Internet of Things 14 (2021) 100048

Contents lists available at ScienceDirect

Internet of Things

journal homepage: www.elsevier.com/locate/iot

Research article

EEG signal analysis for epileptic seizures detection by applying Data Mining techniques

Gaetano Zazzaro[a], Salvatore Cuomo[b,*], Angelo Martone[a], R. Valentino Montaquila[a], Gerardo Toraldo[b], Luigi Pavone[c]

[a] CIRA, Italian Aerospace Research Centre, Capua, CE, Italy
[b] Department of Mathematics and Applications, University of Naples Federico II, Naples, Italy
[c] IRCCS, Neuromed, Pozzilli, IS, Italy

ARTICLE INFO

Article history:
Received 22 January 2019
Revised 8 March 2019
Accepted 9 March 2019
Available online 27 March 2019

Keywords:
Data Mining
Epilepsy
Electroencephalogram
Support Vector Machine
Signal processing
Sliding window

ABSTRACT

Epilepsy is a chronic neurological disorder characterized by frequent seizures, which severely impact the quality of life of epilepsy patients and sometimes are accompanied by loss of consciousness. The most widely accepted and used tool by epileptologists to identify seizures and diagnose epilepsy is the ElectroEncephaloGram (EEG). Seizure detection on EEG signals is a long process, which is done manually by epileptologists. This paper describes how to analyze EEG signal using Data Mining methods and techniques with the main objective of automatically detect a seizure within EEG signals. We have designed and developed a multipurpose and extendable tool for feature extraction from time series data, named Training Builder. Our trained classifier, based on signal processing, sliding window paradigm, features extraction and selection, and Support Vector Machines, showed excellent results, reaching an accuracy of over 99% during the test made on publicly available EEG datasets.

1. Introduction

Epilepsy is a chronic brain disorder characterized by recurrent seizures with different kind of side effects for the patients, from involuntary movements of part of the body (partial) or of the entire body (generalized) to loss of consciousness, which severely impact their quality of life.

Only 70% of patients respond to medical treatments [1], while for the remaining 30% other approaches are needed to manage seizures and improve their quality of life. The most commonly used clinical tool to monitor and diagnose epilepsy is ElectroEncephaloGram (EEG) signal recorded from scalp or intracranially (iEEG), using electrodes inserted epidurally, subdurally or in deep structures of the brain.

Patients with epilepsy generally undergo to continuous EEG monitoring, in order to detect clinical seizures and identify the epileptogenic zone.

Detection of seizures in EEG is commonly made manually, by visual inspection of epileptologists. This process is extremely time-consuming, because of the big amount of data generally available for each patient, and can also lead to

* Corresponding author.
E-mail addresses: g.zazzaro@cira.it (G. Zazzaro), salvatore.cuomo@unina.it (S. Cuomo), a.martone@cira.it (A. Martone), r.montaquila@cira.it (R.V. Montaquila), toraldo@unina.it (G. Toraldo), bioingegneria@neuromed.it (L. Pavone).

https://doi.org/10.1016/j.iot.2019.03.002
2542-6605/

different diagnosis, because the diagnosis of epilepsy is quite subjective and often depends on the experience and on the skills of the epileptologists.

Furthermore, the information provided by the patients and their relatives about seizure severity, seizure frequency and seizure type has been shown to be highly unreliable [2].

All these issues in the diagnosis of epilepsy have encouraged researchers to develop computerized and automated methods for seizure detection, in order to provide a valuable tool for clinicians to speed up the process of seizure detection and at the same time precious data for epilepsy management, providing a higher detection accuracy.

Automatic seizure detection methods aim to detect ongoing seizures in EEG as soon as possible after seizure onset.

Generally, automated method for seizure detection in EEG consists of a preprocessing step, then a feature extraction step from pre-processed EEG signal and then a features classification step to distinguish epileptic from non-epileptic signals.

Among these steps, the most important one is the extraction of highly specific features, able to distinctive characterize EEG patterns, typical of a seizure. In fact, even if classification step is also very important, a good set of features would make the classification problem easier and more accurate.

In this paper, we present a seizure detection method, based on the extraction of 26 features, both bivariate or univariate, by means of an ad-hoc implemented software tool, called Training Builder, and seizures classification using Data Mining (DM) techniques such as Support Vector Machines.

We think that the use of many features, some of them never experienced in seizure detection, together with the implemented pre-processing steps, may represent an advantage of our method, compared to the state of the art methods in seizure detection.

This paper has been structured according to the standard process conceived from the Cross-Industry Standard Process for DM (CRISP-DM) [3] that is organized in six steps: Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation and finally Deployment.

1.1. Structure of the paper

The paper is organized by describing all CRISP phases one by one. In Section 2, the Business Understanding is carried on in order to understand the disorder of epilepsy, the state of the art of seizures detection by applying Data Mining techniques, and to fix the DM goals. Section 3 describes data collection and data sources. In Section 3, all the Data Preparation steps are explained too: in particular, the signal preprocessing phase, the developed software tool, the features calculation, and the hold-out method. In Section 4, details about Modeling are provided: from the training phase to testing phase, including errors analysis evaluation. Finally, in Section 5 a use case of the trained classifier, the conclusion, and the future works are showed.

2. Business understanding

The first step of the CRISP-DM process includes the definition of the business objectives and the DM goals. Moreover, the state of the art of DM techniques for seizure detection is also reported.

2.1. Epilepsy and state of the art of seizure detection

The ultimate goal of epilepsy treatment is to provide seizure control for all the patients, especially those who do not respond to medical treatment.

Until now, many studies had focused on finding automated seizure detection methods not only to support epileptologists during epilepsy diagnosis and detect seizures from EEG signals but also for developing algorithms to be embedded into closed-loop system for epilepsy treatment. Such systems should be able to identify the incoming of a seizures and then take a decision, like sending an alarm to the patient or aborting the seizure stimulating electrically the electrode where the seizure is recorded, for example.

Generally, automatic seizure detection methods in EEG consists of a preprocessing step, then a feature extraction step from pre-processed EEG signal and then a features classification and selection step.

The key step in a seizure detection method is the choice of the features to be used for the classification step. Many types of features have been proposed during the last years, such as entropy-based approaches [4–9], largest Lyapunov exponent [10], energy [11], wavelet transform [12,13].

