personalized seizure detection, support vector machines (SVMs).**

I. INTRODUCTION

PERSONALIZED healthcare depends crucially on large volumes of data about both individuals and populations. It is easy to imagine a near future in which it is common to wear a number of biosensors that continuously monitor various aspects of our physiological state, including heart rate, blood pressure, eye movement, brain activity, and many others. There are two aspects of this enterprise, i.e., gathering the data and efficiently doing something useful with it [1]–[3]. The focus of this brief is the latter in the context of seizure detection.

In a clinical setting, electroencephalography (EEG) combined with video monitoring is the *de facto* gold standard for the detection and diagnosis of various neurological conditions, including seizures [4]. 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. This brief explores a number of different machine learning methods [i.e., logistic regression (LR), k-nearest neighbor (KNN), support vector machines (SVMs), and naïve Bayes (NB)] that can be trained to detect seizures in EEG data. The goal is to understand the tradeoffs between detection accuracy and area, power, and latency when implemented in hardware suitable for use in daily living.

The remainder of this brief describes the various classifiers and feature extraction, how these are implemented in hardware [including field-programmable gate arrays (FPGAs) and an ASIC design], and detailed analysis of accuracy and resource use, with concrete recommendations on which elements are most suitable for personalized healthcare applications.

II. CLASSIFIER AND FEATURE EXTRACTION ANALYSIS

A. Seizure Data and Approach

The data used to evaluate the performance of our system consist of EEG recordings from 22 subjects (5 males, ages 3–22, and 17 females, ages 1.5–19) with intractable seizures collected at Boston Children's Hospital [5]. Each subject was monitored for 9–42 h continuously, sampled at 256 Hz with 16-bit resolution using a standard 10–20 montage. Clinicians monitoring the patients marked start and stop times of all seizures. For our experiments, only 10 of the 22 patients were used. Patients under the age of 5 were excluded since electrical activity of young children can look quite different than that of older children and adults. In addition, the data from the 10 patients selected had the same 22 EEG channels. In all, there was a total of 336 h of data containing 62 seizure onsets.

B. Classifier Selection and Optimization

Four binary classifiers were used and evaluated with respect to accuracy and resource requirements, including KNN with three, five, and seven neighbors; SVMs with linear and polynomial kernels; LR; and NB.

The KNN algorithm classifies test data by a majority vote of the known labels of the KNNs using some distance metric such as Euclidean distance. Training consists of simply storing all of the labeled training data. Classification requires computing the distances between the test instance and each training instance while tracking the k smallest distances.

SVMs work by finding the maximum margin hyperplane, i.e., the linear separator that is as far as possible from the closest positive and negative training instances. Kernel functions can be used to project the data into a high-dimensional space such that the linear separator is highly nonlinear in the input space. The default linear and polynomial kernels are in (1) and (2), respectively, as

$$K(\vec{s_i},\vec{x}) = \vec{s_i}\cdot\vec{x} \tag{1}$$

$$K(\vec{s_i},\vec{x}) = (\gamma \times (\vec{s_i}\cdot\vec{x}) + b)^d . \tag{2}$$

Manuscript received September 30, 2014; revised November 17, 2014; accepted December 16, 2014. Date of publication December 23, 2014; date of current version February 7, 2015. This brief was recommended by Associate Editor R. Rieger.

The authors are with the University of Maryland, Baltimore County, MD 21250 USA.

Color versions of one or more of the figures in this brief are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSII.2014.2385211

Samples are then classified using the function shown in (3), where $\vec{s_i}$ is a support vector, $\vec{x}$ is a test vector, $K(\vec{s_i}, \vec{x})$ is the kernel function, α_i and γ_i are the weight and label of the support vector, and b is the bias

$$f(x) = sign\left(\sum_{i=1}^{NUM\_SV} \alpha_i \gamma_i K(\vec{s_i}, \vec{x}) + b\right). \quad (3)$$

LR is a discriminative probabilistic model that utilizes the logistic function to map real-valued inputs between 0 and 1, which can be interpreted as probabilities. Training is an iterative process that finds maximum-likelihood regression coefficients (weights). Samples are typically classified by selecting the label with the greater probability between $P(Y = 1|X)$ and $P(Y = 0|X)$ given in (4), where $w_{k,i}$ are the weights for each of the N features for the label, k, and X is the input vector as

$$P\left(Y = k|X\right) = \frac{1}{1 + \exp\left(w_{k,0} + \sum_{i=1}^{N} w_{k,i} x_i\right)}. \quad (4)$$

For the purpose of our work, minimizing computation is crucially important. Therefore, each classifier is mathematically optimized to decrease runtime computation.

