AN EMBEDDED SEIZURE ONSET DETECTION SYSTEM

By

Alexander Lawrence Kindle

A Thesis

Submitted to the Faculty

Of the

WORCESTER POLYTECHNIC INSTITUTE

In partial fulfillment of the requirements for the

Degree of Master of Science

In

Robotics Engineering

September 2013

APPROVED:

Dr. Bryan McLaughlin, CS Draper Laboratory

Dr. Taskin Padir, Thesis Advisor

Dr. Michael Gennert, Committee Member

An Embedded Seizure Onset Detection System

Alexander Lawrence Kindle

September 12, 2013

Abstract

A combined hardware and software platform for ambulatory seizure onset detection is presented. The hardware is developed around commercial off-the-shelf components, featuring ADS1299 analog front ends for electroencephalography from Texas Instruments and a Broadcom ARM11 microcontroller for algorithm execution. The onset detection algorithm is a patient-specific support vector machine algorithm. It outperforms a state-of-the-art detector on a reference data set, with 100% sensitivity, 3.4 second average onset detection latency, and on average 1 false positive per 24 hours. The more comprehensive European Epilepsy Database is then evaluated, which highlights several real-world challenges for seizure onset detection, resulting in reduced average sensitivity of 93.5%, 5 second average onset detection latency, and 85.5% specificity. Algorithm enhancements to improve this reduced performance are proposed.

Contents

List of Figures

1 Introduction

1.1 Problem Statement

Epilepsy is the third most common neurological disorder, after Alzheimer disease and strokes. Approximately 1% of people worldwide have epilepsy [29, 28, 12]. Of the epileptic population, 30% have pharmacologically intractable epilepsy, which is epilepsy that is not responsive to two or more anti-epileptic drugs [12]. Existing treatments for pharmacologically intractable epilepsy include polypharmacy, ketogenic diet[31], deep-brain stimulation[4], and seizure focus removal. Benchmarking the performance of seizure mitigation therapies is challenging because it is either dependent on a patient's ability to record seizure frequency[20] and severity or it requires spending enough time in an epilepsy monitoring unit to establish both baseline and mitigated frequencies and severities.

1.2 Objective

There is a clear and present need for a minimally invasive seizure alert system. The electrographic onset of a seizure may precede the physiological onset by as much as a minute [27]. Clinical onset is the onset of debilitating or inhibiting clinical symptoms, such as convulsions, loss of bladder or bowel control, or impairment of consciousness [7]. A practical electrographic onset detector could alert a patient of an impending seizure, which could at least enable the patient to enter a more safe environment. Basic activities, such as climbing stairs, bathing, and crossing the street can present a safety risk to people with epilepsy [37]. An alert of an impending seizure could provide enough time for the patient to enter a safer state than their current state.

The goal is to develop an ambulatory seizure detection platform to aid in epilepsy management and improve quality of life [2]. We are simultaneously addressing both the hardware and the software challenges. We have developed an ambulatory electroencephalogram (EEG) monitoring system. We have also developed an embedded patient-specific support vector machine (SVM) seizure detection algorithm, based on the work in [29, 28]. This algorithm is configured for real-time execution in an embedded environment, such as on an ARM microprocessor. In our development, an ARMv6 architecture processor has been used.

1.3 Layout of Thesis

First, background information on epilepsy, EEG, seizure detection, and SVM classification will be presented. In part 3, the software design is presented, and the design choices are discussed. In Part 4, the hardware design is presented and discussed. In part 5, results against a public database are derived and compared to an industry-standard detector. Then, results against a new, larger database are derived and discussed in detail. In part 6, we make some conclusions about the system and proposals for future work.

2 Background

In this section, a brief overview of seizure detection algorithms is presented. A variety of possible approaches to feature extraction and classification are discussed. The approach used in this thesis is analyzed. Computational implications of approaches in general and in the proposed approach specifically are highlighted.

2.1 Seizure Detection Algorithms

Seizure detection algorithms as a whole are a class of algorithms whose objective is to detect seizure onset. This is distinct from seizure prediction. Seizure detection can only occur after the start of seizure activity, while seizure prediction aims to predict when a seizure will happen [24]. Seizure detection determines that a seizure is happening. Typical seizure onset detection is done with electroencephalographic signals, however work has been done to augment or replace this data with other sensor sources, such as electrocardiographic or actigraphic data [8, 18, 26, 27].

The generalized problem behind seizure detection, regardless of data source, is binary classification: given the data currently available, is a seizure occurring? This classification may be performed on just a current slice of data, or it may be performed in the context of some quantity of historical data. The classification cannot take into account future data for obvious reasons.

2.1.1 EEG for Seizure Detection

EEG, or electroencephalography, is the process of measuring the electrical signals that are indicative of brain activity. There are three main groups of EEG signal measurements - intracranial EEG, where signals are measured on the surface of the brain; scalp EEG, where signals are measured on the

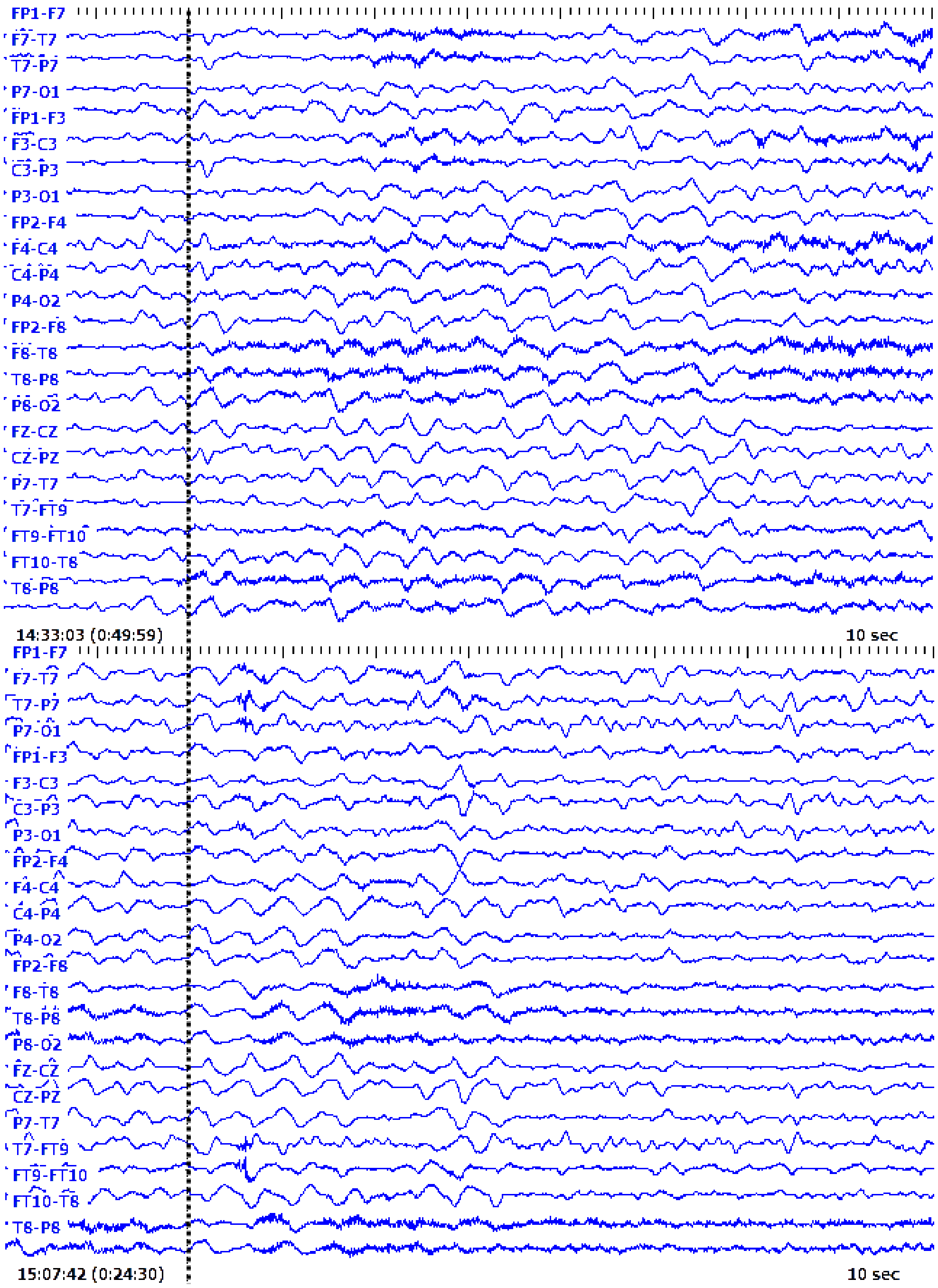


Figure 1: Two seizures from patient 1 of the CHBMIT database [15]. The electrographic seizure onset is indicated by the vertical dashed black line.

surface of the scalp; and single-unit recordings, which measure the electrophysiological responses of a single neuron. While a variety of brain activities can be detected through EEG, the focus of this work is on detecting the presence of epileptic seizures.

Epileptic seizures manifest in EEG as a sudden shift in the power spectral density [29, 28]. The appearance of epileptic seizures in EEG is very patient-dependent in terms of individual signal morphology as well as distribution across the patient's anatomy and consequently across multiple EEG channels. However, for a given patient, epilepsy signals are very consistent across seizures, as shown in figure 1. A patient-specific detector trained on an individual's seizure history is an approach that been successfully demonstrated in [29, 28].

For the purposes of a noninvasive ambulatory monitoring device, scalp EEG is the least invasive signal set. While intracranial signals demonstrate high degrees of localization - specific signals are very limited in where they can be measured on the geography of the brain - scalp EEG is much more generalized. Similarly, intracranial EEG has high-frequency components, while scalp EEG does not. The skull's electrical impedance functions as a low-pass filter, which limits both the high frequency components of EEG and the high locality of the signals. In clinical applications, this limits the ability of determining which region of the brain a signal originates from, but for the our applications, it reduces the number of electrodes needed to fully cover the anatomical signal space.

In terms of bandwidth, the signals of interest to seizure detection appear in the 3-25 Hz band. The signal potentials at the scalp of these signals are very small - typically less than 20 microvolts.

2.1.2 State of the Art Detection Algorithms

Current state of the art seizure detection algorithms can be separated into two groups - patient-specific and patient-nonspecific. Patient-specific detectors, such as the one implemented in this thesis, generally have superior detection capabilities than nonspecific detectors [29, 28], but have the drawback of requiring a training dataset of the patient's EEG, including seizure EEG. This is most significantly a problem in patients whose seizures are infrequent, particularly because provoked seizures, such as those triggered by photo stimulation or hyperventilation, may by physiologically different from a patient's spontaneous epileptic seizures [36]. Patient-nonspecific algorithms do not require any information about the patient under test, but sacrifice detector performance.

In general, EEG-based epileptic seizure detection is composed of the following process: data acquisition, feature extraction, classification, and results. The variables in this system are the feature extraction technique and the classification method. A multitude of possible feature extraction techniques have been explored, including Fourier transform based, frequency domain, time-frequency based, wavelet based, and a wide variety of time domain techniques have all been explored to varying degrees of success [36, 34, 33, 3, 32]. Similarly, a wide variety of classifiers have been evaluated, including threshold, rules-based decisions, linear classifiers, artificial neural networks, support vector machines, nearest-neighbor, quadratic analysis, logistic regression, naive Bayes classifiers, decision trees, Gaussian mixture models, and adaptive neurofuzzy inference systems [36, 32, 34, 33].