Classification step is also critical in such methods, in fact many type of classification approaches have been used as well, such as artificial neural network [14], Bayesian linear discriminant analysis [15], fuzzy-rule based [16,17]. In [18] the authors apply machine learning techniques, in particular Support Vector Machine, k-nearest neighbor, and Naïve Bayes, for the classification of preictal and ictal states, based on statistical and spectral moments. Moreover, a complete flowchart for epilepsy prediction is provided, including feature extraction step.

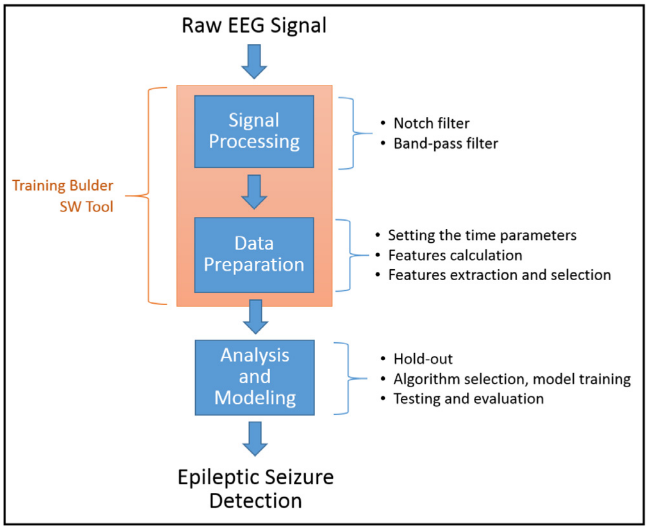

Fig. 1. Process flow for seizure detection.

2.2. Data Mining process for seizure detection

Fig. 1 shows the general process flow for seizure detection. The process is a customization of the Knowledge Discovery in the Database Process (KDD), widely used for DM tasks [19–22]. The figure also displays, in the orange box, the functionalities of the developed SW tool, called Training Builder, useful in the Data Preparation and Signal Processing steps.

The Analysis and Modeling step focuses on Statistics and Data Mining algorithms aimed at building a mathematical model for epileptic seizures detection.

In the next sections, all the steps of this process are described.

2.3. Business objective and Data Mining goals

The business objective is to develop an algorithm able to detect epileptic seizures from EEG recordings of epileptic patients, using DM techniques, in order to detect the seizures automatically. The problem of seizure detection can be seen as a problem of binary classification, which discriminates EEG with seizures from EEG without seizures. Moreover, classification models will be trained in order to detect the incoming of epileptic seizures.

3. Data understanding and preparation

The Data Understanding step of CRISP-DM includes the initial data collection, data description, data exploration, and the data quality checking. Whilst during the Data Preparation phase, data are processed and expressed in a format accepted by DM algorithms, in order to obtain the final dataset to be used in the next Modeling phase. Moreover, a description of the SW tool for signal processing and features extraction is provided hereafter. The Data Preparation step is the longest and most challenging phase of the whole process. In other application scenarios the Internet of Things methodologies are used to collect and infer information in order to extract knowledge [23,24].

3.1. Freiburg seizure EEG database

In order to test our algorithm, we used data from Freiburg Seizure Prediction EEG database (FSPEEG) [25,26]. This EEG database was made available for researchers working on seizure prediction and detection. The database was collected from 21 patients with drug-resistant epilepsy and contained intracranial EEG recordings acquired during invasive presurgical epilepsy monitoring at the Epilepsy Center of the University Hospital of Freiburg, Germany. Recordings were made with a 128 channel EEG system at 256 Hz sampling rate with 16 bit A/D converter. Six EEG channels were selected by certified epileptologists by visual inspection, three from electrodes located near to the region where the seizures started or where seizure activity had been detected (“InFokus” channels) and three from electrodes located in areas distal from the seizure focus, where no seizure activity had been detected (“OutFokus” channels). Furthermore, for each patient, two kind of recordings are available: “ictal” recording, containing seizures, and “preictal” recordings, which are recordings acquired immediately before or after the ictal ones but they does not contain seizures. In order to test our method, we used data from only one representative patient (patient number 16).

3.2. Signal processing

A digital signal is the sum of many digital samples, always affected by noise, partly inherited from the analog signal, namely the physical quantity under examination, partly due to the quantization during the process of Analog Digital Conversion (ADC). For this reason, it is crucial to filter the signal before of any other processing step [27].

In our case, the EEG signal from each InFokus/OutFokus electrode was filtered through six different frequency bands (8–12 Hz, 13–20 Hz, 21–30 Hz, 30–45 Hz, 40–70 Hz and 70–120 Hz) by using a band-pass filtering, thus obtaining six signals. The upper limit of 120 Hz is determined by the sampling frequency used during signal acquisition (256 Hz), according to the Nyquist–Shannon sampling theorem. The frequency bands used in this work for signals processing correspond to well-known human brain oscillations and they cover almost the entire available frequency spectrum.

A generic signal can be seen as a superposition of modes in the frequency domain. In general, for a square-integrable function x(t) in R, the following relation (1) defines its Fourier transform:

$$X(f) = \int_{-\infty}^{+\infty} x(t)\ e^{-j2\pi ft} dt. \tag{1}$$

Consequently, by using the Fourier transform, the signal $x(t)$ can be expressed as follows (2):

$$x(t) = \int_{-\infty}^{+\infty} X(f)\ e^{j2\pi ft} df. \tag{2}$$

The band-pass filters used have been implemented, in the Matlab environment (Matlab R2015a, The Mathworks Inc., USA), by using the FFT (Fast Fourier Transform) method. In particular, the following formula was used (3):

$$\mathrm{Output} = \mathrm{IFFT}(\mathrm{Passband} * \mathrm{Fourier}) \tag{3}$$

where *Fourier* represents the FFT of the input signal, *Passband* the filters discussed above and implemented in the frequency domain and finally *Output* is the IFFT (Inverse FFT) of their product and represents the filtered signal.

3.3. Sliding window technique

In order to train a classifier able to recognize specific fundamental fractions of the signal within time series, we applied sliding window technique, which is a strategy widely used in Machine Learning and Stream Data Mining [28–30]. Sliding window is a method to rearrange a time series dataset as a supervised learning problem. A class of algorithms for stream processing focuses on the last points of time series by applying a sliding window on the data. In this way, only the last values of each streaming time series is considered for the analysis. Given a continuous time series stream, the sliding window technique (or paradigm) examines the most recent data points and moves S steps along the time axis as new measurements arrive, where s is the step size. In other words, every S points (or seconds) the analysis focuses on the last L points (or seconds) of the time series. S and L are the temporal shift and the length of the window, respectively.