For example, the original classification form of LR shown in (4) can be reduced by using probability ratios, as shown in (5). A seizure is classified if the final form is positive

$$\frac{P(Y = 1|X)}{P(Y = 0|X)} \Rightarrow \exp\left(w_0 + \sum_{i=1}^{N} w_i x_i\right) \Rightarrow \sum_{i=0}^{N} w_i x_i. \quad (5)$$

NB is another linear probabilistic classifier that applies Bayes' theorem and assumes independence between the features given the class label. During training, all conditional probabilities between features and class labels are computed along with the prior probabilities of the class labels. Since the input in our case is continuous, a Gaussian NB is used. Like LR, classification is done by selecting the label with the highest posterior probability given by (6) where k is the class label, P_k is the prior probability, X is the input vector, $\mu_{k,i}$ and $v_{k,i}$ are the mean and variance of feature i given label k as

$$P(Y{=}k|X){=}P_k \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi v_{k,i}}} \exp\left(\frac{-(x_i - \mu_{k,i})^2}{2v_{k,i}}\right). \quad (6)$$

Similar to LR, the original NB classification model given in (6) can be also substantially reduced in complexity, as shown in (7a)–(7c), by utilizing logs to convert products of sequences into summations.

The final form is given in (7c) where the right-hand expression is a precomputed constant as

$$P(Y = 1|Xt) > P(Y = 0|X) \quad (7a)$$

$$\Rightarrow P_1 \prod_{i=1}^{N} \frac{\exp\left(\frac{-(x_i - \mu_{1,i})^2}{2v_{1,i}}\right)}{\sqrt{2\pi v_{1,i}}} > P_0 \prod_{i=1}^{N} \frac{\exp\left(\frac{-(x_i - \mu_{0,i})^2}{2v_{0,i}}\right)}{\sqrt{2\pi v_{0,i}}} \quad (7b)$$

$$\Rightarrow \sum_{i=1}^{N} \frac{(x_i - \mu_{1,i})^2}{v_{1,i}} - \sum_{i=1}^{N} \frac{(x_i - \mu_{0,i})^2}{v_{0,i}} > \underbrace{2\log\left(\frac{P_1}{P_0}\right) + \sum_{i=1}^{N} \log\left(\frac{v_{0,i}}{v_{1,i}}\right)}_{\text{constant}}. \quad (7c)$$

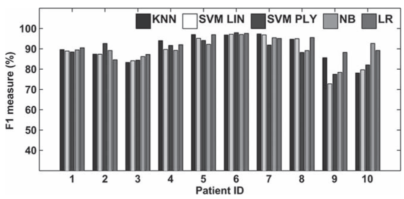


Fig. 1. Comparison of F1 measures for four classifiers with the original nine features when single patient data are used for both training and testing. All classifiers are able to get over 80% for almost all patients.

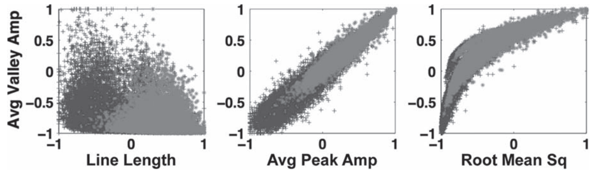


Fig. 2. Comparison of different feature pairs' information gain to optimize the number of features. Average valley amplitude versus line length provides more information than average peak amplitude or root mean square.

