Modular Framework for Detection of Inter-ictal Spikes in iEEG

Filip Kesner[1], Lukas Sekanina[1] and Milan Brazdil[2]

***Abstract*— In this paper, we present a new modular approach for detection of inter-ictal spikes in intracranial iEEG recordings from patients that are suffering from pharmacoresistant form of epilepsy. This new approach is presented in the form of a detection framework consisting of three primary modules: first level detector, second level feature extractor, and third level detection classifier, where each module is responsible for a specific functionality. This detection framework can be perceived as a three slot system, where modules can be easily plugged in their slots and replaced by a different module or implementation on demand, in order to adapt the quality of detection (measured in terms of sensitivity, precision or inter-recording adaptability) and computational cost. Using complex real-world data sets it was confirmed that the proposed framework provides highly sensitive and precise detection, while it also significantly reduces the computation time.**

I. INTRODUCTION

Electrical signals measured directly from the brain, specifically signal events known as inter-ictal spikes are one of the essential biomarkers used for an epilepsy diagnosis and research. It is believed that spikes participates in epileptiform process [1][7]. The inter-ictal spikes can be recorded also by the scalp EEG technique, but for better localization of their source, usually for surgical treatment of epilepsy, it is necessary to acquire intracranial recordings by depth electrodes and/or subdural electrode grids. Recordings are usually acquired in more than hundred channels simultaneously, and recording process runs for several hours per patient. With a common 5 kHz sampling rate that is used in order to allow also detection of other biomarkers such as HFOs [10], the generated data are of enormous size. These data would have to be analyzed by medical doctors – neurologists – manually. Although the gold standard for interictal spike detection has been and still mainly is the manual evaluation, it has been shown that higher consistency of results can be achieved by automated detection algorithms [2]. Detection algorithms can also save enormous amount of work for reviewers and provide a faster data analysis for research or even clinical practice. Several algorithms for spike detection from scalp EEG already exist [9]. But algorithms [5][2] for spike detection in intracranial EEG (iEEG) are much more scarce and they rarely address computational efficiency and speed [3].

In this paper, we present a new modular framework for detection of inter-ictal spikes in intracranial iEEG recordings. Our framework is decomposed into three key modules: first level detector, second level feature extractor, and third level detection classifier. The modules were highly optimized in terms of computation time and quality of data processing. The objective is to confirm using complex real-world data sets that the proposed framework provides highly sensitive and precise classification and significantly reduces the computation time in comparison with existing approaches.

II. SOURCE DATA

The proposed framework will be evaluated using signal data that were recorded from patients suffering from pharmacoresistant form of epilepsy. The areas of brain where the stereo electrodes have been positioned vary through patients. This variability of signal source is useful for algorithms testing, providing a complex good-quality dataset. Signals have been recorded approximately for 30 minutes, each in 129 - 150 channels. Recordings also contain 6 non-iEEG channels such as ECG, EOG, and calibration signals, which can be omitted from processing. The recording device records the data with 25 kHz sampling rate, subsequently down-sampling them into a 5 kHz range. To illustrate the enormous size of such data recordings, the channel size is 5000 Hz * (30 min * 60 sec) * 4 bytes, where the average recording file contains 150 such channels, resulting into the file size of 5.4 GB.

As recordings of intracranial EEG are huge files and terabytes of such data are available for processing, it is crucial to optimize the proposed framework for the execution time.

III. DETECTION FRAMEWORK

The proposed detection framework is constructed as a three slot system which allows automated experiment performing (Fig. 1). The modularity of our detection framework is convenient because it provides possibility to easily replace an implementation of a given module by another implementation, without the need of replacing modules in the other slots. Obtained configuration can then automatically be benchmarked and compared with other configurations of the framework or other systems. The whole design is implemented using the object oriented programming (OOP) paradigm i C++. It is not necessary to fill all three slots of the system. The detection framework can also work in several "hybrid" configurations. One of these configurations can employ only the first level detector, while feature extractor and classifier slots are empty. This is useful when we want to evaluate a detector without an influence of feature extraction and classifier. We may also want to save the computational time and memory requirements in an embedded device.

[1] Filip Kesner and Lukas Sekanina are with Faculty of Information Technology, Brno University of Technology, Brno, Czech Republic, ikesner@fit.vutbr.cz; sekanina@fit.vutbr.cz

[2] Milan Brazdil is with 1st Department of Neurology, St. Anne's University Hospital and Medical Faculty of Masaryk University, Brno, Czech Republic milan.brazdil@fnusa.cz

Another usage is to produce training and testing datasets for the following levels. Other "hybrid" configurations are useful when third-party detection algorithms are ported into our framework structure. This is advantageous for the possible comparison.