The sliding window moves on the time axis identifying a group of k ordered data. If f is the data sampling rate (or time series frequency), and L is the length of the window in seconds, $k = f^*L$ is the number of the points in the window. While it is not easy to fix the length L of the sliding window (or the number k of its points), the value of L is related to both the historical series and the analysis technique used. Each window may have elements in common with the previous window, but if there are no elements in common between the two windows ($S > L$), the sliding window is called “tumbling window”. In most applications, each window is passed to a data processing unit, which performs some kind of time series classification, clustering, or anomaly detection.

Windowing is one of the most frequently used processing methods for data streams. An unbounded stream of data (events) is split into finite sets, or windows, based on specified criteria, such as time. A window can be conceptualized as an in-memory table in which events are added and removed based on a set of policies.

This subsection describes how sliding and tumbling windows work. Both types of windows move across continuous streaming data, splitting the data into finite sets. Finite windows are helpful for operations such as aggregations, joins and pattern matching.

3.3.1. Tumbling window

In a tumbling window, tuples are grouped in a single window based on time or count. A tuple belongs to only one window. For example, a tumbling window can be evaluated every five seconds, with no overlap between different time windows; each segment represents a distinct time segment.

3.3.2. Sliding window

In a sliding window, tuples are grouped within a window that slides across the data stream according to a specified interval. A time-based sliding window with a length of ten seconds and a sliding interval of five seconds contains tuples that arrive within a ten-second window. The set of tuples within the window are evaluated every five seconds. Hence, sliding windows can contain overlapping data; an event can belong to more than one sliding window.

An example is showed in Fig. 2. The first window (*w1*, the green box) contains events that arrived between the zeroth and tenth seconds. The second window (*w2*, the orange box) contains events that arrived between the fifth and fifteenth

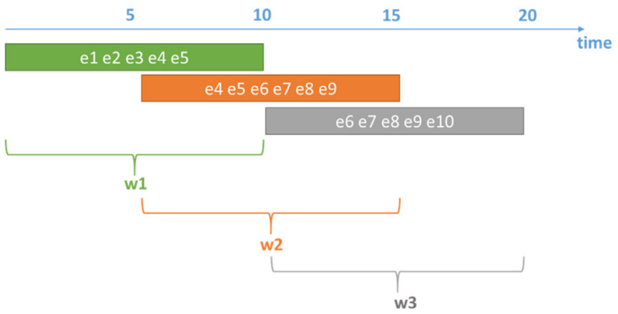

Fig. 2. A sliding windowing process.

Table 1
Web form input fields description.

Field name	Meaning	Value
L Parameter	the length of the signal to be analyzed, expressed in seconds	[0, 3600]
S Parameter	the slippage of the signal to be analyzed, expressed in seconds	[0, 3600]
Patient	the patient number on which to perform the analysis	[1, 21]
Phase	the phase of a recording to analyze	PreIctal, Ictal
Registration File	the number of EEG recording to analyze	Number of recording (it depends on Patient)
Fokus	the electrodes to analyze	[IN, OUT]
Bandwidth	the frequency bandwidth to analyze	B08= [8,12], B13=[13,20], B21=[21,30], B30=[30,45], B40=[40,70], B70=[70,120]
Univariate feature	the features of univariate type to be computed	See Table 3
Bivariate feature	the features of bivariate type to be computed	See Table 3
Bivariate calculation method	it indicates with respect to which reference signal to calculate the bivariate features	Wrt Previous L, Wrt Zero, Wrt Outfokus Electrodes

seconds. Note that events *e4* through *e5* are in both windows. When window w2 is evaluated at time $t = 15$ s, events *e1, e2* and *e3* are dropped from the event queue.

An example would be to compute the moving average of a stock price across the last five minutes, triggered every second.

3.4. Training Builder SW tool

The Training Builder is a software application for the massive extraction of features from time series, through which temporal analysis parameters and band-pass filters can be chosen. The idea behind this software tool is that, currently, there are not freely available standalone system or framework that, at the same time, provides efficient implementations of features extraction and data pre-processing techniques for time series data. Our software implementation tries to combine different approaches (features algorithms calculation, time series visualization and parallel computation) in order to facilitate the user in the time series data processing and preparation tasks.

The final purpose of Training Builder is to create training sets that will be the inputs to Analysis and Modeling following step. Given that, training sets can vary depending on:

- Time series (or better the recording of them).
- Temporal analysis parameters: *L* and *S*.
- Band-pass filters: [8,12,13,20], etc.
- Features to be computed (Hjorth Parameters, Statistical Moments, etc.).

Each training set consists of a csv (comma-separated values) file, where features are recorded as vectors.

The application was designed following the Client / Server architectural model, in which the Server part is composed by the algorithms for features calculation and other support utilities, while the Client part is composed by a browser-based application, responsible for visualizing output results and for input parameters selection. By using the responsive web-oriented graphical user interface, shown in Fig. 3, all the temporal parameters, bandwidths, input sources, and features can be selected by the user.

The implemented web form allows the user to choice all possible combination of input parameters for the Training Builder application. In the top right of the GUI, a help online button is provided to easily get a detailed description of each field.

The Table 1 shows the description of the selectable option fields, and their possible values. If $L = S$, in particular, two consecutive windows do not have overlapping areas (no points in common) and then we are dealing with tumbling windows.