More generically, the desired goal is to take a set of input signals and group them into two classifications. In this case, the classifications are epileptic EEG and non-epileptic EEG. Perfect feature extraction will result in highly separable features; epileptic EEG features will be very distinct from non-epileptic features. From these highly separable features, a binary classifier can then assign a label to unknown features: epileptic or non-epileptic. This is the foundation of the epileptic seizure onset detector. Unfortunately, epileptic and non-epileptic EEG tend to be less than perfectly separable. EEG artifacts are a common source of separability impairment. These artifacts may either be a result of physiologic processes, such as muscle or ocular artifacts, or a result of EEG recording procedures, such as electrode interface artifacts [35, 10, 19, 16, 39]. These artifacts may contaminate non-epileptic EEG with signals that may resemble epileptic EEG. Similarly, many patients with epilepsy will exhibit inter-ictal spike, wave, and spike-wave complex signals, which are brief EEG signals that resemble seizure onset EEG but are not indicative of seizure onset [36, 28, 29]. Being able to properly classify artifacts and inter-ictal EEG is an important challenge for seizure detection algorithms.

It is unlikely that the features extracted from a patient's epileptic and non-epileptic EEG will be perfectly separable. As a result, the classifier used must be able to tolerate misclassification in some way. A single misclassified feature doesn't necessarily result in a false positive, though; a practical defense against such a feature, which may be an inter-ictal non-epileptic event, is to require some quorum of positive classifications to declare a seizure onset. Such a quorum, however, represents a tradeoff of detection latency for improved detector specificity, as additional time needs to pass to gain the additional features used to vote for a seizure onset.

Lastly, computational complexity must be accounted for. A feature extraction technique that results in perfectly separable features, or a classifier that results in perfect classification off of a minimum amount of data, is useless as a wearable therapy if it is too computationally complex to perform on embedded hardware in the performance range of a smartphone. This is further bounded in our case by the need of the system to be able to perform detections online, meaning the feature extraction plus classification must take less time than the time it takes to capture new data, and by the need of the platform to be ambulatory, which puts a limit on how much electrical power (and, consequentially, computational power) can be employed. Luckily, the capabilities of embedded technology have skyrocketed in recent years, partially thanks to the smartphone boom, which has driven a huge demand for faster, lower-power embedded processors and smaller, more energy-dense battery sources.

Here we develop an ambulatory real-time implementation of a detector using filterbank-derived energy-based features and Gaussian radial basis function support vector machine classifiers, which were shown to be a highly successful method of seizure detection in [29, 28]. This algorithm is a patient-specific detector. The primary disadvantage of patient-specific detectors, as mentioned earlier, is the potential difficulty in collecting seizure EEG from a patient. With our ambulatory solution, which includes EEG recording, collecting seizure EEG from patients with rare spontaneous seizures becomes a much simpler task than the traditional epilepsy monitoring unit approach. As a result, our detection system will be able to enjoy the much higher performance of a patient-specific detector.

2.2 Support Vector Machines for Seizure Detection

Support vector machines, or SVMs, are a class of supervised learning binary classification algorithms originally developed by [5, 9]. In the most basic formulation, the SVM is used to find a maximally separating linear decision hyperplane between two groups of data, such as in the two-dimensional example in figure 2.

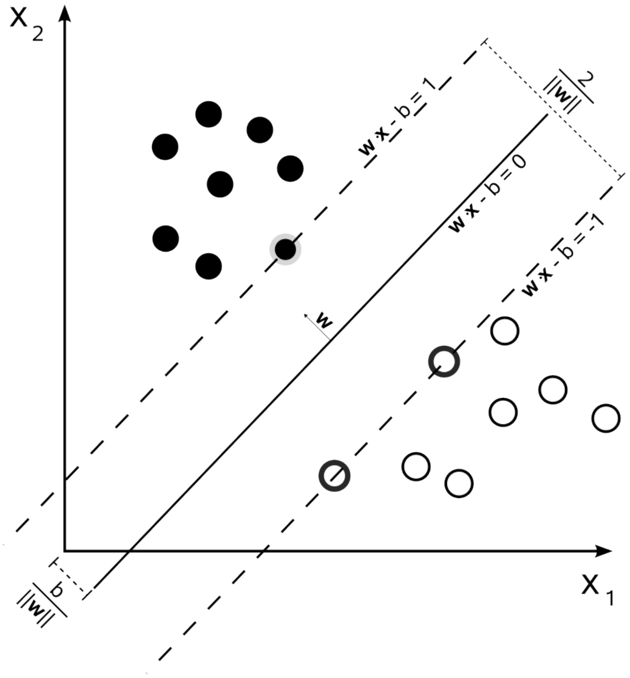


Figure 2: A maximally-separating hyperplane between two classes of data. The points on the dashed lines are the support vectors. Cyc / Wikimedia Commons / Public Domain

2.2.1 Formulation

The SVM as originally described in [5] provided only for hard margins; it does not function on groups of data where no hyperplane can cleanly separate two groups of data. In [9], a soft margin SVM is proposed, which allows for misclassification of the training set. In this form, maximize (in α_i):

$$\tilde{L}(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j) \tag{1}$$

with $0 \leq \alpha_i \leq C$ and $\sum_{i=1}^{n} \alpha_i y_i = 0$ where α are Lagrange multipliers, y_i are labels in $\{-1, 1\}$ corresponding to seizure and nonseizure training data x_i, and C is the cost parameter of misclassification. W, which can be seen in figure 2, can be computed from these terms:

$$\mathbf{w} = \sum_{i} \alpha_i y_i \mathbf{x}_i \tag{2}$$

2.2.2 Kernels

The original SVM formulation in [38] was strictly a linear classifier. In [5], the modification for efficient nonlinear decision boundaries was introduced with a kernel function, such as the nonlinear boundary in figure 3.

The purpose of the kernel is to serve as a function to project a data set that is not linearly separable in its current space into a higher-dimensional space where it is linearly separable. A contrived example of this can be seen in figure 4. A two-dimensional data set, which is not linearly separable in the two-dimensional problem space, has a kernel function applied which projects the data into a three-dimensional kernel space, where the data is linearly separable.

While there are a variety of kernels, such as a polynomial kernel or hyperbolic tangent kernel, the kernel successfully used for seizure detection is

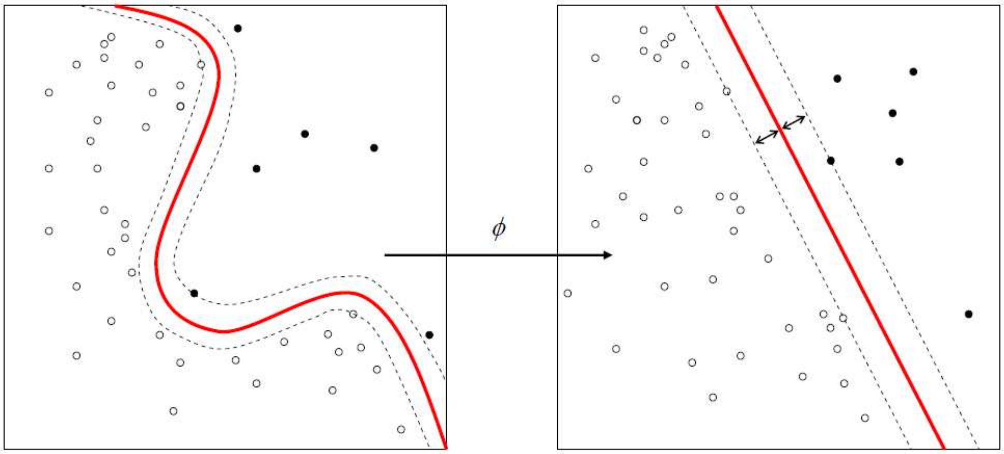
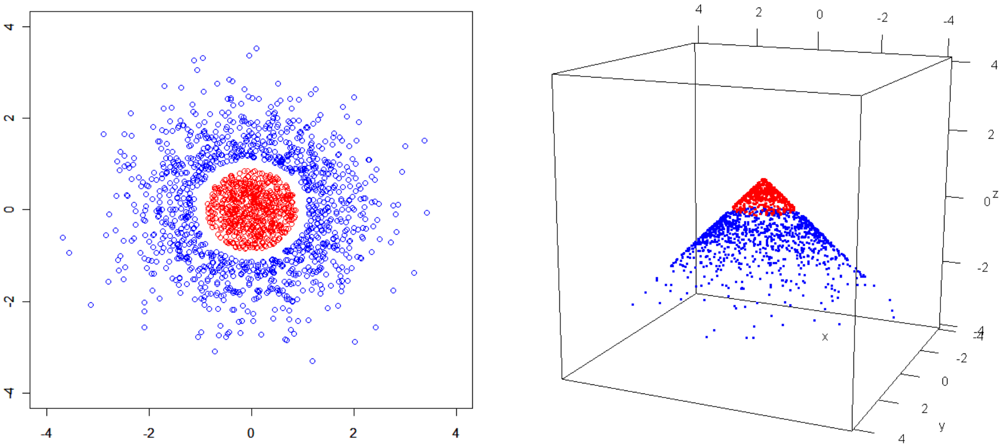


Figure 3: Kernel trick phi allows for a nonlinear decision boundary. Alisneaky / Wikimedia Commons / Public Domain



(a)

(b)

Figure 4: A binary dataset in (a) is transformed from linearly inseparable in the problem space to linearly separable in the kernel space in (b). StompChicken / StackExchange / cc by-sa

the common Gaussian radial basis function kernel [29, 28]:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2) \tag{3}$$

where γ is the kernel parameter, where $\mathbf{x}_i$ and$\mathbf{x}_j$ are the input vectors. This kernel, when plugged into (1), gives the formulation for a Gaussian radial basis function support vector machine like the one used in this detector.

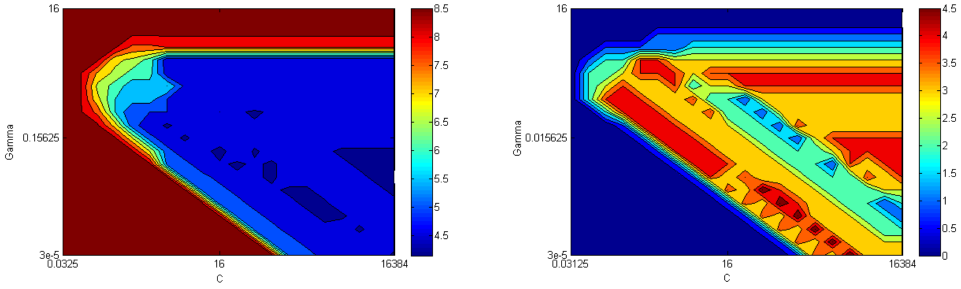
Finally, given some trained SVM classifier, a classification can be described as:

$$f(\mathbf{p}) = sign(\sum_{i=1}^{n} y_i \alpha_i k(\mathbf{p}, \mathbf{x}_i)) \tag{4}$$

Where $\mathbf{p}$ is the feature vector being classified, $\mathbf{x}$ is the matrix of training vectors, y is the vector of training vector labels, and α is the Lagrange multiplier.