C. Feature Extraction and Optimization

Using raw time-series data as input to most classifiers results in low accuracy, making feature extraction crucial. Because a trained human can look at the EEG waveforms and determine whether or not a seizure is beginning with close to perfect accuracy, many of the features summarize the shape of the time series. The features include area under the wave, normalized decay, line length, mean energy, average peak amplitude, average valley amplitude, peak variation, and root mean square and were taken from [6], which contains detailed descriptions. The features were obtained for each EEG channel using a 1-s window with 50% overlapping windows. This, therefore, provided a half-second time interval to extract the features for all channels and to classify the window as either normal or onset. For our study, the training, validation, and testing sets were all drawn from the same patient. The fractions of total seconds to each of the sets are as follows: 67% training set, 7% validation set, and 26% testing set [7], [8]. Each classifier contains a number of training parameters. A sweep of these parameters was initially performed to find good values. These parameters were kept consistent across all patients, and repeated experiments were performed with randomized training/test data to prevent overfitting. The bar plots in Fig. 1 show the F_1 comparison between the best classifiers found when single patient data are used for training and testing based on all nine features. All classifiers got an F1 over 80% with LR performing the best overall. Using all nine features as a baseline, various subsets of the features were compared to determine the minimum set of features that would provide similar accuracy. As shown in Fig. 2, some features are redundant or highly correlated.

After some analysis, it was found that only five of the features are needed to obtain nearly identical F1 scores. These features include area under the wave, normalized decay, line length, average peak amplitude, and average valley amplitude. The formulas for these five features are given in Table I. The box plots in Fig. 3 show that the average patient F1 measure using all nine features closely matches that of the reduced five features.

The final efficacy metrics for each classifier when trained using the five features are summarized in Table II. The

TABLE I
FORMULAS FOR THE REDUCED FIVE FEATURES: AREA UNDER THE WAVE, NORMALIZED DECAY, LINE LENGTH, AVERAGE PEAK AMPLITUDE, AND AVERAGE VALLEY AMPLITUDE.
$W = window\ length, x = input, P = \#\ peaks, V = \#\ valleys$

Area Under Curve	Normalized Decay	
$A = \frac{1}{W} \sum_{i=0}^{W-1} x_i$	$D = \lvert\frac{1}{W-1} \sum_{i=0}^{W-2} I(x_{i+1} - x_i < 0) - .5\rvert$	
Line Length	Avg Peak Amplitude	Avg Valley Amplitude
$\ell = \sum_{i=1}^{W} -1\lvert x_i - x_{i-1}\rvert.$	$P_A = \log_{10}\left(\frac{1}{P} \sum_{i=0}^{P-1} x_{p(i)}^2\right)$	$V_A = \log_{10}\left(\frac{1}{V} \sum_{i=0}^{V-1} x_{v(i)}^2\right)$

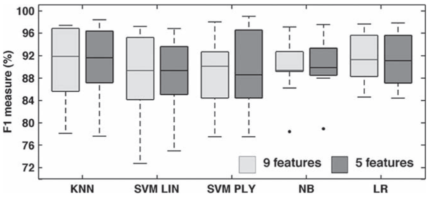


Fig. 3. Comparison of F1 measure for nine features and the reduced set of five features for each classifier on all ten patients' data. The reduction in features has little impact on F1 accuracy and variation.

TABLE II
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

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

metrics include F1 measure, onset-based sensitivity, window-based sensitivity, and number of false alarms per hour. Note that onset-based and window-based sensitivity are defined as the percentage of correctly identified seizure onsets and windows, respectively. A number of other works such as [9] and [10] utilize spectral analysis (frequency binning) for feature extraction similar to our earlier work [11]. Our proposed system is able to perform on par if not better than these other systems in terms of accuracy while being of much lower complexity. For example, on average, the low-complexity five-feature/ch + LR setup achieves a 100% onset sensitivity and 95% window sensitivity with 0.45 ± 0.56 false alarms/hr. In [10], the authors obtained a 100% onset sensitivity with 8 false alarms/hr for their digital setup and a 95% onset sensitivity with 30 false alarms/hr for their analog setup, and in [9], their spectral analysis + SVM-based system achieves, on average, a 93% onset sensitivity with 0.3 ± 0.7 false alarms/hr. In [12], a more recent implementation of [9], the authors obtained 100% onset sensitivity and 93.8% window sensitivity with 0.05 false alarms/hr.

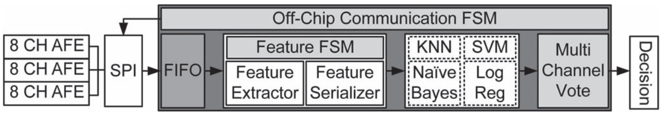


Fig. 4. Block diagram of the flexible seizure detection processor containing feature extraction, classifier, multichannel vote, and IO interface. Note that SPI and AFE blocks are not implemented.