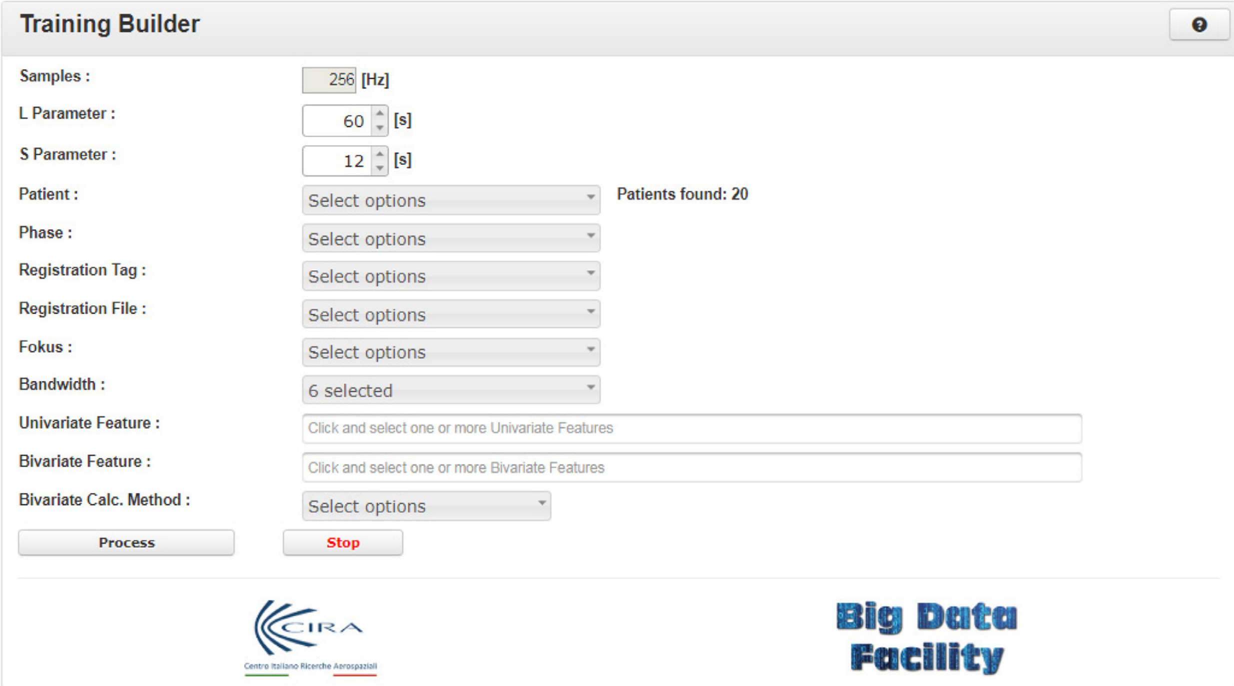

Fig. 3. Training Builder GUI.

Table 2
Temporal parameters of the Training Builder.

L	*S*
The length of the signal to be analyzed, in seconds 5 s	the slippage of the signal to be analyzed, in seconds 1 s

3.5. Time parameters

For the training of the classifier, the temporal parameters of the Training Builder are fixed, as reported in Table 2. So doing, every second the analysis focuses on the last five seconds of the selected time series.

3.6. Features calculation

Following the sliding window paradigm, every s seconds the previous L seconds of signal are analyzed and 26 features could be computed. Features could be extracted either from one signal (univariate) or from two or more signals (multivariate). In particular, bivariate features are based on a similarity measure that compares two time series objects and returns a value that encodes how similar the two objects are. Distance metrics are a kind of similarity measures commonly used to define if two time series are similar.

In Training Builder, feature algorithms are developed using parallel computation paradigm made available by Java threads technology. In our tests, based on an Intel™ multicore CPU (with Hyper-Threading technology on board), we have seen an almost linear performance speedup, by increasing the number of threads used.

Currently, all the implemented features have been divided into 7 classes and can be both Univariate (*U*) or Bivariate (*B*):

1. SM: Statistical Moments.
2. HP: Hjorth Parameters.
3. EB: Entropy Based.
4. CB: Complexity Based.
5. SE: Seismic Evaluators.
6. MC: Mutual Conditioned.
7. DB: Distance Based.

In Table 3 the list of all implemented features is reported where the last column provides also details about the programming language or the software environment used for implementation.

Table 3
Computed Features.

Id	Feature name	Code	U/B	Coding
1	Standard Deviation	SM1	U	Java
2	Variance	SM2	U	Java
3	Skewness	SM3	U	Java
4	Kurtosis	SM4	U	Java
5	Mean	SM5	U	Java
6	Hjorth mobility	HP1	U	Java
7	Hjorth complexity	HP2	U	Java
8	Shannon Entropy	EB1	U	Java
9	Log-Energy Entropy	EB2	U	Matlab
10	Kolmogorov Complexity	CB1	U	Matlab/C
11	Upper limit Lempel-Ziv complexity	CB2	U	Matlab/C
12	Lower limit Lempel-Ziv complexity	CB3	U	Matlab/C
13	Peak Displacement	SE1	U	Java
14	Predominant period	SE2	U	Java
15	Averaged period	SE3	U	Java
16	Squared Grade	SE4	U	Java
17	Squared time to peak	SE5	U	Java
18	Inverted Time to Peak	SE6	U	Java
19	Conditional Entropy	MC1	B	Java
20	Joint Entropy	MC2	B	Java
21	Mutual Information	MC3	B	Java
22	Cross correlation index	MC4	B	Java
23	Euclidean Distance	DB1	B	Java
24	Dynamic Time Warping	DB2	B	Java
25	Longest Common Sub-Sequence	DB3	B	Java
26	Levenshtein Distance	DB4	B	Java

A description of the more relevant implemented features is reported below.

3.6.1. Statistical Moments

In mathematics, a Statistical Moment (SM2-SM5) is a specific quantitative measure of the shape of a function. In our framework, the first four Statistical Moments have been calculated, plus standard deviation measure. All algorithms where developed in Java by using the Apache Commons Math library [31].

3.6.2. Hjorth's parameters

Hjorth's parameters [32] (HP1-HP2) (normalized slope descriptors) of mobility and complexity quantifies the root-mean-square frequency and the root-mean-square frequency spread of a given signal, respectively.

3.6.3. Shannon Entropy

In Information Theory, the Shannon's Entropy [33] (EB1) represents the average amount of information produced by a stochastic source of data. Formally, it is defined as the expected value of self-information. The latter represents the information contained in a given event x, emitted by the source X and it is defined as follows:

$$I(X) = -\log_2 p(x) \tag{3a}$$

Thus, the entropy of a source X turns out to be:

$$H(X) = E[I(X)] = E[-\log_2 p(X)] \tag{4}$$

where $p(X)$ is a probability mass function for a discrete random variable X.

3.6.4. Log-Energy Entropy

The Log-Energy Entropy [34] (EB2) is a feature closely related to Shannon's Entropy and to Wavelet Transform. In fact, after an appropriate wavelet decomposition, it is possible to calculate the Log-Energy Entropy by using the following relation:

$$E(S) = \sum_i \log_2 (s_i^2) \tag{5}$$

where s_i are the N coefficients of the wavelet transform for the signal S end the index i belongs to the interval [1, N].