2.2.3 Optimization

The important detail in all this math is that there are two sets of optimizations that must be performed: the quadratic optimization of the SVM, which determines which features are support vectors, and the optimization of gamma and C, which will tune the detector's performance. The quadratic optimization problem is heuristically solved by LIBSVM [6], but the optimization of gamma and C is not. Adjustments to gamma and C will impact the detector's sensitivity, which contributes to the percentage of seizures correctly identified, and specificity, which contributes to false positive rate. There is no one-size-fits-all optimization that can be applied to these parameters, as an increase in sensitivity can result in a decrease in specificity. In some cases, maximum sensitivity with little regard for specificity may be a desired outcome; in other cases, a more balanced approach may be desirable. As a result,



(a) Average detection latency in seconds from seizure onset.

(b) Total false positive detections.

Figure 5: γ vs C; heatmaps of latency and false positive rate. Neighboring γ, C combinations tend to have similar performance characteristics. A best-caseγ, C is as low as possible (dark blue) on each map.

some fitness function $f(\gamma, C)$ does not exist and cannot be minimized. In order to facilitate parameter selection, then, an operator is needed at training time. A grid search of $C \in \{2^{-5}, 2^{-4}, ..., 2^{14}, 2^{15}\}$ $\gamma \in \{2^{-15}, 2^{-14}, ..., 2^{2}, 2^{3}\}$, as recommended in [21], is used to facilitate this process. In figures 5a and 5b, heatmap visualizations of this process can be seen. For each pair of $\{C, \gamma\}$, an SVM is trained and cross-validated; the results of the process are stored in a matrix. These results can then be used to select parameters or visualize detector performance across different parameter combinations.

2.2.4 Features

The features used in the Gaussian RBF SVM are composed of log-sum-absolute energies of frequency bins of the source EEG. The input data in this case is $x_{m,n}$, a matrix where the m dimension is the number of EEG channels used and the n dimension is the number of samples in the feature. First, $x_{m,n}$ is decimated into $x'_{m,n}$. Then, $x'_{m,n}$ is passed through a filter bank, where it is divided into some number of frequency bins, $x'_{m,n,o}$, which is a matrix of dimensions $\{channels, samples, bins\}$. The feature matrix is calculated from this as:

$$y_{m,n} = \ln(\sum_{n} \|x_{m,n,o}\|) \tag{5}$$

Concatenating the columns in this feature matrix into a vector produces the feature vector, Y_p, where the p dimension is the number of features (where the number of features is $channels * bins$). In the context of the SVM formulation in (1):

$$\mathbf{x}_i \in \{Y_p, Y_{p+1}, Y_{p+2}, ...\} \tag{6}$$

Additionally, before these features should be used by the SVM classifier, they should be scaled. While it is not frequently discussed in theoretical literature surrounding the SVM, scaling is very important in practice; un-scaled feature vectors may result in SVMs where subsets of the feature vector dominate the solution only for numerical reasons. In order to scale the feature vectors:

$$\bar{M} = \{\max(\mathbf{x}_{i,1}), \max(\mathbf{x}_{i,2}), ..., \max(\mathbf{x}_{i,p})\} \tag{7}$$

$$\bar{m} = \{\min(\mathbf{x}_{i,1}), \min(\mathbf{x}_{i,2}), ..., \min(\mathbf{x}_{i,p})\} \tag{8}$$

$$Y_p = \frac{Y_p - \bar{m}}{\bar{M} - \bar{m}} \tag{9}$$

Which will scale the values in Y_p to the range [0..1].

2.2.5 Computational Complexity

Two components of the above math are interesting for the computational complexity of real-time seizure detection: classification decisions and feature extraction. The computational complexity of SVM training is an interesting subject thoroughly explored elsewhere, but in the context of online seizure classification it is not particularly relevant because training can be performed offline, with stored training results loaded at runtime by the classifier.

The SVM classification function is a sum of products of vectors, where one of the vectors is the output of the kernel function. The kernel function is an exponentiation of a multiplication of a constant and a squared sum. The SVM classification function is linear with respect to the number of support vectors, and it is linear with respect to the dimensions of the feature vectors.

The feature extraction consists of several FFTs and vector multiplications. As a result, it is quasilinear with respect to the number of EEG channels or samples, because an increase in either of these results in an increase in the length of the FFTs performed, and FFTs are $O(n \log n)$. An increase in the number of bins results in a linear increase in the number of FFTs performed, but does not change the nature of the FFTs themselves, so it is linear.

The support vectors are the dominant factor in terms of the amount of data that must be stored in order to perform classifications. It is difficult to estimate the number of support vectors needed for classification due to the nature of the SVM optimization algorithm; however, it is safe to assume as many as a few thousand may be used. While this did not prove problematic in our implementation, there is ongoing work in SVM data reduction and approximation, which can reduce the number of support vectors needed significantly, although with some classification performance hit [23, 25]. In addition to the support vectors themselves, the scaling vectors $\bar{M}$ and $\bar{m}$ must also be stored, along with the constant γ.

3 Software

In this section, the seizure detection algorithm implementation is described in detail. The algorithm is a support vector machine (SVM) classifier, which is a supervised learning binary classification algorithm. Our implementation is based on [29, 28]. In this algorithm, the EEG data is processed into frequency binned log-sum-absolute energies. Seizure and nonseizure data in this feature space are not linearly separable; the feature space has no linear boundary between seizure and nonseizure features. The traditional support vector machine formulation assumes a linearly separable problem set, but through the use of a kernel trick the feature vectors may be projected into a higher-dimensional space in which they are linearly separable. This higher-dimensional linear separation is equivalent to a nonlinear separation in the original feature space. In this application, a Gaussian Radial Basis Function kernel has been shown to be effective [29, 28]. The features used here are constructed from three hertz-wide energy bins. Each signal in the data set is filtered into eight frequency bins. The energy in each of these bins is calculated and concatenated into a vector. The resulting feature vector has (number of signals) * (number of frequency bins). EEG patients will typically have around 20 signals with 8 frequency bins, for around 160 features. This number will vary from patient to patient because it is dependent on the EEG configuration.

The algorithm is evaluated with the CHB-MIT data set [15], which was also used in [29, 28]. This makes it easier to benchmark our algorithm's performance against a reference implementation. The algorithm is then evaluated against the European Epilepsy Database [22].

3.1 Feature Extraction

The feature extraction stage was developed as a standalone application in C++ with the intent that the resulting feature data could be processed in a variety of ways. Currently, this feature data may be fed into: the training and cross-validation software, the embedded detection software, and MATLAB through a custom plugin. C++ was the language of choice for almost all of the software in this thesis because most of the software needs to be able to run in an embedded system. In such an environment, something like the Java Virtual Machine or an operating system represents a huge overhead in both computational resources and power. Our C++ software can much more easily be compiled for bare-metal operation, which will help produce a more embedded system than may otherwise be possible.

Feature extraction is a many-layered process. At the outermost layer, a set of EDF (European Data Format - the standard for storing EEG data) files are processed. For each EDF file, the signals are windowed. For each window, signal data is processed. For each signal chunk, a set of features are generated and saved.

3.1.1 Patient Data Processing

The input data set for a given patient consists of a group of EDF or EDF+ files. These files tend to individually contain around an hour of recorded EEG data. Additionally, a metadata file will contain annotations about file start and end times, seizure start and end times, and EEG channel configurations. The first layer of the feature extraction application starts with this metadata file. The application reads this file to determine the names of the input files to use, as well as any seizure occurrences within those files. The program opens an EDF file and passes that file and seizure timestamps on to the next stage.

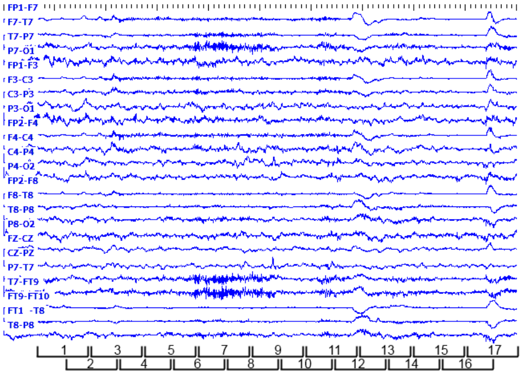


Figure 6: Example of windowing EEG data for feature extraction. Data segment from CHBMIT Database[15]

3.1.2 Individual EDF Processing

The EDF file from the first stage is opened and data parsing begins. The number of samples is read, as well as the number of signals. Data structures are created based on these parameters. The matrix of feature vectors for this file is allocated as a three-dimensional array, whose dimensions are {(total number of windows to parse file),(signal number),(frequency bin)}. The feature extractor stage is then prepared with some metadata and configuration parameters, such as sampling frequency of the data and window length. Now, the application begins sliding the window over the EDF file. This generates blocks of signals that are two seconds long, with one second of overlap between blocks. This windowing process can be seen in figure 6. Each of these windows is sequentially processed by the feature extractor.

3.1.3 Feature Extractor

Finally, at the bottom layer, the signal processing takes place. This section makes heavy use of the FFTW3 library [14]. First, the data is transformed into the frequency domain with an FFT. A decimation filter is applied, and the data is transformed back to the time domain, where decimation is completed. This decimated data is then transformed back to the frequency domain, where a comb filter is applied. The data is converted back to the time domain, and log-sum-absolute energies are calculated for each frequency bin. These energies are returned to the higher-level code, where they are assembled into a feature vector.

3.1.4 Output

Feature vectors are serialized to disk at the highest layer. This serialization uses the Google Protocol Buffers[17] format and library to store the data in an easily-digested widely-supported binary format. The Protocol Buffers project has native libraries for C++ and Java, with third party ports to dozens of other languages and platforms. As a result, the feature vectors generated here are highly portable with minimal custom storage code.

3.2 MATLAB Inspection and Validation

These features can now go several places. First, let's consider the process of validating our generated features in MATLAB. This process is very simple, thanks to the power of Protocol Buffers[17]. While the original binary data was serialized in C++, there exist libraries in nearly every major language for serializing and deserializing the Protocol Buffer binary message format. While there is a third-party version of the library for use in MATLAB, it was, at the time of use, not up to date with the current Protocol Buffers library version, so it did not work with our data. Luckily, Java/MATLAB interoperability is extremely simple, and Java is one of the officially-supported languages of the Protocol Buffers project. A simple features-to-MATLAB loader was built in Java around this library, and easily allows for feature inspection and validation in MATLAB.

3.3 Offline Training

The most significant use of these pre-calculated features is offline SVM training. Like the feature extraction stage, this stage was written in C++. While the offline training should not take place on the embedded platform, the entire cross-validation portion of the training procedure is identical to the online detection process. Additionally, cross-platform cross-validation helps validate program functionality. As a result, in the interest of code reuse and library compatibility, the training software was developed in C++, even though it will generally never run on the embedded platform. Several libraries are used in the training software to facilitate the process. Google's Protocol Buffers are used to read the precomputed feature data from disk [17]. LIBSVM is used for SVM training and calculations [6].

The training process starts by supplying a file of pre-calculated features. The data is loaded and separated into seizure and nonseizure chunks. Next, the SVM parameter search space is generated. By default, the system performs a grid search of C values [2^-5, 2^-4, 2^-3...2^20] and gamma values [2^-15, 2^-14, 2^-13...2^10], which is a larger version of the range from the suggested range in the LIBSVM documentation [21]. In LIBSVM, C controls the cost of misclassified training points, while gamma influences the width of the Gaussian radial basis function kernel. For each parameter combination, SVMs are trained using leave-one-out cross-validation. In this validation scheme, the SVM is trained on all all seizure data except for the seizure under cross-validation and a subset of the nonseizure data. The nonseizure data subset was constructed by using one out of A nonseizure feature vectors. This value A was fixed at 30 for the CHBMIT database, but was varied on the European Epilepsy Database [22]. Not all of the nonseizure data is used in training, because there are several orders of magnitude more nonseizure data than seizure data. Supplying all of the nonseizure data could result in over-training or mis-training, and would take significantly longer to process.