III. FEATURE EXTRACTION AND RECONFIGURABLE CLASSIFIER HARDWARE

Fig. 4 shows the top-level block diagram of the proposed reconfigurable classifier and feature extraction processor. Although this is not the focus of this brief, we envision that, external to the processor, there will be three peripheral boards each consisting of an eight-channel analog front-end (AFE) IC, such as TIs ADS1299, which is a low-noise, programmable gain amplifier, 24-bit ADC with internal storage. The AFEs will be connected via a shared bus to a discrete SPI controller. Each channel is sampled at a rate of 256 Hz. A state machine implemented in the FPGA would use the SPI controller to read 1-s windows (256 samples/channel) from the AFEs and buffer the data in an internal first-in first-out (FIFO). Buffering a channel of samples allows the FPGA to receive the next window of samples while processing the previous one, reducing overall computation latency. The "Feature Extractor" block calculates and outputs the features in parallel to the "Feature Serializer". The serializer buffers the five features and outputs them serially to the classifier block. The "Feature FSM" block controls both the extractor and the serializer in addition to reading samples from the FIFO.

One of the four classifiers is selected at compile time using Verilog HDL parameters: KNN, SVM, NB, or LR. Only the selected classifier is instantiated to avoid using unneeded hardware resources. The classifier serially 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. This block can be extended to perform more complex decision-making algorithms.

A detailed architecture of the feature extraction process is shown in Fig. 5(a). Constants are eliminated from the formulas to reduce hardware complexity, and the modular approach allows for logic reuse between features. The block receives 256 samples serially from which it calculates five features in parallel. The peak/valley finder uses the 255 differences between consecutive samples to locate points where the slope changes, and the difference is greater than a predefined tolerance.

Details of the four classifiers used for the proposed processor are shown in Fig. 5(b)–(e). The SVM is highly parameterized: the kernel degree, number of features, fixed-point word width, and resolution are chosen at compile time. Features are input serially and are individually adjusted to have equal weight in the SVM. This result is passed to the pipelined dot-product, which has a number of stages equal to the number of features. The support vectors and other configurations are stored in read-only memory devices. For the linear kernel, the dot product is passed directly to the post-processing stage. For degrees greater than one, the dot product result goes through the polynomial stage prior to the post-processing stage. The sign of the final result is output as the classification. The configurable

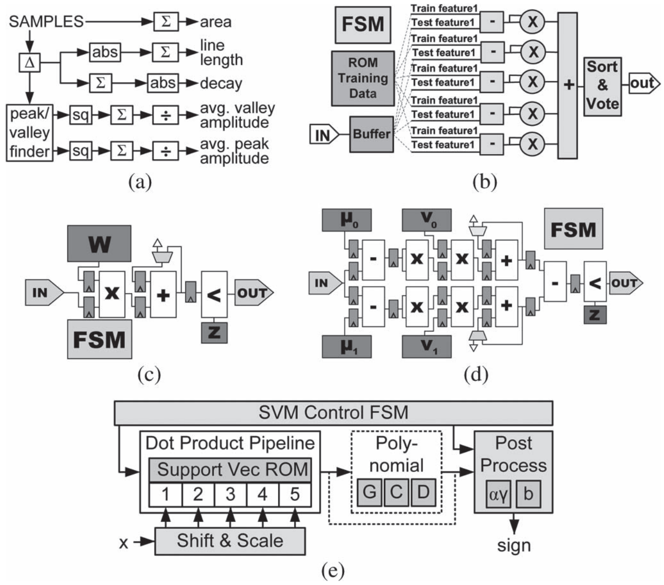


Fig. 5. Block diagrams of the Feature Extraction and reconfigurable classifiers including LR, NB, SVM, and KNN. The parameters for the classifier algorithms are explained in Section II. (a) Feature Extraction; (b) KNN; (c) LR; (d) NB; and (e) SVM.

KNN classifier can be exploited using different values for k (number of neighbors) and different sized training sets. Having received the sample test data, this classifier measures the sample data/training data distance and dynamically sorts the measured distances to find the nearest neighbors. NB and LR classifiers have very straightforward implementations that follow the derivation given in (7) and (5). NB consists of four small lookup tables to store the mean and variance for both class labels. LR contains just one lookup table to store the regression weights.