3.6.5. Kolmogorov Complexity

In Algorithmic Information Theory, the Kolmogorov Complexity [35] (CB1) of an object, such as a piece of text, is the length of the shortest computer program (in a predetermined programming language) that produces the object as output. Thus, it is a measure of the computational resources needed to specify the object and it is also known as descriptive complexity.

3.6.6. Lempel-Ziv Complexity

The Lempel-Ziv Complexity [36] (CB2-CB3) of a given finite binary sequence is an index associated with the number of sub-sequences that can be identified. In particular, this process can take place through methods that tend to highlight the greater or lesser complexity of the given sequence. Therefore, taking into account the two extremes, it is possible to calculate those that are interpreted as the upper and lower limit of this index.

3.6.7. Conditional Entropy

Conditional Entropy (MC1) expresses the amount of information needed to describe a random variable y given the value of another random variable x:

$$CE = \sum_{x \in X} \sum_{y \in Y} p(x, y) \log \frac{p(x)}{p(x, y)} \tag{6}$$

3.6.8. Joint Entropy

Joint Entropy (MC2) is a measure of uncertainty associated with a set of random variables:

$$JE = -\sum_{x \in X} \sum_{y \in Y} p(x, y) \log p(x, y) \tag{7}$$

3.6.9. Mutual Information

Mutual Information (MC3) of two random variables is a measure of the mutual dependence between the two variables. It determines how similar the joint distribution $p(x,y)$ is to the products of factored marginal distribution p(y) and $p(x)$:

$$MI = \sum_{x \in X} \sum_{y \in Y} p(x, y) \log \frac{p(x)}{p(x)p(y)} \tag{8}$$

If x and y are independent, then x contains no information about y and vice versa, so their Mutual Information is zero.

3.6.10. Seismic Evaluators

The seismic features (SE1-SE6) have been calculated by considering [37], where the authors prove an analogy between earthquakes and epileptic seizures, and by considering [38] for their computation.

3.6.11. Dynamic Time Warping

Dynamic Time Warping [39] (DB2) is a technique that uses dynamic programming to compare two sequences of different lengths and allows non-linear alignments, one-to-many, or vice versa, thanks to a temporal distortion. A nonlinear (elastic) alignment produces a more intuitive measure of similarity and favours those cases in which the sequences are similar but locally out of phase.

3.6.12. Longest Common Sub-Sequence

The Longest Common Sub-Sequence [40] (DB3) is a variation of the edit distance. The basic idea is to match two sequences by allowing them to stretch, without rearranging the sequence of the elements but allowing some elements to be unmatched. The advantages of the LCSS method are twofold:

- Some elements may be unmatched, where in Euclidean and DTW all elements must be matched, even the outlier.
- The LCSS model allows a more efficient approximate computation, as will be shown later (whereas in DTW you need to compute some costly L_p distance.

3.6.13. Levenshtein Distance

The Levenshtein Distance [41] (DB4), also known as Edit Distance, is widely used in string-based matching problems, by counting the minimum number of operations (insertion, deletion and substitution) required to transform one string into the other. This distance, which has been widely applied for string matching sequences, can also be applied for time series case; we only need to replace the insertion, deletion and replacement costs of symbols with operation costs for numeric values.

3.7. Methods for bivariate features calculation

Bivariate algorithms are used to compute similarity distance between the current running signal and a “reference” signal. This reference signal could be of three different types:

1. Previous L: the same signal taken at a previous L interval (s seconds backwards).
2. Zero: the constant signal equal to 0.
3. Different Synchronous Signal: the synchronous signal happening in the same instant but originated from different electrodes.

Table 4
Target class distribution.

NO	YES
30,665	605

Table 5
Features distribution by count.

#	Feature	U/B	Count
1	Longest Common Sub-Sequence	B	31
2	Levenshtein Distance	B	31
3	Mutual Information	B	29
4	Join Entropy	B	27
5	Conditional Entropy	B	18
6	Dynamic Time Warping	B	16
7	Log-Energy Entropy	U	14
8	Shannon Entropy	U	13
9	Kurtosis	U	13
10	Kolmogorov Complexity	U	8
11	Euclidean Distance	B	8
12	Standard Deviation	U	4
13	Squared Grade	U	2
14	Peak Displacement	U	2
15	Inverted Time to Peak	U	1
Tot			217

3.8. Target class creation

Within ictal recordings, time period during which epileptologists identify seizures is also available.

The target class is instantiated with the value "NO" if the recording under investigation does not contain a seizure, with the value "YES" if it contains a seizure.

3.9. Final dataset

The final dataset has got 2377 features (2376 features + 1 target class) because:

$$2376 = (a + b * c + b * d) * e * f \tag{9}$$

Where:

- $a = 18$ (univariate features).
- $b = 8$ (bivariate features).
- $c = 3$ (bivariate modality calculation).
- $d = 3$ (type of reference signal).
- $e = 6$ (number of the bandwidths).
- $f = 6$ (electrodes: 3 InFokus + 3 OutFokus).

In this case study, we selected only one patient (patient 16) from the database because, thanks to a preliminary statistical analysis, it had a high number of seizures during its temporal monitoring and the seizures lasted longer than those of other patients. In particular, he had 6 epileptic seizures with an average duration of 121 s.

The final dataset has got 31,270 instances.

As shown in Table 4, the target class distribution is unbalanced, because the number of samples belonging to ictal recordings represent a small percentage of the final dataset.

3.10. Features selection

The number of input features can be reduced by applying a feature selection algorithm, for example by applying the following Information Gain (IG) formula:

$$\mathrm{IG}(\mathrm{Class}, \mathrm{Attribute}) = \mathrm{H}(\mathrm{Class}) - \mathrm{H}(\mathrm{Class}|\mathrm{Attribute}) \tag{10}$$

Where H is the Information Entropy.

We get that IG ϵ [0, 0.3657]. We choice to remove all attributes that have a score of less than 0.2. Thus, the number of features decreases from 2376 to 217.

The selected features are shown in Table 5 in descending order. We can see how the bivariate features are very high in the ranking and counting, and they have a high IG on average.

Table 6
Electrodes distribution by count.