Before this data is used for training, it must be normalized. The range of values at each index of the feature vector should be on the range of [0...1] in order to prevent indices with numerically prominent data from overpowering numerically smaller indices. This scaling takes the form of

$$scaledValue = \frac{value - minimum}{maximum - minimum} \tag{10}$$

Once this SVM is trained, the remaining block containing the seizure under test and up to one hour of non-seizure data is run through the detector. The known label and the SVM-predicted label are both logged to a results file. This cross-validation scheme is repeated for each seizure in the data set, and the entire training process is repeated for every possible combination of C and gamma. After these results are analyzed, a human operator may select a (C, gamma) pair that produces optimal results. There are currently no facilities for automatic parameter selection, because the desired performance of the detector is highly orthogonal - high sensitivity and lower latency generally imply a higher false positive rate, so there is rarely a (C, gamma) that can be considered "best" for any one patient. Should such a fitness function exist, it would be reasonably straightforward to implement a greedy search for optimality. SVM optimization is NP-hard, so it is not currently possible to optimally solve for optimal parameters. In any case, if the user supplies a desired (C, gamma) then the corresponding SVM will be saved to disk for later use, such as in online detection.

3.4 Online Detection

The online detection stage is closely related to the feature extraction and offline training processes. This process flow is laid out in figure 7. Currently, the online functionality is simulated by reading the data from EDF files on disk, due to an absence of sources of live EEG containing seizures. Hook-

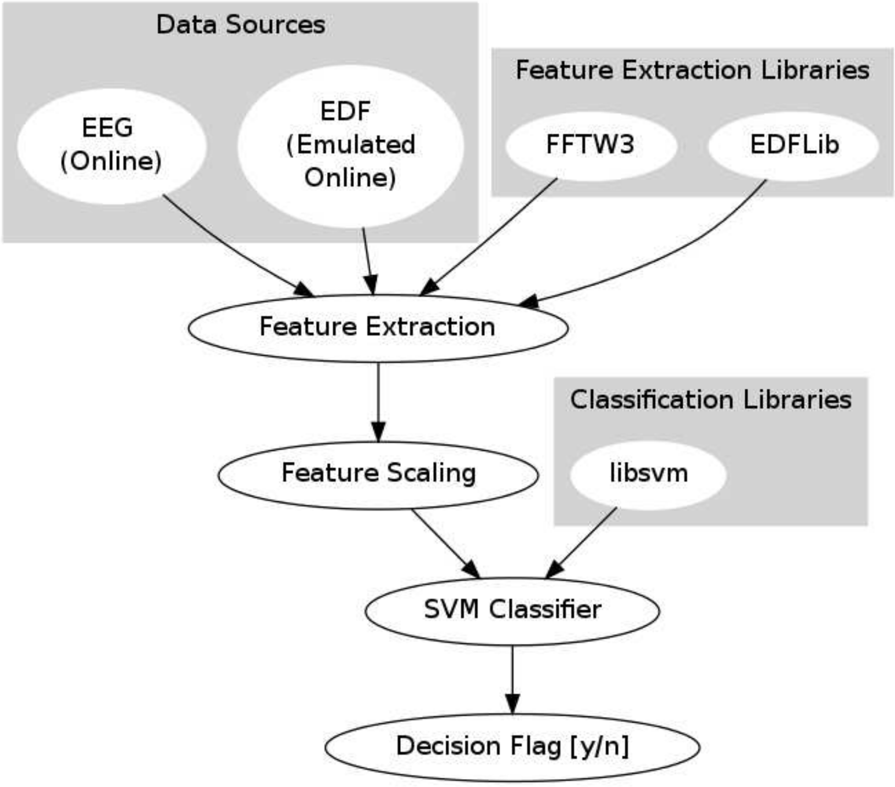


Figure 7: Embedded algorithm execution flow

ing the software up to an incoming data stream from the analog front end described in section 4 is a trivial adjustment.

First, features are extracted from the fresh data. This functions exactly as described in the feature extraction section. In addition to serializing the features to disk, however, the feature vectors are also immediately scaled and classified. Storing feature vectors from online detection enables retrospective performance evaluation and manual system adjustments if needed.

Classification is performed in a manner similar to the cross-validation stage of the SVM training procedure. At startup, a SVM is loaded from disk. As features are extracted, they are classified by this saved classifier. SVM decisions are logged in the same format as the cross-validator's output in conjunction with live n out of m seizure declaration. This provides a fully online seizure declaration system with the ability to perform retrospective analysis to potentially improve performance.

3.5 Onset Declaration

At this time, both the offline trainer and the online detector simply output SVM classifier decisions. The SVM classifier outputs a 1 for feature vectors classified as seizure and a -1 for feature vectors classified as nonseizure. These individual flags could be used as declarations, but doing so leads to a high false positive rate (as seen in figure 14 and figure 18 in section 5.1.2). Instead, a results processor was written, which evaluates these classifications separately from the classification generation. This separation of duties enables a wide array of voting schemes to be evaluated without rerunning the entire classification, which would take several hours per patient. The results processor fundamentally works by using an n out of m voting scheme to declare seizure onsets. As it reads in classifications, the m most recent classifications are stored. If the number of positive classifications is greater than or equal to n, a seizure onset is declared. This process repeats for each classifier output.

These onset declarations could be compared directly against the expected label (seizure, nonseizure) at the time of declaration and treated as either a true positive or false positive if it matches or not. However, that approach has some significant drawbacks. For the purposes of training, each seizure is limited to a maximum of ten seconds of positive seizure onset labeling. Additionally, these labels have been assigned by an epileptologist in a manner that is not strictly quantitative - there is no consensus or objective definition of what constitutes a seizure [13]. The labeled onsets assigned by two different epileptologists can differ by multiple seconds. As a result, the labels used for training may lag the true electrographic onset by multiple seconds. The classifier may then trigger a detection closer to the true electrographic onset than the labels are. If the detection is strong enough to be a declaration, and that declaration is compared to a potentially mislabeled negative, then a true positive detection may unintentionally be mistreated as a false positive.

A mechanism for allowing detections in a window around the labeled onset is needed. This is handled by logging the past j declarations, which are compared against the current label value. If any of the past j declarations are positive when a series of positive labels get processed, the declaration is stored as a true positive. If a positive declaration's age exceeds j, a false positive is stored.

Handling high latency detections is processed in a similar manner. The time since the last positive label is stored. If a positive declaration is raised while that time is less than some time k, a true positive is recorded. If time greater than k has elapsed, then a false positive is recorded.

Finally, there is the issue that both true positive and strong false positive events are likely to have a large number of positive classifications associated with them. Raising several onset declarations in a short period of time does not make sense, whether it is during a seizure or not. To simplify this, onset declarations are treated with a cooldown timer p. Once an onset is declared (whether it is a true positive or a false positive), a timer is initiated; during that timer window, no additional declarations will be raised, and all declaration history windows will be initialized to empty once the timer has expired.

4 Hardware

In this section, the various hardware developed and used is described in detail. There are two main embedded systems: the ambulatory recording platform and the embedded seizure onset detecting hardware. These platforms were developed and used in parallel; a future version combining the two into a unified EEG recording and seizure onset detecting device is planned.

4.1 EEG Hardware

A highly integrated battery operated EEG recording platform was developed over several generations. The first generation of devices were focused on exploring wirelessly powered minimally invasive implantable EEG systems. The goal form factor for these devices was a strip device that could be inserted beneath the scalp through a small incision, combined with an external wireless supply and data logging device.

4.1.1 First Generation

The first generation of EEG recording platform prototypes focused on the TI ADS1298 analog front end. This family of devices all featured a single ADS1298 in a variety of configurations. For more information about this generation of hardware, see [11].

ADS1298 Evaluation Board The first board developed was a custom evaluation board for the ADS1298. This board featured an ADS1298, an MSP430, a serial port, power supply circuitry, and many breakout connections and test points. The purpose of this board was to evaluate the feasibility of using the ADS1298 to record biosignals. For the sake of rapid testing, the primary focus was on ECG signals because they are very easy for untrained individuals to identify. This project primarily focused on exploring the problem space and tackling the challenges of working with the ADS1298 family of devices and their data results.

Wireless Board A related system was a wireless EEG recording platform, which was developed to begin experimenting with wireless power systems. The wireless board featured an ADS1298, an MSP430F2272, an antenna, and some related wireless power hardware. The receiver board featured a TI TRF7960A, an Actel Igloo Nano AGL250 FPGA, an MSP4302272, a microSD card, and power supply systems. The ADS board was wirelessly powered by the TRF7960A on the receiver board, and would load modulate the power link to transmit data. The FPGA demodulated the data and managed the wireless link. The MSP430 managed the FPGA, TRF7960A, and serialized the data to the microSD card in a custom format. This project primarily focused on improving the performance and reliability of the wireless power and data link. Preliminary EEG performance analysis was also performed.

4.1.2 Second Generation

The second generation of devices currently features a complete prototype logging platform based on the TI ADS1299. The device is pictured in figure 9; a configuration on a test patient with wet electrodes can be seen in figure 10. The ADS1299 is a higher power, lower noise version of the ADS1298. It is pin-compatible with the ADS1298 in certain power supply configurations; the ADS1298 operates off of a 3 volt AVCC while the ADS1299 requires a 5 volt AVCC. The current second generation platform has three ADS1299s, an MSP430F5310, an Actel Igloo FPGA, and supporting components. Almost all digital signals on the board are routed through the FPGA for maximum flexibility. A block diagram of this design can be seen in figure 8. The original