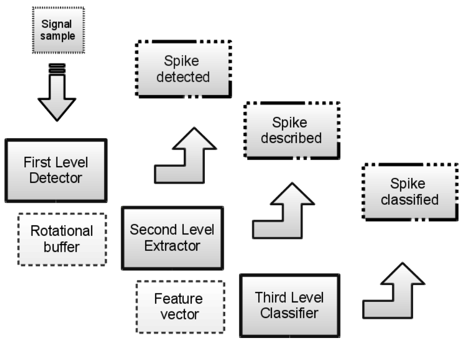

Fig. 1. Detection Framework

In this paper, we will present a configuration of the detection framework which consists of all three modules. These modules will be described in the order, in which they are applied for signal data processing. Because of very restricted space, a detailed description is devoted to the first level detector and the configuration of remaining modules is only briefly sketched.

A. First Level: Detector

The first level detector is a module responsible for marking positions or small areas, where potential spikes are present. As this module can greatly reduce the amount of data that has to be processed by second and third level modules, its design significantly influences the overall performance of the framework. If this module misses a potential detection (such as spike) in the signal, the second and third level will not encounter this signal area, and thus they cannot increase the sensitivity of the system above the sensitivity of the first level module. The output of the this module, i.e. the potential detections, can be marked in several ways, such as:

- first level trigger index, channel number
- start index, end index, channel number
- start index, top index, end index, channel number
- first level trigger index, start index, detection top index, end index, channel number

The first level module used in our experiments consists of several smaller internal modules, which are connected in a pipeline way as shown in Fig. 2. Algorithm (Alg. 1) can be perceived as a virtual energy capacitor (as we would describe it), which can have constant discharge current or dynamic discharge current proportional to the remaining energy stored in the capacitor. This virtual capacitor is represented as a $r_{max} - r_{min}$ difference, where r stands for "recent". The input signal sample is first passed to a high pass Butterworth filter of second order ($f_{HP} = 20$ Hz). The output of this filter is then passed to a low pass Butterworth filter of fourth order ($f_{LP} = 50$ Hz) in order to remove higher frequencies

Data: signal sample
Result: detected biomarkers
for *each sample* **do**
  read sample;
  filter high pass 20 Hz (sample) ;
  filter low pass 50 Hz (filtered sample) ;
  optionally store filtered sample to rotational buffer;
  if *filtered sample* $>$ *rMax* **then**
    rMax = filtered sample;
  end
  if *filtered sample* $<$ *rMin* **then**
    rMin = filtered sample;
  end
  if *rMax - rMin* $>$ *current threshold* **then**
    if *minimal time distance reached since last detection* **then**
      spike detected;
    end
  end
  decrease rMax;
  increase rMin;
  if *optional second level is enabled* **then**
    filter high pass 1 Hz (sample) ;
    filter low pass 35 Hz (filtered sample) ;
    store filtered sample to rotational buffer;
  end
end

Algorithm 1: First level detection algorithm design

(mostly interference) which are not necessary for the spike detection. The output of this second filter is then compared with r_{max}, and if it is greater, it is assigned to r_{max} variable. The output of this second filter is then compared with r_{min}, and if it is smaller, it is assigned to r_{min} variable. Then the $r_{max} - r_{min}$ difference is computed and compared against a constant threshold. If this threshold is passed and was not passed with a previous sample, the rising edge is detected and the index position is marked as a center area with a possible spike. At the end of processing of each sample, the r_{max} value is decreased and r_{min} value is increased, what will eventually lead to $r_{max} - r_{min}$ difference falling below the threshold. The decay of r_{min} and r_{max} value is independent of the sampling rate.

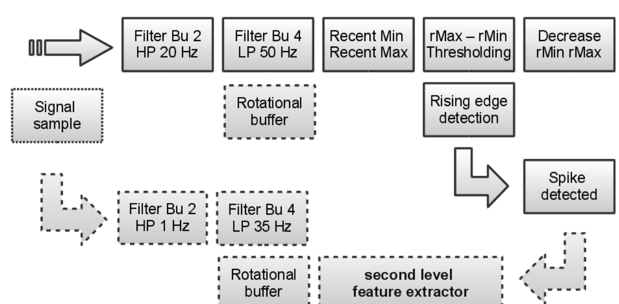

Fig. 2. First Level Detector

B. Second Level: Feature Extractor

The purpose of this module is to extract and compute features describing the signal present in the area of detection.