#	Electrode	Count
1	E1	80
2	E2	76
3	E3	57
4	E5	4

Table 7
Bandwidths distribution by count.

#	Bandwidth	Count
1	B30	55
2	B70	47
3	B40	42
4	B21	34
5	B08	29
6	B13	10

Table 8
Distribution of seizures files.

Patient 16			
Training set		Test set	
Registration file	EEG phase	Registration file	EEG phase
0007	Preictal	0040	Preictal
0008	Ictal	0041	Ictal
0009	Preictal		
0026	Preictal		
0027	Ictal		
0042	Preictal		
0047	Preictal		
0048	Ictal		
0082	Preictal		
0083	Ictal		

The electrodes distribution of the selected features is reported in Table 6, where we can see that electrodes 4 and 5 are not present in the list. The features calculated from the signals recorded by the infokus electrodes have, as we expected, a higher IG value, due to their proximity to the epileptogenic areas.

Finally, Table 7 provides us the distribution of the selected features by bandwidth and we can see that in the low filtering frequencies less features are selected.

3.11. Cross-Validation and hold-out methods

For DM goal, we adopted the working strategy named hold-out method [19]. In this method, the original data with labeled examples is partitioned into two disjoint sets, called training and test sets, respectively. A model is induced from the training set and its performances, e.g. its precision, are evaluated on the test set. Moreover, the algorithms parameters are fixed by applying the cross-validation method [19] with 10 folds, thanks to which it is possible to train models in the presence of small datasets.

The training and the test sets are composed by different files of epileptic seizures. In particular, these registration files are divided in training and test sets by following the distribution reported in Table 8. Moreover, in order to overcome the class imbalance problem, the subset of the records labeled with “NO” of the training set has been undersampled, in order to be equal in number to the ones labeled with “YES”.

The training set has got 508 records with target class label “NO” and the same number for the records with target class label “YES”.

4. Modeling

Modeling follows Data Preparation; in Modeling phase detection models are trained by using Support Vector Machine algorithms and finally they are tested.

Table 9
SVMs Detailed Accuracy by Class.

N.	CONFUSION MATRIX		Accuracy	TP	FP	Precision	AUC	Class
F1	502	6	95.5709	0.988	0.077	0.928	0.996	N
	39	469		0.923	0.012	0.987	0.996	Y
F2	502	6	97.5394	0.988	0.037	0.964	0.975	N
	19	489		0.963	0.012	0.988	0.975	Y
F3	497	11	96.4567	0.978	0.049	0.952	0.99	N
	25	483		0.951	0.022	0.978	0.994	Y

Table 10
SVM Detailed Accuracy by Class on test set.

CONFUSION MATRIX		Accuracy	TP	FP	Precision	AUC	Class
4430	17	99.6259	0.996	0	1	0.998	N
0	97		1	0.004	0.851	0.998	Y

4.1. Support Vector Machine

Support Vector Machine (SVM) is the algorithmic technique chosen to realize the model for seizure detection. SVM is a widely used Machine Learning technique for supervised learning models. It is used for classification and regression analysis. A SVM training algorithm builds a model that assigns new examples into a class (or category) or the other. A good separation in SVM is achieved by the hyperplane that has the largest distance to the nearest training data point of any class [19].

Alternative approaches are available such as those base on manifold regularization [42–44].

4.2. Training by applying SVM algorithm

By analyzing the fraction of the EEG signal immediately preceding the beginning of the epileptic seizure and the seizure itself, a classifier is trained on the computed features, in order to detect the signals with seizure, by using the sliding window technique. The classifier based on SVMs, therefore, can be used to tag the time series of EEG signals.

By using the training set data, the SVM algorithm found a high dimensional surface that separated the two classes of signals, EEG signal with seizure and EEG signal without seizure.

For the training of SVMs, the LibSVM library was used in the Matlab environment and three types of kernel (linear, Gaussian, polynomial) were implemented and compared using the evaluation metric. These 3 types of kernels are dependent on a different number of hyper-parameters: the linear kernel from the regularization parameter C, the Gaussian kernel from C and gamma parameters, and finally the polynomial kernel from C, gamma and the p degree.

In order to select the best kernel hyper-parameters (C, gamma, and p), a search algorithm was applied using multi-dimensional grids, depending on the type of kernel, and using the graphics of performances (Accuracy and Area Under the ROC Curve).

In Table 9 the performances of three SVM-based classifiers are shown, varying the three types of kernel, using the cross-validation method and the balanced training set. The F1 model was obtained with a linear kernel, the F2 model with a Gaussian kernel, and the F3 classifier with a polynomial kernel.

For the following testing phase, we chose the F2 model, which showed the best performances.

4.3. Model testing and errors analysis evaluation

The Table 10 shows the detection model performances on the test set formed by recordings 40, belonging to the preictal phase and thus not containing a seizure and 41, belonging to ictal phase and thus containing a seizure.

The low amount of false positives demonstrates the high capacity of the binary classifier to discriminate between the elements coming from the preictal recordings and those coming from the ictal ones.

The Fig. 4 shows the trend of the E1-B30-LE feature; "E1-B30-LE" is a compact name which indicates that the variable refers to Log-Energy Entropy (LE), of the signal recorded by the electrode number 1 (E1), computed in the frequency band 30–45 Hz (B30). The red circles indicate the "YES" records (that come from the ictal signal), while the blue points indicate the "NO" records (that come from the preictal signal). Furthermore, the blue squares represent 13 false positives (false alarms), and they are very close and immediately before the true positives (hit rates).

In the chart of Fig. 4, all the false positives are close to the incipit of the epileptic seizure and are also consecutive: therefore, could the SVM have a predictive nature? The answer to this question will certainly be a subject for a future investigation.