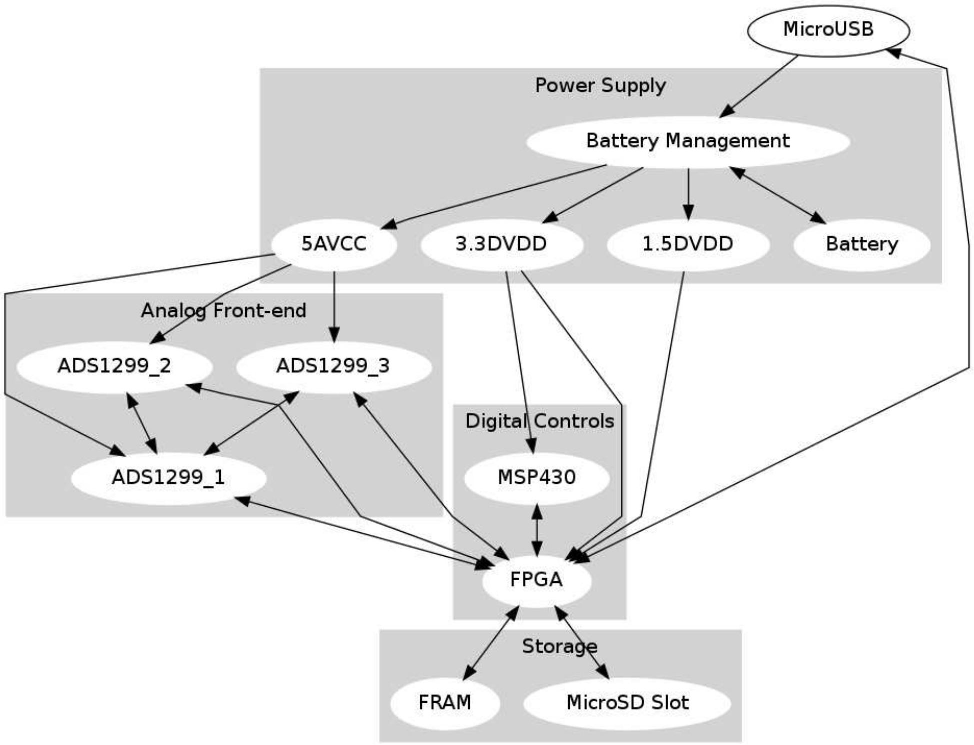


Figure 8: Block Diagram of Second Generation EEG Platform

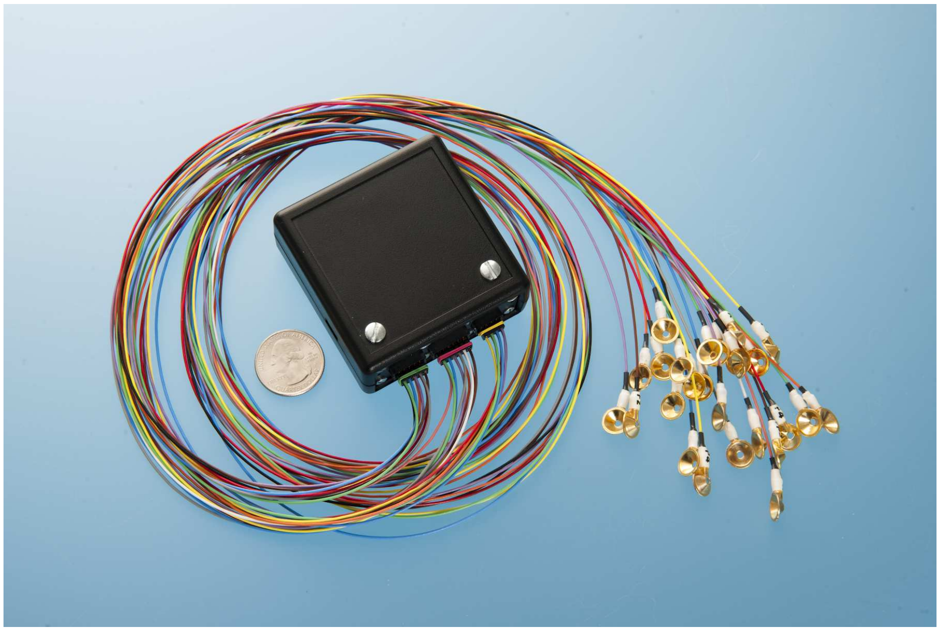


Figure 9: Second generation hardware platform

design intent was to embed a detection algorithm in the FPGA, with the microcontroller providing the majority of platform management. However, during the firmware design phase, these objectives changed significantly. In the final firmware design, the FPGA manages the entire system. The FPGA contains a state machine to write to the SD card in the same custom format as the wireless board. It also contains a state machine for initializing all three ADS1299s. The ADS1299 features an internal reference generator and internal clock generator. In this application, it is desirable to have all three ADS1299s using the same reference voltage and clock for optimal results, so there is a master ADS1299 chip and two slave ADS1299 chips. The master is configured to output its clock and reference, while the slaves are configured to operate off of an external clock and reference. Similarly, the ADS1299 features a bias generator. This circuit is used to improve common-mode rejection of the system by driving the body with the average of all input

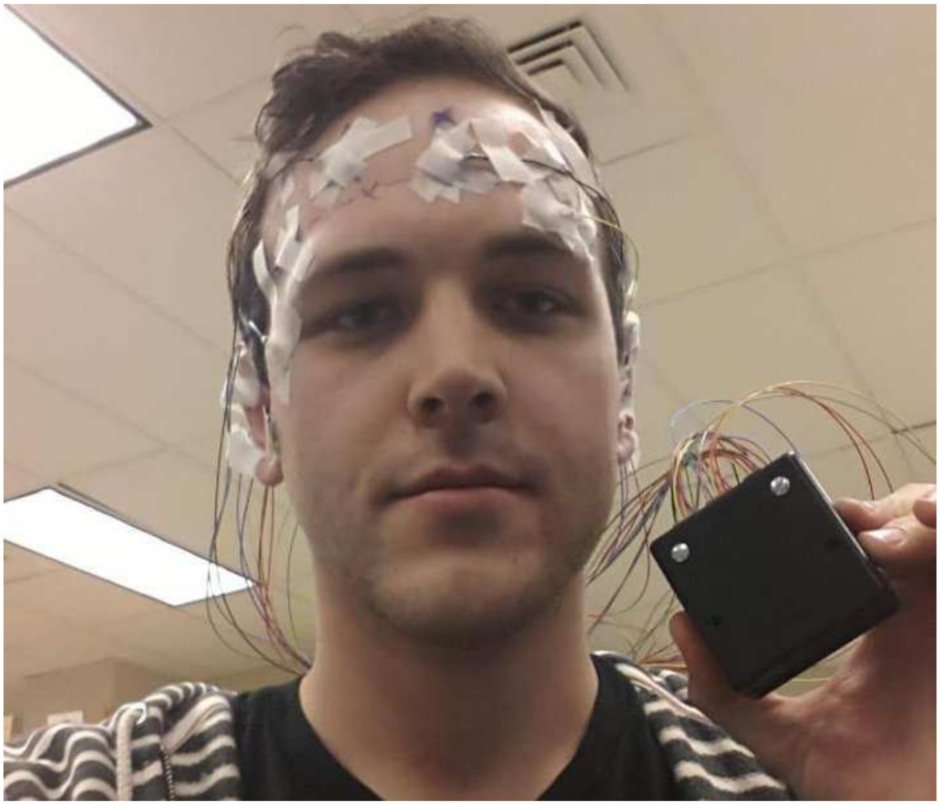


Figure 10: Device in use on the author

channels. In th multi-device configuration, the master ADS1299 is configured to generate this bias, while the slaves provide the averages of their channels to the master. While the devices can be configured in a number ways to reduce the number of I/O pins used for communications, all three ADS1299s are configured as though they are on independent SPI buses with the FPGA.

While the only second-generation boards implemented thus far feature the ADS1299, the board can also be used with the significantly lower-power ADS1298 if an alternative power supply is populated.

4.2 Embedded Seizure Detection Environment

In the interest of parallel development, an ARM-based development kit was used to facilitate embedded algorithm development. In the interest of minimizing extraneous work, a raspberry pi was purchased. The raspberry pi is an inexpensive DIY-oriented ARMv6 board. The processor on the board is a BCM2835, which is an ARM11 processor with hardware floating point support from Broadcom. There are several distributions of Linux targeted to the raspberry pi specifically, with extensive community support. Our software compiles and executes on both PC and ARMv6. Precise power measurements were not made during ARM11 execution, but the estimated power consumption with the current ARM environment kit is approximately 400 milliwatts during algorithm execution [1]. While this conservative estimate is relatively high for an embedded device, power optimization has not yet been explored for this device. Lower-power devices, such as Silicon Labs/Energy Micro EFM32 devices support a newer ARM instruction set at a significantly lower power consumption point [30].

5 Results

There are a few important performance metrics to consider for the purposes of seizure onset detection:

1. **Sensitivity**: the ability to detect as many seizure onsets as possible.

2. **Specificity**: the ability to detect only true seizure onsets; a seizure detector with low specificity will have a high false positive rate.

3. **Latency**: the time delay from the electroencephalographic onset of a seizure to the detected onset of a seizure. For the purposes of evaluation, electroencephalographic onsets used here are those assigned by the epileptologist who annotated the EEG data sets.

The ideal detector exhibits perfect specificity - zero false positives, perfect sensitivity - zero false negatives, and perfect latency - zero seconds of actual onset to declared onset delay. These results are unlikely to be achieved in a real detector. In a real detector, these three statistics are difficult to optimize all at once. An increase in sensitivity (a reduction in false negative rate) tends to come with the cost of a decrease in specificity, an increase in detection latency, or both. An increase in specificity (a reduction in false positive rate) tends to come with the cost of a decrease in sensitivity, an increase in detection latency, or both. A decrease in detection latency tends to come with reduced sensitivity, reduced specificity, or both.

Based on empirical observations, high-latency declarations, such as declaration latencies greater than 10 seconds, are uncommon. If a seizure is going to be detected reliably, it is detected quite early in the onset. Instead, high sensitivity correlates strongly with reduced latency. Particularly, the set of detectors with the highest degree of sensitivity display identical or nearly-identical detection latencies for a given voting scheme. From this set of detectors, the detector with highest sensitivity, lowest latency, and lowest false positive rate is selected.

First, the performance of the implementation described above will be compared to that of the reference implementation in [28] with a few case studies from the same data set. Then, detection results on new data will be presented and analyzed.

5.1 Detector Performance

The performance of the detector implementation will first be benchmarked against the results of the reference implementation in [28]. After discussing the differences in performance, detection results on previously unpublished data will be presented and discussed. The results process has a few parameters of its own to keep in mind in addition to the SVM detector's various parameters.

As discussed in 3.5, there are several interesting parameters to control the performance of the detector; m is the number of samples voting, n is the number of votes to pass, j is the pre-onset detection window, k is the post-offset detection window, and p is the cooldown timer duration. The results presented below will use and discuss different combinations of n and m, but the remaining parameters were fixed for all presented runs. Parameter j was 120 seconds. Parameter k was 30 seconds. Parameter p was 160 seconds.

5.1.1 Reference Comparison

The reference implementation's results were acquired against the PhysioNet CHBMIT database [15]; this is a publicly-available database of over 600 hours of EEG recordings containing 192 seizures across 23 subjects. From this database, patients 1, 2, 3, 6, 7, 8, and 23 were selected for comparisons. This selection was largely a function of which patients had EDF files that worked

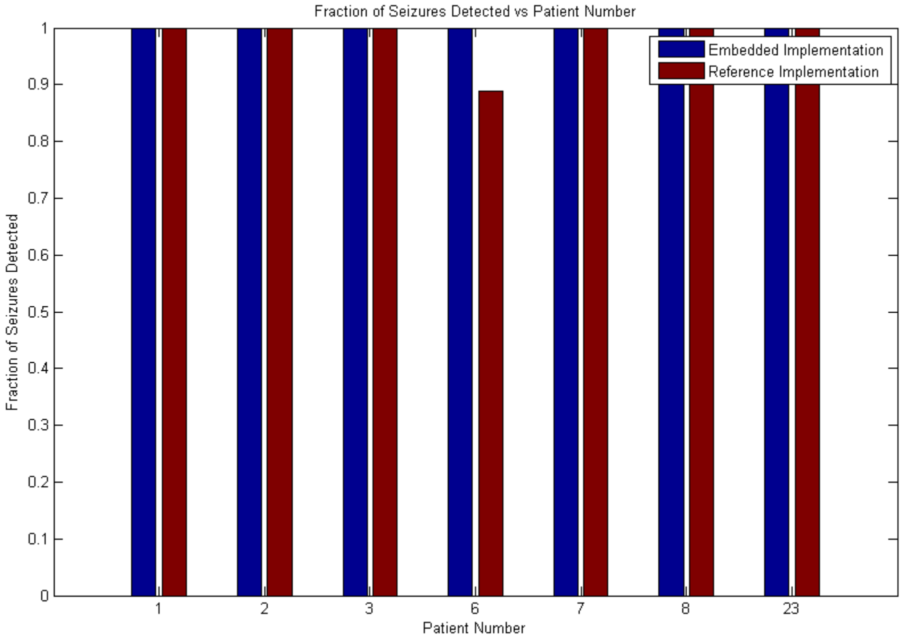


Figure 11: Percentage of seizures detected

with our loading library and did not have channel configuration adjustments during the recording process, which would require additional preprocessing to align the channels across all recordings. These 7 patients adequately establish a baseline level of performance against the reference implementation, while the additional results on new data further demonstrate the capabilities and limitations of the design.