These features can be generic and relatively simple, such as standard deviation of signal amplitude, or they can be more complex, such as selected coefficients of frequency spectrum or wavelet transformations. They can also be specialized for characteristic description of a specific signal event such as inter-ictal spikes, or high frequency oscillations. Because of the OOP style of the framework design, it is possible to run several feature extractor modules simultaneously in one framework setup. In order to achieve the highest possible modularity, there can be instantiated one feature extractor module for each computed feature. It is also possible to have one module that computes several features, which can be enabled or disabled by a module parameter at run-time. Then an automated experiment can be executed, and based on the results, the feature extractor modules providing features with the lowest differentiating ability can be disabled in order to reduce computational time and power consumption. This approach is advantageous in battery powered devices operating inside a shielded recording room or for the future use in fully mobile (possibly wearable) devices. The core of this module, the feature computation, is performed on signal area, which is referenced by indexes, that are internally stored. Some features are computed only inside the specified area, others may also require an access to surrounding area, relative to the event index range. Features can be computed after all signal sections are processed by the first level detector and when all potential detections are marked. However, this approach requires to store all the signal data in RAM (or at least to store the signal data surrounding each detection). While this is possible with modern computer workstations (with 8+ GB of RAM), another strategy has to be taken for embedded and less-powerful devices.

When the signal data are passing through the first level detector, they can optionally be stored in a rotational buffer with a limited capacity whose size is usually optimized according to the processors cache memory size. Optionally more than one buffer can be used, if there is enough cache memory to be split between them. The second level module can be signaled automatically with specified sample/ms delay after the first level triggered detection. Then the rotational buffer can be processed while it is still present in the cache memory. Employing the rotational buffer led to reduction of the computation time and memory requirements. This reduction may vary based on how many features are computed and how many times the buffer needs to be passed through.

We have implemented several generic and some specialized feature extraction algorithms. In the experiments, the following generic features have been used: *area average, area standard deviation, area maximum, area minimum, area maximum vs. minimum difference*. We have also used some specific features inspired by Barkmeiers algorithm [2]: *amplitude, left amplitude, right amplitude, left slope, right slope* and *duration*. The features for description of inter-ictal spikes should be independent of the spike polarity to achieve better training and classification results from following modules.

C. Third Level: Detection Classifier

The third module slot is designated for a classifier responsible for classification of potential detections that are produced by the first level detector and described by the features from the second level module. The classifier can be implemented in a standard way, i.e. it will assign detection into exactly one class. The framework also supports a fuzzy classification, where percentage possibility of belonging into several classes can be defined. A simple implementation of this module can be created as a hard-coded decision structure with variable thresholds. While this may be tempting, we would advice against it. It may work for signals and features which for it was manually adapted to, but only if very effective differentiating features are available. We propose to employ a machine learning based classification algorithm such as support vector machine (SVM), artificial neural network (ANN) or decision tree (DT). The current framework setup uses a standard SVM-based classifier [8][4] with linear base function that classifies detections into exactly 2 classes.

D. Evaluation Platform

In order to fairly perform experiments with the proposed detection framework, we have implemented an evaluation platform capable of measuring parameters and properties of the first level detector (computational time, signal time / computational time ratio, signal samples / computation time ratio, cache efficiency etc.), the second level feature extractor (computational time per area, cache efficiency, etc.) and the third level classifier (training cpu time, classification cpu time etc.). Finally, the global statistical quality indicators (such as sensitivity and precision) and performance indicators are provided. Measurements can be performed in single- or multi-thread way in a parallel computing environment with multiple CPUs.

IV. EXPERIMENTS AND RESULTS

In order to evaluate the framework architecture and module setup, our experiments are performed in two scenarios. First, the framework is evaluated as a complete system. In the second scenario, only second and third level modules are applied on manually marked data. Our signal dataset contains intracranial recordings from 10 different patients, which were in various states. Data used in experiments also contains noisy and seizure recordings in order to analyze how the detection framework can handle such data input.

A. Manual Evaluation

A golden standard for spike detection practically does not exist. It was shown [2][6] that the inter-reviewer variability is huge. In order to at least partially suppress this inter-reviewer variability and also number of overlooked spikes, signals were reviewed by a two member group of biomedical engineers. Only detections where a full agreement was achieved have been considered. The signal shape got a decision priority over the amplitude of detection in order to include also spikes propagated from the surrounding area of electrode. Detections made by proposed detector have been visualized into the signal window by half-transparent marks using fading green and red colors. The group of reviewers has been counting missed spikes into one category (false negatives), spike-free detections into another (false positives) and correct detections (true positives) separately. The number of true negatives would be hard to estimate because it is the rest of the signal without marks. Results (Table I) have been obtained from uniform subset of each recording with a sensitive configuration of the first level detector (threshold = 50; constant decay = 15), in order to achieve high sensitivity of the whole system, while potential false positives were balanced out by the SVM classifier well trained on selected feature set in order to increase precision. Variable settings have been used the same for all recordings without any change or manual adaptation.