IV. FPGA Implementation and Results

A. Architecture Parallelism Optimization

A key consideration for the proposed processor architecture 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. Since there is a large half-second interval allotted for computation, the feature extraction and classification for each channel can be done serially. Furthermore, the classifiers' algorithms can be implemented serially; however, further analysis showed that alternative architectures provide a lower power solution for the SVM and KNN classifiers. Several architectures for the SVM's core dot product operation were explored: full serial, full parallel, and pipelined. All three implementations produced a decision within the time window, but the effective duty cycle power for the serial implementation was more than ten times that of the pipelined or parallel architectures. This interesting result is largely due to the distribution of resources. Due to the nature of the SVM, the memory requirements do not change based on the structure of the dot product. Since the FPGAs' dedicated block random access memory devices are a major contributor to the power, up to 50% in some cases, turning off the processor for the remainder of the window in which it is unused greatly reduces the effective dynamic power.

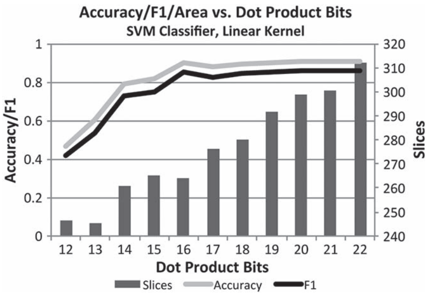


Fig. 6. Accuracy, F1 score, and area versus dot product word width for SVM classifier based on the post place and route results.

The pipelined architecture was chosen for the SVM dot product since it had similar performance to the parallel architecture but was significantly more configurable. A similar analysis for the KNN classifier illustrated that a parallel distance calculation yielded an appropriate balance between area, power, and latency. Due to the low complexity of the LR and NB algorithms, serial implementations were deemed optimal.

B. Fixed-Point Word Width Optimization

The feature extractor and each of the classifier blocks offer a significant number of parameters that can be set at compile time to tweak the performance of the design. For each block, a bit-accurate MATLAB model was used to simulate the hardware. The MATLAB models allow parameter configurations to be tested against real patient data to determine their impact on the accuracy and F1 score. In combination with the sweep of the same parameter configurations, the best tradeoff between performance and area could be identified. For example, Fig. 6 shows the impact of the number of bits used to represent the output of each dot product pipeline stage versus the accuracy, F1 score, and FPGA implementation area of the SVM classifier.

C. Resource and Power Results

Table II compares the performance of fully placed and routed classifiers with feature extraction. 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 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. The 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 to 0.004 s. Regardless, for the proposed system, the latencies are all relatively small for a wearable seizure detection problem. The more important metrics are the power, area, and accuracy. Figs. 7 and 8 illustrate FPGA resource utilization and dynamic energy for these classifiers, respectively. For dynamic power, a nominal frequency was selected for each classifier to meet the half-second interval.

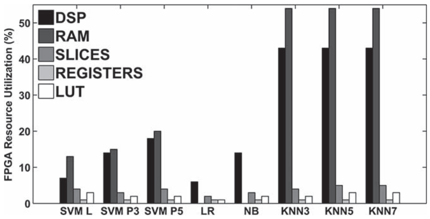


Fig. 7. FPGA resource breakdown for fully placed and routed classifiers: SVM linear, polynomial degree 3 and 5, NB, LR, and KNN with three, five, and seven neighbors.

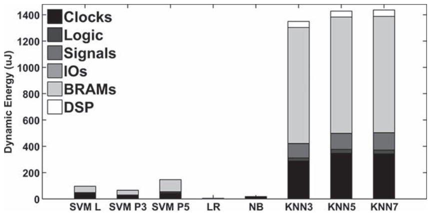
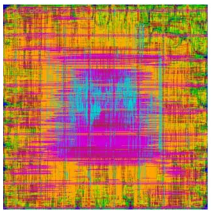


Fig. 8. Energy dissipation and breakdown for fully placed and routed classifiers: SVM linear, polynomial degree 3 and 5, NB, LR, and KNN with three, five, and seven neighbors. Note that the energy numbers are based only on dynamic power dissipation due to dominating leakage power of Virtex FPGA.