First, the percentage of detected seizures is displayed in figure 11. The embedded implementation compares favorably to the reference implementation[28] by detecting all of the seizures in the subset; the reference implementation missed one. This is slightly better than the overall 96% detection rate established in [28].

A comparison of false positives declared per 24 hours can be seen in 12. The embedded implementation performs exceptionally well on average, with

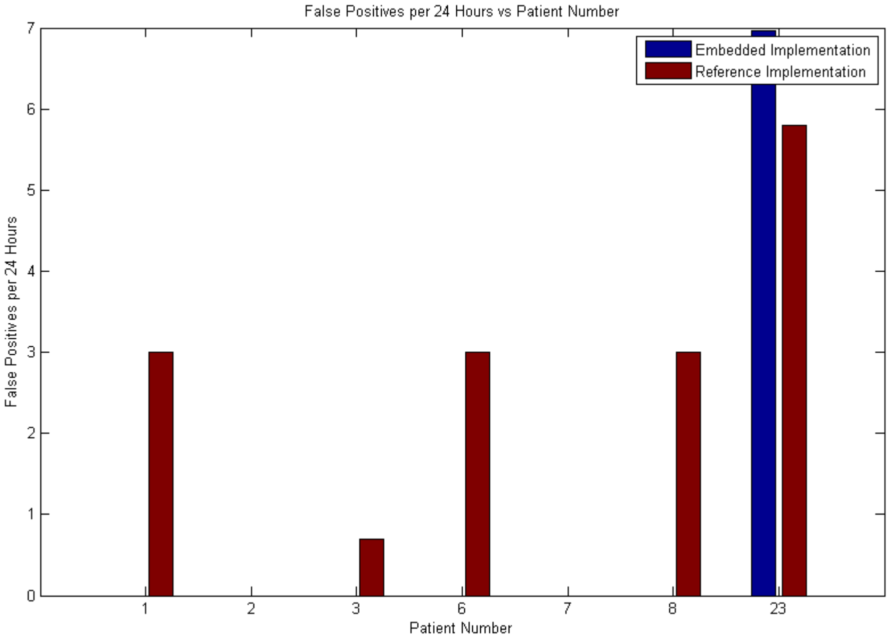


Figure 12: False positives per 24 hours

just one patient displaying any false positives at all. This outlier is displays 7 false positives per 24 hours, giving an average of 1 false positive per 24 hours.

Detection latencies, as seen in figure 13, are extremely competitive. The reference implementation had an average detection latency of 3.39 seconds, while the embedded implementation had an average detection latency of 3.54 seconds.

Our embedded detector implementation has demonstrated competitive detection performance in the categories of sensitivity, specificity, and latency. Our implementation accomplished superior sensitivity and specificity at the cost of 0.15 seconds of detection latency.

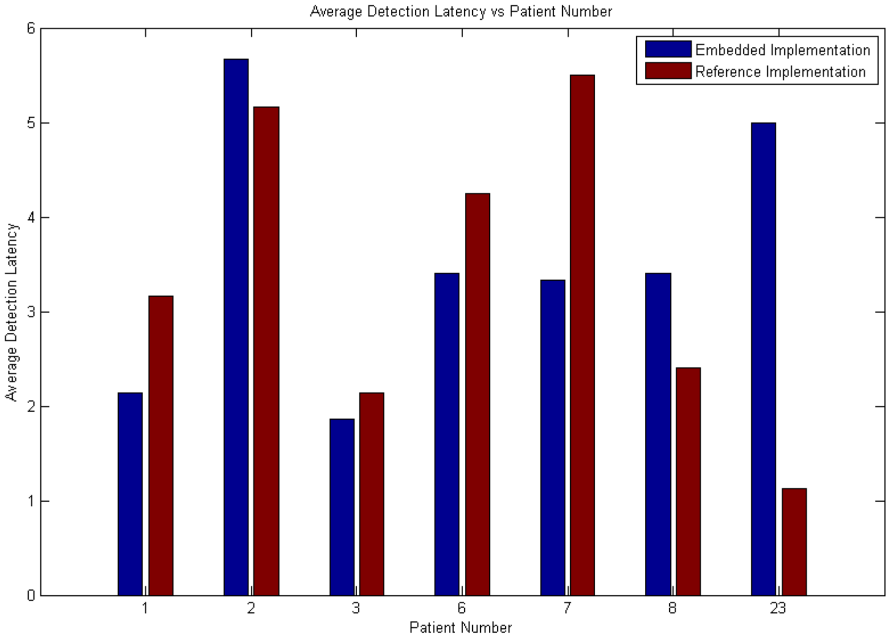


Figure 13: Average detection latencies

5.1.2 The European Epilepsy Database Results

The European Epilepsy Database is a pair of databases that can be licensed by research organizations [22]. There are two databases of seizure data; one database features intracranial recordings, while the other has surface recordings. From the scalp database, 18 patients with 167 seizures in 2,682 hours of data were used. An additional 12 patients were not used at this time due to time constraints. A notable difference between this data and the CHBMIT data is the relative frequency of seizures. The European Epilepsy Database has roughly 1 seizure per 16 hours of data, while the CHBMIT database has roughly 1 seizure per 3 hours of data. The average recording duration for each patient in the European Epilepsy Database is 150 hours, while the average duration for each patient in the CHBMIT database is 26 hours.

One of the factors in the training process is the parameter A of the trainer. As mentioned in 3.3, this decimation factor was a constant 30 for the CHB-MIT database. However, in the European Epilepsy Database, there is over 5 times more non-seizure data for each seizure. Initial results with an A of 30 were mediocre, so varying this decimation factor was explored. First, the results for a decimation factor of 750 are presented. This high decimation factor sacrifices some overall performance for higher training speed, due to the highly reduced dataset being handled. In general, however, this decimation scheme is very likely to miss outliers in the nonseizure feature space, as these outliers are statistically uncommon. All of the results presented here will be with various decimation factors A, but it is hypothesized that a more robust feature space reduction algorithm, such as those presented in [40, 41], would improve detector performance and training time.

The ROC curve, or receiver operating characteristic curve, in figure 14 displays the tradeoff of sensitivity for specificity. Increased sensitivity results in decreased specificity. A perfect detector would have a point in the top-left

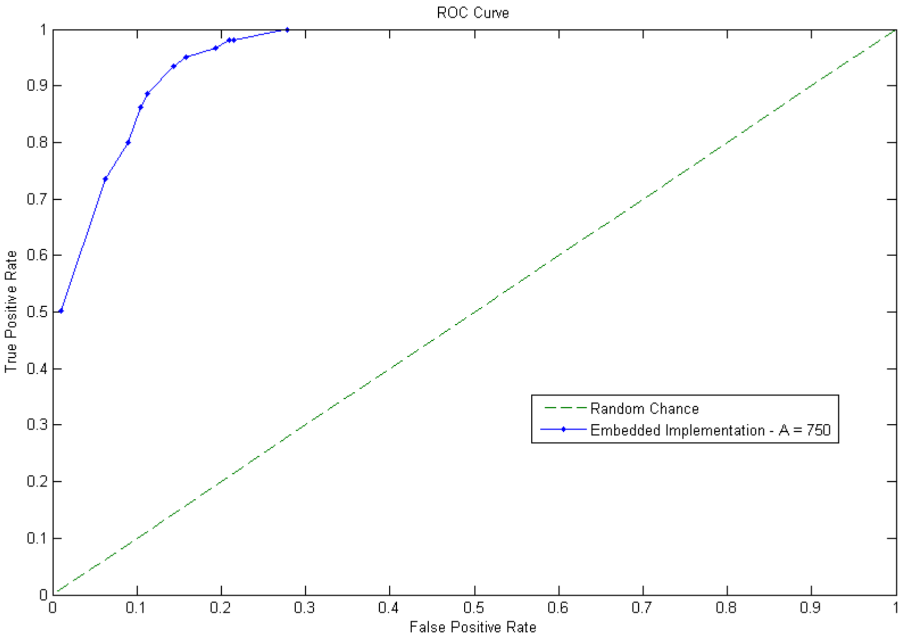


Figure 14: ROC curve of embedded implementation with A = 750 on the 18 patients selected from The European Epilepsy Database

corner - a true positive rate of 1 with a false positive rate of 0. The embedded implementation does not achieve perfection, but does exhibit significantly higher than random chance performance. The datapoints on the ROC curve correspond to a variety of n out of m voting parameter combinations: 1 out of 1, 2 out of 3, 3 out of 3, 2 out of 4, 3 out of 4, 4 out of 4, 2 out of 5, 3 out of 5, 4 out of 5, 5 out of 5, and 10 out of 10. The entry with the lowest euclidean distance to the perfect detector is the 3 out of 4 scheme. At this time, an optimal tradeoff of specificity for sensitivity has not been established, so the euclidean distance will be used.

This detector achieved a detection rate, or sensitivity, of 93.5% and a specificity of 85.5%. The average latency was 4.99 seconds. The average false positives per hour was 3.80.

The sensitivity for each patient can be seen in figure 15. The false positives per hour for each patient can be seen in figure 16. The average detection latency can be seen in 17. Compared to the CHBMIT results, these results are generally lower. The average latency is 1.5 seconds higher, the sensitivity is 6.5% lower, and the false positives per hour are 95 times higher. The changes in latency and sensitivity are acceptably small, but the false positive rate is relatively poor. In the interest of improving it, the decimation rate A of the trainer was changed from the relatively high value of 750 to 90.