TABLE I

DETECTION FRAMEWORK SENSITIVITY AND PRECISION

signal	sensitivity [%]	precision [%]	spikes (TP)
recording 1	98.444	98.828	253
recording 2	100.00	93.678	163
recording 3	96.429	96.923	189
recording 4	97.512	99.157	588
recording 5	98.182	98.480	324
recording 6	99.618	86.424	261
recording 7	98.667	98.667	148
recording 8	99.625	96.727	266
recording 9	97.489	94.260	427
recording 10	97.692	99.219	127
overall sum	96.690	97.757	2746
Barkmeier algorithm	86.320	73.151	=

B. *Features and Classification of Manual Detections*

It is important to evaluate the framework setup as a complete system, but in order to optimize second and third level modules it is beneficial to do also an independent evaluation without first level detector. The output of this omitted first level module is simulated by manual detections with consistent length. On purpose there are manual true positives but also false positive markings containing areas with and without spikes (or other biomarkers). The raw signal recordings usually contain lot of various signal waves while the inter-ictal spikes are a minority event from the point of frequency of occurrence. For this reason training data have been artificially balanced to contain about the same amount of positive and negative detections. Evaluation has been performed on testing data, which were not used in the training process. Results (Table II) indicate, that used combination of features can efficiently differentiate event-containing and event-free signal areas.

TABLE II

FEATURE EXTRACTION AND CLASSIFICATION

signal	sensitivity [%]	precision [%]	total marks
manually selected areas	97.26	96.10	539

C. *Computational Performance*

Compared to Barkmeier's algorithm [2] (which we have chosen because it has been used by researchers at Mayo clinic) running in Matlab, which takes in average 32 minutes to process the same 35 minutes long file, with 150 channels, this first level detection algorithm implemented in C/C++ requires in average only 50 seconds running on the same hardware (HP Z420 workstation Intel(R) Xeon(R) CPU E5-1620). For the fairness of comparison we have also created an implementation of Barkmeiers algorithm in C in a very efficient manner, but it was still more than 5 times slower.

V. CONCLUSION

The proposed detection framework allows implementation of modular efficient detection, description and classification methods for inter-ictal spikes and other possible electrophysiological biomarkers such as high frequency oscillations in intracranial iEEG or even scalp EEG recordings. In general it is possible to use framework architecture even for different type of biomedical signals such as ECG, myoelectrical signals, and even more-dimensional signal data such as images. Thanks to its integration with evaluation platform it is possible to perform automated experiments and comparisons of different modules for each of the framework slots.

ACKNOWLEDGEMENTS

This work was supported by Brno University of Technology grant under number FIT-S-17-3994.

REFERENCES

[1] K. Staley, A. White, and F. Dudek, Interictal spikes: harbingers or causes of epilepsy? *Neuroscience letters*, 2011, 497(3):247-250.

[2] D. Barkmeier, A. Shah, D. Flanagan and et al., High inter-reviewer variability of spike detection on intracranial EEG addressed by an automated multi-channel algorithm. *Clinical Neurophysiology*, 2012;123(6):1088-1095.

[3] F. Kesner, J. Cimbalnik, I. Dolezalova, M. Brazdil and L. Sekanina, Fast Automated Interictal Spike Detection in iEEG/ECoG Recordings (Using Optimized Memory Access) *in proceedings of Neurotechnix conference*, 2015.

[4] Y. Pan, S. S. Ge, F. R. Tang, and A. Al Mamun. Detection of epileptic spike-wave discharges using svm. *IEEE International Conference on Control Applications*, Singapore, 2007, pp. 467-472.

[5] N. Gaspard, R. Alkawadri, P. Farooque, I. I. Goncharova, and H. P. Zaveri. Automatic detection of prominent interictal spikes in intracranial eeg: Validation of an algorithm and relationship to the seizure onset zone. *Clinical Neurophysiology*, 125:10951103, 2014.

[6] M. W. Brown III, B. E. Porter, D. J. Dlugos, J. Keating, A. B. Gardner, P. B. Storm, and E. D. Marsh. Comparison of novel computer detectors and human performance for spike detection in intracranial eeg. *Clinical Neurophysiology*, 118:17441752, 2007.

[7] J. Engel Jr., A. Pitknen, . A. Loeb, F. E. Dudek, E. H. Bertram, A. J. Cole, S. L. Mosh,S. Wiebe, B. E. Fureman, F. E. Jensen, I. Mody, A. Nehlig, and A. Vezzani. Epilepsy biomarkers. *Epilepsia*, 54:6169, 2013.

[8] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:127:27, 2011.

[9] S.S. Lodder and J. Askamp and J.A.M. van Putten, Inter-ictal spike detection using a database of smart templates, *Clinical Neurophysiology*, 2013, 124, 12, 2328-2335.

[10] G. Worrell, K. Jerbi, K. Kobayashi, J. M. Lina, R. Zelmanng and M. Le Van Quyen, Recording and analysis techniques for high-frequency oscillations, *Progress in neurobiology*, 2012, 98, 3, 265-78