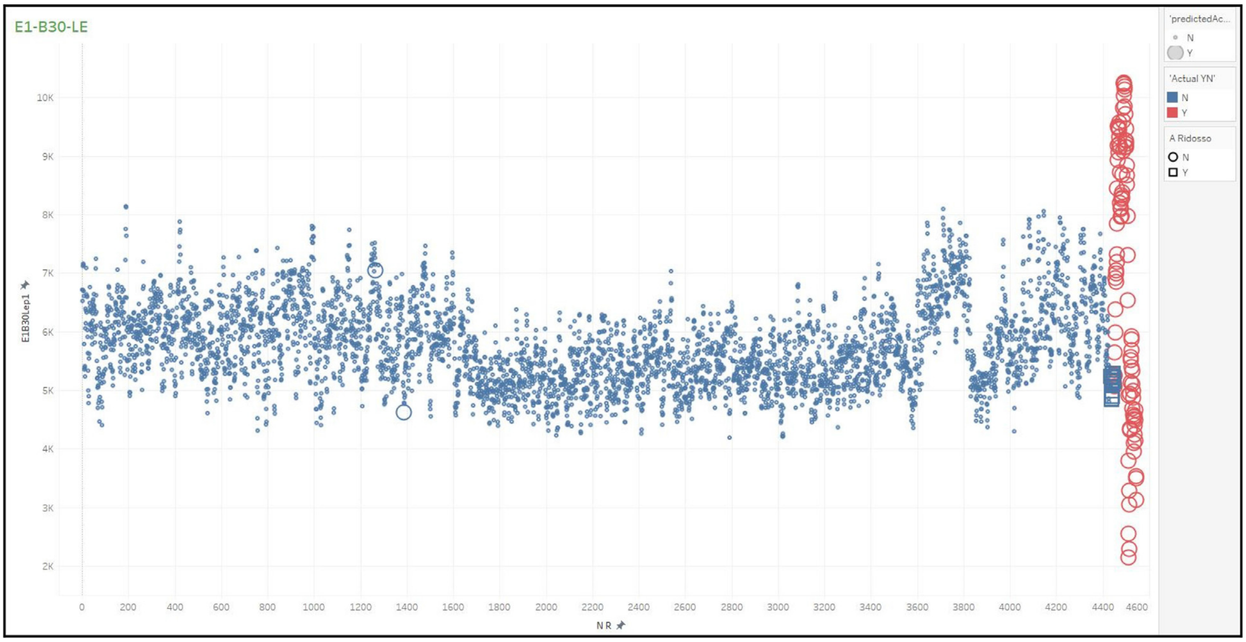

Fig. 4. Trend of the E1-B30-LE variable and classification errors.

5. Conclusion and future works

In this paper, an automated tool for seizure detection in EEG signals has been proposed with the aim of classifying and identifying epileptic seizures and seizure-free signals.

The trained SVM-based model is able to classify with high accuracy the instances of the test file consisting of records from both seizure-free and epileptic EEG signals. The high performances achieved by the classifier are due to the ability of the calculated features to describe EEG signals, to the feature selection process using the Information Gain measure, and certainly to the specificity of the model for the selected patient. In fact, this system allows channel selection and feature extraction for a single patient, the further step will be to test the model on the entire dataset in order to demonstrate the independence of the model from the specific patient and its feasibility for different patients and different epilepsy types. Anyway, the high number of features and also the use of a big frequency spectrum give us reason to hope that it can deal with different patients and different epilepsy types. The goal will be to develop a unique model able to identify the onset of the seizures for all the patients and for all the different kinds of epilepsy. In fact, this can have a high translational value, not only because it will be a valuable tool for epileptologists to speed up the process of seizure detection, but, more important, because this method can be embedded in a closed loop system that can detect seizures in real time and abort seizures. For this real-time application, it is also important to have a method with a low computational cost, so further development will include also the improvement of feature selection step, because the a priori knowledge about the optimal feature subset can provide a lower computational cost during feature extraction and a subsequent lower computational cost for the classification stage.

Furthermore, the obtained model may be expanded by training a classifier, or an ensemble of classifiers, capable of tagging the seizures of different patients. Finally, future studies should focus on testing these algorithms online on continuous unlabeled data.

As future works, we are going to integrate in our tool some representation techniques that can reduce the dimensionality of time series. Representation techniques have been proven to limit time and memory consuming, especially when a similarity distance between time series has to be found. Approaches based on Internet of Things frameworks could be used to collect data [45,46]. Moreover, we plan to extend our framework to other feature algorithms implementation and to move towards one of the new emerging massive data processing framework, e.g. Apache Spark [47] or Apache Flink [48].

Finally, the proposed methodology and techniques, including the Training Builder software tool and the sliding window paradigm, can also be used in other domains and for other objectives. Mainly, since an epileptic seizure can be viewed as an anomaly with respect to the predominant behavior of the signal, the developed platform can be used to discover outliers, or sequences of outliers, within time series recorded by sensors of any kind. Moreover, the software platform has been developed to be extensible, so new algorithms for features calculation can be easily added. Surely, in order to adapt the platform to new source types, an upgrade and a customization of the tool are needed. Additionally, the proposed analytical approach constitute a base of support for developing mathematical models in many industrial applications, such as the diagnostics of complex systems (IT, avionics, aerospace, etc.), the detection of fraudulent attacks, financial and otherwise, and so on.

We think that the advantage of our method, compared with other seizure detection methods, is represented from the use of many features, both bivariate and univariate, which can give us reason to believe that it can deal with different patients and different epilepsy types. Furthermore, the developed software tool can help us to understand how to calibrate the different parameters of the various steps of the algorithm to achieve better results.

Acknowledgment

The authors would mention the Big Data Facility project, funded by the Italian PRORA, in which the tool has been designed and developed, and data analysis has been performed. Moreover, S.C., G.Z., and G.T. would like to thank Dr. Amalia Spera for her support in some numerical experiments. S.C. and G.T thank the GNCS-INDAM to partially support this paper.