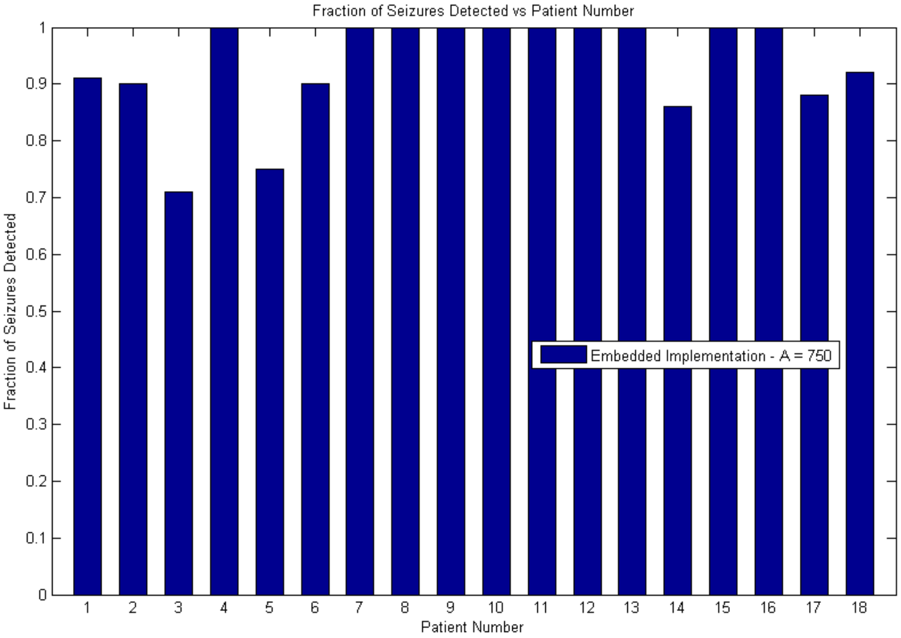


Figure 15: Fraction of seizures detected by the A = 750, n = 3, m = 4 detector on the 18 patients selected from The European Epilepsy Database

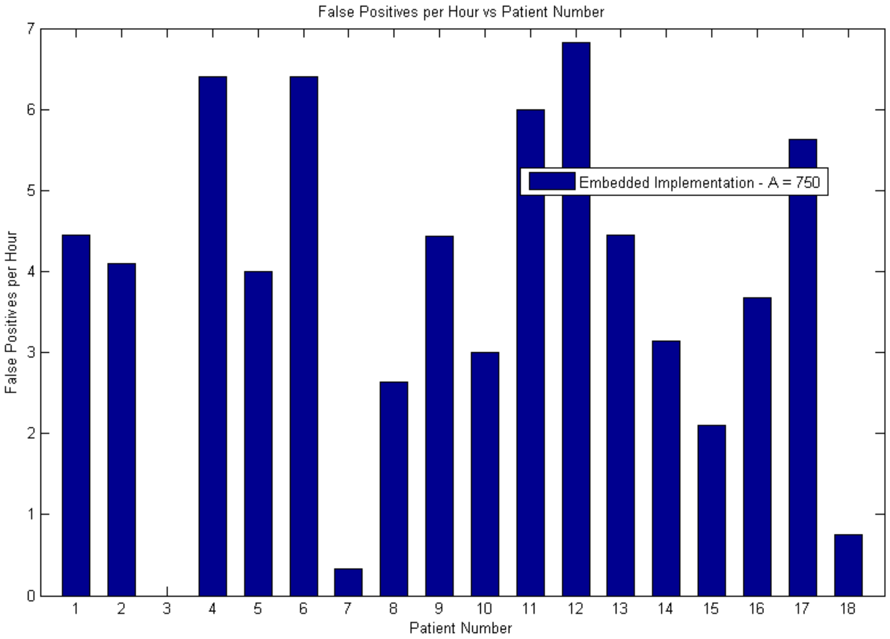


Figure 16: False positives per hour of the A = 750, n = 3, m = 4 detector on the 18 patients selected from The European Epilepsy Database

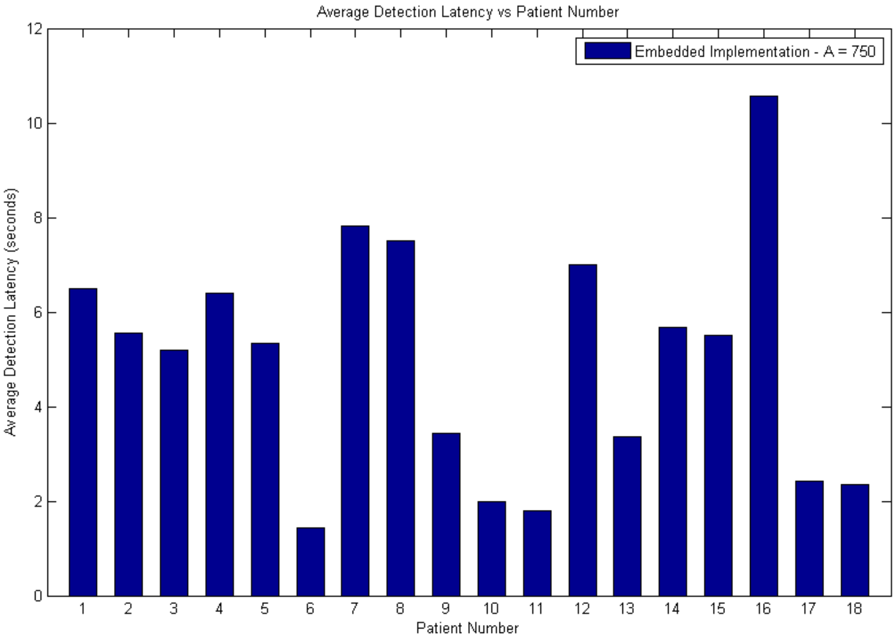


Figure 17: Average detection latency of the A = 750, n = 3, m = 4 detector on the 18 patients selected from The European Epilepsy Database

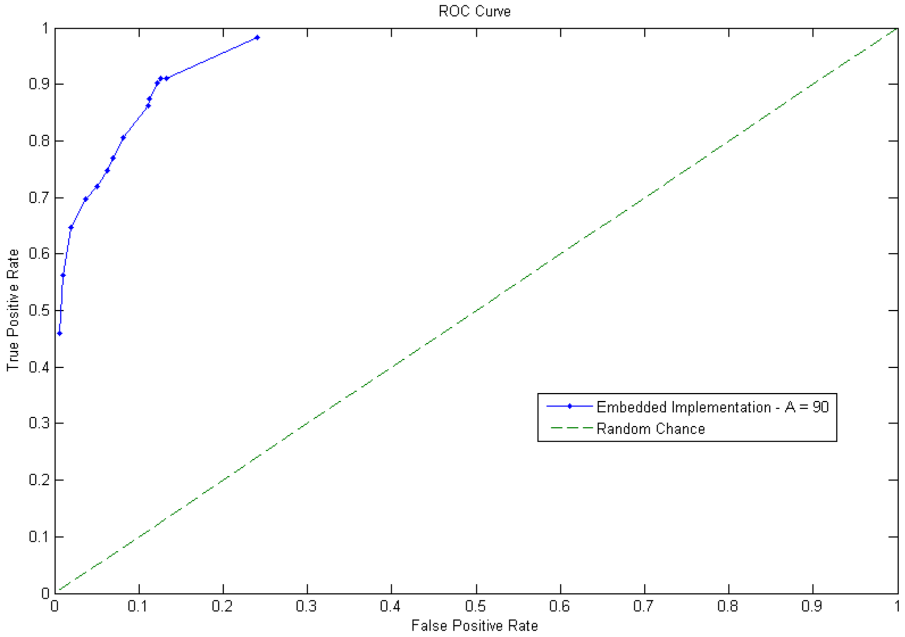


Figure 18: ROC curve of embedded implementation with A = 90 on the 18 patients selected from The European Epilepsy Database

The A=750 results displayed strong sensitivity and latency, but weak specificity. An A value relatively closer to the original value of 30 was sampled next, because preliminary results collected with a wide variety of A values indicated that a lower A value would yield higher specificity (but reduced sensitivity) for a given voting parameter combination.

Unfortunately, the change in performance for a given voting parameter combination does not shift the ROC curve in figure 18 as a whole. Instead, the voting parameter combinations are shifted along approximately the original curve, such that the overall performance is the same even though an individual voting parameter combination's performance has changed. The n out of m voting parameter combinations used in this ROC curve were 1 out of 1, 2 out of 3, 3 out of 3, 2 out of 4, 3 out of 4, 4 out of 4, 2 out of 5, 3 out

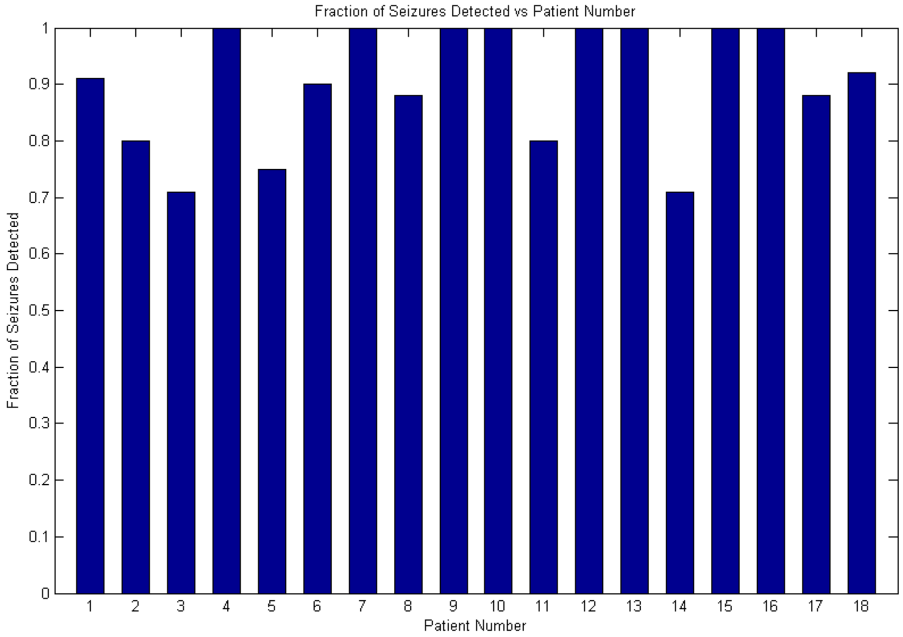


Figure 19: Fraction of seizures detected by the A = 90, n = 2, m = 5 detector on the 18 patients selected from The European Epilepsy Database

of 5, 4 out of 5, 5 out of 5, 2 out of 6, 3 out of 6, 2 out of 7, and 3 out of 7. using the same euclidean distance overall performance metric, a detector with n = 2, m = 5 was selected for closer evaluation.

This detector achieved a sensitivity of 90.3% and a specificity of 87.7%. The average latency was 6.24 seconds. The average false positives per hour was 3.09.

The sensitivity for each patient can be seen in figure 19. The false positives per hour for each patient can be seen in figure 20. The average detection latency can be seen in 21. Compared to the A = 750 results, these results show a slight tradeoff of sensitivity and latency for improved specificity. Sensitivity decreased by 2.2% and average latency increased by 1.25 seconds, but there were 0.71 fewer false positives per hour.