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. 9. Layout view and post-layout implementation 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 the half-second interval window.

V. ASIC Implementation and Results

The seizure detection processor with the LR classifier configuration is also synthesized and placed and routed in the 65-nm Taiwan Semiconductor Manufacturing Company, Ltd., CMOS technology. Fig. 9 shows the layout of the proposed seizure detection hardware (feature extraction + LR classifier) and post-layout results. The processor occupies 0.008 mm^2 and dissipates approximately 77 mW when running 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 chip dissipates approximately 37 nW (linearly scaled with frequency), which results in 19 nJ at 1 V to classify 256-sample input. 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 III compares the proposed 22-channel feature extraction + LR processor with multichannel seizure detection processors in [12] and [13]. The SoC power consumption for both previous works includes AFEs and ADCs.

TABLE III
Comparison With Previous Work on Seizure Detection Processors

Design	[13]	[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

VI. Conclusion

This brief has presented a low-power, flexible, and multichannel 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. Five low-complexity features were identified that could be derived from the EEG data to effectively classify windows of samples with an accuracy value of over 80% for data collected from ten patients. 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, 11 000× less memory, and consumes 26 960× less energy than a KNN-based processor. The ASIC implementation of the same combination in 65-nm CMOS demonstrated that the design fits in a 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. Viseh, M. Ghovanloo, and T. Mohsenin, “Toward an ultralow-power onboard processor for tongue drive system,” *IEEE Trans. Circuits Syst. II*, vol. 62, no. 2, pp. 174–178, Feb. 2015.

[2] A. Page and T. Mohsenin, “An efficient and reconfigurable FPGA and ASIC implementation of a spectral Doppler ultrasound imaging system,” in *Proc. IEEE 24th Int. Conf. ASAP*, Jun. 2013, pp. 198–202.

[3] J. Bisasky, H. Homayoun, F. Yazdani, and T. Mohsenin, “A 64-core platform for biomedical signal processing,” in *Proc. 14th ISQED*, Mar. 2013, pp. 368–372.

[4] S. Benbadis *et al.*, “Interrater reliability of EEG-video monitoring,” *Neurology*, vol. 73, no. 11, pp. 843–846, Sep. 2009.

[5] A. Goldberger *et al.*, “PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals,” *Circulation*, vol. 101, no. 23, pp. e215–e220, Jun. 2000.

[6] D. F. Wulsin, J. R. Gupta, R. Mani, J. A. Blanco, and B. Litt, “Modeling electroencephalography waveforms with semi-supervised deep belief nets: fast classification and anomaly measurement,” *J. Neural Eng.*, vol. 8, no. 3, Jun. 2011, Art. ID. 036015.

[7] J. Turner, A. Page, T. Mohsenin, and T. Oates, “Deep belief networks used on high resolution multichannel electroencephalography data for seizure detection,” in *Proc. AAAI Spring Symp. Big Data Becomes Pers., Knowl. Meaning*, Stanford, CA, USA, 2014.

[8] A. Page, J. Turner, T. Mohsenin, and T. Oates, “Comparing raw data and feature extraction for seizure detection with deep learning methods,” in *Proc. 27th Int. Conf. FLAIRS*, Pensacola Beach, FL, USA, 2014.

[9] N. Verma *et al.*, “A micro-power EEG acquisition soc with integrated feature extraction processor for a chronic seizure detection system,” *IEEE J. Solid-State Circuits*, vol. 45, no. 4, pp. 1625–1637, Apr. 2010.

[10] D. Sukumaran *et al.*, “A low-power, reconfigurable smart sensor system for EEG acquisition and classification,” in *Proc. IEEE APCCAS*, Dec. 2012, pp. 9–12.

[11] J. Chander, J. Bisasky, and T. Mohsenin, “Real-time multi-channel seizure detection and analysis hardware,” *IEEE Biocas Conf.*, pp. 41–44, Nov. 2011.

[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,” *in Proc. IEEE J. Solid-State Circuits*, vol. 48, no. 7, pp. 1625–1637, Jul. 2013.

[13] J. Yoo *et al.*, “An 8-channel scalable EEG acquisition soc with patient-specific seizure classification and recording processor,” *IEEE J. Solid-State Circuits*, vol. 48, no. 1, pp. 214–228, Jan. 2013.