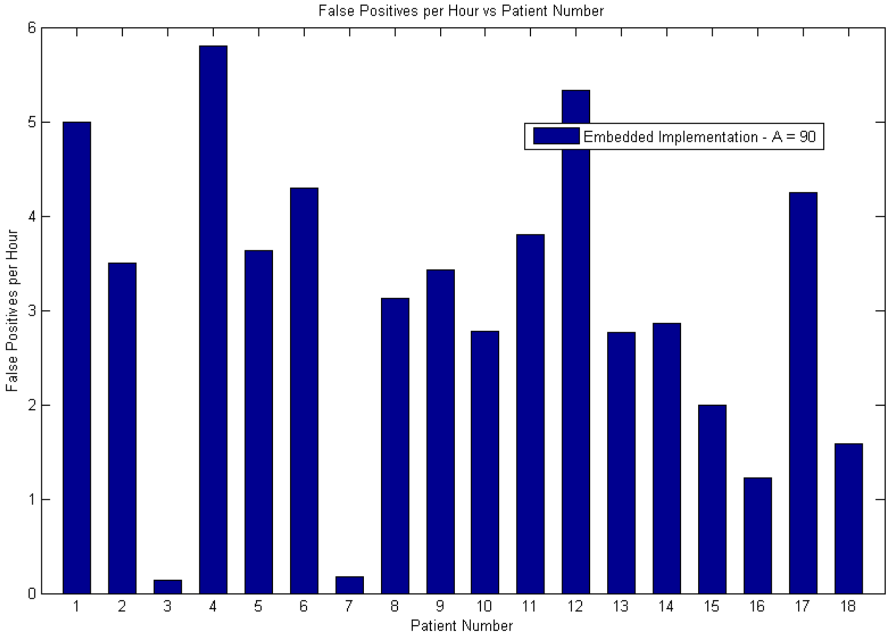


Figure 20: False positives per hour of the A = 90, n = 2, m = 5 detector on the 18 patients selected from The European Epilepsy Database

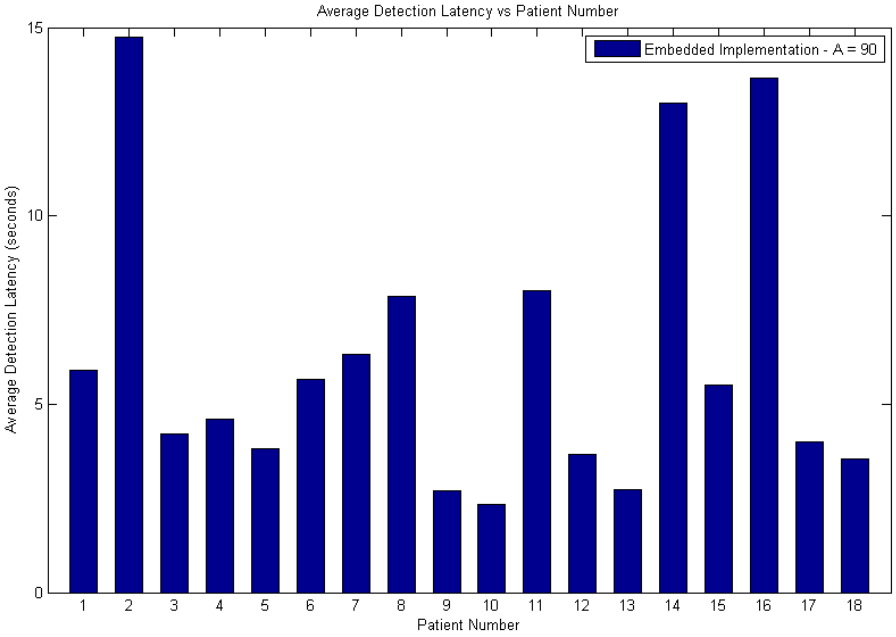


Figure 21: Average detection latency of the A = 90, n = 2, m = 5 detector on the 18 patients selected from The European Epilepsy Database

While the specificity is improved relative to the A = 750 detector, it is still significantly worse than the CHBMIT specificity. At this time, it is hypothesized that an alternative non-seizure feature pre-selection process could improve specificity. Rather than use a somewhat arbitrary subset of the non-seizure feature space, a growing neural gas, such as in [40, 41], could help better express the non-seizure feature space. In the current implementation, increasing A decreases the probability of selecting unusual non-seizure features, such as muscle artifacts, mechanical interface artifacts, or inter-ictal spikes, for the non-seizure training space. These features may then be easily misclassified as seizures during the detection process.

6 Conclusion

We have successfully developed and deployed an embedded ambulatory EEG recording platform. We have also successfully developed and evaluated an embedded seizure detection algorithm. This algorithm exceeds the baseline performance of the design established in [29, 28] against the same patient dataset. This algorithm also displays high sensitivity and low latency characteristics against the significantly larger European Epilepsy Database, but with reduced specificity relative to the CHBMIT database. It is hypothesized that an improved non-seizure feature space selection algorithm would improve this specificity.

In addition to enhancing the feature space selection of the classifier, there are additional datasets to process. Additional data in the European Epilepsy Database still needs to be evaluated, and there is another 20 patient database coming from Boston Children's Hospital. These datasets will be evaluated with future revisions of the algorithm and system as a whole.

References

[1] ARM. ARM1176 Processor - http://www.arm.com/products/processors/classic/arm11/arm1176.php.

[2] Gus A. Baker, Ann Jacoby, Deborah Buck, Carlos Stalgis, and Dominique Monnet. Quality of Life of People with Epilepsy: A European Study. *Epilepsia*, 38(3):353–362, March 1997.

[3] G Bodenstein and H M Praetorius. Feature extraction from the electroencephalogram by adaptive segmentation. *Proceedings of the IEEE*, 65(5):642–652, 1977.

[4] Paul Boon, Kristl Vonck, Veerle De Herdt, Annelies Van Dycke, Maarten Goethals, Lut Goossens, Michel Van Zandijcke, Tim De Smedt, Isabelle Dewaele, Rik Achten, Wytse Wadman, Frank Dewaele, Jacques Caemaert, and Dirk Van Roost. Deep Brain Stimulation in Patients with Refractory Temporal Lobe Epilepsy. *Epilepsia*, 48(8):1551–1560, March 2007.

[5] BE Boser, IM Guyon, and VN Vapnik. A training algorithm for optimal margin classifiers. *. . . of the fifth annual workshop on . . .*, 1992.

[6] CC Chang and CJ Lin. LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and . . .*, 2(3):27:1–27:27, 2011.

[7] H Choi, M R Winawer, S Kalachikov, T A Pedley, W A Hauser, and R Ottman. Classification of partial seizure symptoms in genetic studies of the epilepsies. *Neurology*, 66(11):1648–1653, 2006.

[8] Isa Conradsen, Sándor Beniczky, and Peter Wolf. Automatic multimodal intelligent seizure acquisition (MISA) system for detection of mo-

tor seizures from electromyographic data and motion data. *Computer methods and ...*, In Press,(2):1–14, August 2012.

[9] C Cortes and V Vapnik. Support-vector networks. *Machine learning*, 1995.

[10] RJ Croft and RJ Barry. Removal of ocular artifact from the EEG: a review. *Neurophysiologie Clinique/Clinical Neurophysiology*, 30(1):5–19, 2000.

[11] Andrew Czarnecki. *Efficient inductively coupled resonant power transfer for an implantable electroencephalography recording device.* PhD thesis, Northeastern University, 2012.

[12] Epilepsyfoundation.org. Epilepsy Incidence and Prevalence, 2013.

[13] M G Frei. Seizure detection. *Scholarpedia*, 8(3):5780, 2013.

[14] M. Frigo and S.G. Johnson. The Design and Implementation of FFTW3. *Proceedings of the IEEE*, 93(2):216–231, February 2005.

[15] a. L. Goldberger, L. a. N. Amaral, L. Glass, J. M. Hausdorff, P. Ch. 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*, 101(23):e215–e220, June 2000.

[16] II Goncharova and DJ McFarland. EMG contamination of EEG: spectral and topographical characteristics. *Clinical ...*, 114(9):1580–1593, September 2003.

[17] Google. Protocol Buffers - https://code.google.com/p/protobuf/.

[18] Br Greene, Gb Boylan, and Rb Reilly. Combination of EEG and ECG for improved automatic neonatal seizure detection. *Clinical ...*, 118(6):1348–59, June 2007.

[19] H Hallez and M De Vos. Removing muscle and eye artifacts using blind source separation techniques in ictal EEG source imaging. *Clinical ...*, 120(7):1262–1272, July 2009.

[20] Christian Hoppe, Annkathrin Poepel, and CE Elger. Epilepsy: accuracy of patient seizure counts. *Archives of neurology*, 64(11):1595–1599, 2007.

[21] CW Hsu, CC Chang, and CJ Lin. A practical guide to support vector classification. 1(1):1–16, 2003.

[22] Matthias Ihle, Hinnerk Feldwisch-Drentrup, César A Teixeira, Adrien Witon, Björn Schelter, Jens Timmer, and Andreas Schulze-Bonhage. EPILEPSIAE - a European epilepsy database. *Computer methods and programs in biomedicine*, 106(3):127–38, June 2012.

[23] Kuan-Ming Lin and Chih-Jen Lin. A study on reduced support vector machines. *IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council*, 14(6):1449–59, January 2003.

[24] Florian Mormann, Ralph G Andrzejak, Christian E Elger, and Klaus Lehnertz. Seizure prediction: the long and winding road. *Brain : a journal of neurology*, 130(Pt 2):314–33, February 2007.

[25] XL Nguyen, Ling Huang, and AD Joseph. Support vector machines, data reduction, and approximate kernel matrices. *Machine Learning and Knowledge ...*, 2008.

[26] TME Nijsen and RM Aarts. Automated detection of tonic seizures using 3-D accelerometry. *... European Conference of ...*, pages 188–191, 2009.

[27] MZ Poh. Continuous assessment of epileptic seizures with wrist-worn biosensors. 2011.

[28] AH Shoeb. Application of machine learning to epileptic seizure onset detection and treatment. 2009.

[29] Ali Shoeb, Herman Edwards, and Jack Connolly. Patient-specific seizure onset detection. *Epilepsy & Behavior*, 1:419–22, January 2004.

[30] Silicon Labs and Energy Micro. EFM32 Wonder Gecko Microcontroller Family - http://www.energymicro.com/products/efm32-wonder-gecko-microcontroller-family.

[31] Joseph Sirven, Barbara Whedon, Deborah Caplan, Joyce Liporace, David Glosser, Joyce O'Dwyer, and Michael R. Sperling. The Ketogenic Diet for Intractable Epilepsy in Adults: Preliminary Results. *Epilepsia*, 40(12):1721–1726, December 1999.

[32] V. Srinivasan, C. Eswaran, Sriraam, and N. Artificial Neural Network Based Epileptic Detection Using Time-Domain and Frequency-Domain Features. *Journal of Medical Systems*, 29(6):647–660, December 2005.

[33] Abdulhamit Subasi. Application of adaptive neuro-fuzzy inference system for epileptic seizure detection using wavelet feature extraction. *Computers in Biology and Medicine*, 37(2):227–244, 2007.

[34] Abdulhamit Subasi. EEG signal classification using wavelet feature extraction and a mixture of expert model. *Expert Systems with Applications*, 32(4):1084–1093, 2007.

[35] WO Tatum. Artifact and recording concepts in EEG. *Journal of clinical . . .*, 28(3), 2011.

[36] AT Tzallas and MG Tsipouras. Automated epileptic seizure detection methods: a review study. *Epilepsy-histological ...*, 2012.

[37] Carolyn Unsworth. Living with epilepsy: Safety during home, leisure and work activities. *Australian Occupational Therapy Journal*, (April):89–98, 1999.

[38] Vladimir Vapnik. Pattern recognition using generalized portrait method. *Automation and remote control*, 24:774–780, 1963.

[39] Xinyi Yong, RK Ward, and GE Birch. Facial EMG contamination of EEG signals: Characteristics and effects of spatial filtering. *Communications, Control and ...*, (March):12–14, 2008.

[40] Lean Yu, Shouyang Wang, and Kin Keung Lai. Investigation of Diversity Strategies in SVM Ensemble Learning. *2008 Fourth International Conference on Natural Computation*, pages 39–42, 2008.

[41] Mario Zechner and Michael Granitzer. A competitive learning approach to instance selection for support vector machines. *Knowledge Science, Engineering and ...*, 2009.