References

[1] World Health Organization, 2018, http://www.who.int/news-room/fact-sheets/detail/epilepsy.
[2] C.E. Elger, F. Mormann, Seizure prediction and documentation–two important problems, Lancet Neurol. 12 (2013) 531–532, doi:10.1016/s1474-4422(13)70092-9.
[3] P. Chapman et al., "CRISP-DM 1.0. Data Mining guide," 2000, http://www.citeulike.org/group/1598/author/Chapman:P
[4] F.C. Morabito, et al., Multivariate multi-scale permutation entropy for complexity analysis of Alzheimer's disease EEG, Entropy 14 (2012) 1186–1202.
[5] J.R. Huang, et al., Application of multivariate empirical mode decomposition and sample entropy in EEG signals via artificial neural networks for interpreting depth of anesthesia, Entropy 15 (2013) 3325–3339.
[6] H. Ocak, Automatic detection of epileptic seizures in EEG using discrete wavelet transform and approximate entropy, Expert Syst. Appl. 36 (2009) 2027–2036.
[7] P.S. Kumar, N. Sriraam, P.G. Benakop, B.C. Jinaga, Entropies based detection of epileptic seizures with artificial neural network classifiers, Expert Syst. Appl. 37 (2010) 3284–3291.
[8] Y. Song, J. Crowcroft, J. Zhang, Automatic epileptic seizure detection in EEGs based on optimized sample entropy and extreme learning machine, J. Neurosci. Methods 210 (2012) 132–146.
[9] N. Nicolaou, J. Georgiou, Detection of epileptic electroencephalogram based on Permutation Entropy and Support Vector Machines, Expert Syst. Appl. 39 (2012) 202–209.
[10] E.D. Ubeyli, Lyapunov exponents/probabilistic neural networks for analysis of EEG signals, Expert Syst. Appl. 37 (2010) 985–992.
[11] A. Temko, G. Boylan, W. Marnane, G. Lightbody, Robust neonatal EEG seizure detection through adaptive background modeling, Int. J. Neural Syst. 23 (4) (2013).
[12] Y.U. Khan, J. Gotman, Wavelet based automatic seizure detection in intracerebral electroencephalogram, Clin. Neurophysiol. 114 (2003) 898–908.
[13] D. Chen, S. Wan, J. Xiang, F.S. Bao, A high-performance seizure detection algorithm based on Discrete Wavelet Transform (DWT) and EEG, PLoS One 12 (2017) e0173138.
[14] L. Guo, D. Rivero, J. Dorado, J.R. Rabuñal, A. Pazos, Automatic epileptic seizure detection in EEGs based on line length feature and artificial neural networks, J. Neurosci. Methods 191 (1) (2010) 101–109.
[15] S. Yuan, W. Zhou, Q. Yuan, Y. Zhang, Q. Meng, Automatic seizure detection using diffusion distance and BLDA in intracranial EEG, Epilepsy Behav. 31 (2014) 339–345.
[16] A. Aarabi, R. Fazel-Rezai, Y. Aghakhani, A fuzzy rule-based system for epileptic seizure detection in intracranial EEG, Clin. Neurophysiol. 120 (2009) 1648–1657.
[17] F. Rabbi, R. Fazel-Rezai, A fuzzy logic system for seizure onset detection in intracranial EEG, Comput. Intell. Neurosci. 2012 (2012) ID705140.
[18] S.M. Usman, M. Usman, S. Fong, Epileptic seizures prediction using machine learning methods, Hindawi Comput. Math. Methods Med. 2017 (2017) 10 Article 9074759pages.
[19] P. Tan, M. Steinbach, V. Kumar, Introduction to Data Mining, Pearson Addison Wesley, 2005.
[20] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I.H. Witten, The WEKA data mining software: an update, SIGKDD Explor. 11 (Issue 1) (2009) 11–18.
[21] J. Han, M. Kamber, Data Mining: Concepts and Techniques, Morgan Kaufmann Publishers, 2001.
[22] I.H. Witten, E. Frank, Data Mining. Practical Machine Learning Tools and Techniques, Morgan Kaufmann, 2005.
[23] A. Chianese, F. Marulli, F. Piccialli, I. Valente, A novel challenge into multimedia cultural heritage: an integrated approach to support cultural information enrichment, in: Proceedings of the International Conference on Signal-Image Technology and Internet-Based Systems, SITIS, 6727195, 2013, pp. 217–224.
[24] F. Piccialli, J.E. Jung, Understanding customer experience diffusion on social networking services by big data analytics, Mobile Netw. Appl. 22 (4) (2017) 605–612.
[25] FSPEEG WebsiteSeizure Prediction Project Freiburg, University of Freiburg, 2018 http://epilepsy.uni-freiburg.de/freiburg-seizure-prediction-project/eeg-database/ (Accessed date: 8 August 2018).
[26] Freiburger Zentrum fur Datenanalyse und Mollbildung: The Freiburg seizure prediction project. https://epilepsy.uni-freiburg.de/freiburg-seizure-prediction-project/eeg-database.
[27] J.G. Proakis, D.G. Manolakis, Digital Signal Processing, 4th Edition, Prentice Hall, 2006.
[28] J. Gama, Knowlwdge Discovery from Data Streams, Chapman & Hall/CRC, 2010.
[29] M. Last, A. Kandel, H. Bunke, Data Mining in Time Series Databases, World Scientific Publishing Co. Pte. Ltd., 2004.
[30] M. Datar, A. Gionis, P. Indyk, R. Motwani, Maintaining Stream Statistics over Sliding Windows, in: Proceedings of the 2002 Annual ACM-SIAM Symp. on Discrete Algorithms, 2002, pp. 635–644.
[31] "Commons Math: The Apache Commons Mathematics Library," http://commons.apache.org/proper/commons-math/, September 2018.
[32] B. Hjorth, EEG analysis based on time domain properties, Electroencephalogr. Clin. Neurophysiol. 29 (3) (1970) 306–310.
[33] D. Phung, D. Tran, W. Ma, P. Nguyen, T. Pham, Using Shannon entropy as EEG signal feature for fast person identification, in: Proceedings of the European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, 2014.
[34] S. Aydin, H.M. Saraoğlu, S. Kara, Log energy entropy-based EEG classification with multilayer neural networks in seizure, Ann. Biomed. Eng. 37 (12) (2009) 2626–2630.
[35] G. Xu, J. Wang, Q. Zhang, J. Zhu, An epileptic seizure prediction algorithm from scalp EEG based on morphological filter and Kolmogorov complexity, in: Proceedings of the First International Conference on Digital Human Modeling, 2007.
[36] Y. Zhang, S. Wei, C. Di Maria, C Liu, Using Lempel–Ziv complexity to assess ECG signal quality, J. Med. Biol. Eng. 36 (5) (2016) 625–634.
[37] I. Osorio, H.P. Zaveri, M.G. Frei, S. Arthurs, Rationales for analogy between earthquakes, financial crashes, and epileptic seizures, Epilepsy: the Intersection of Neurosciences, Biology, Mathematics, Engineering, and Physics, CRC Press, Taylor & Francis Group, 2011.

[38] G. Zazzaro, F.M. Pisano, G. Romano, Bayesian networks for earthquake magnitude classification in a early warning system, Int. J. Environ. Chem. Ecol. Geol. Geophys. Eng. 6 (4) (2012) 152–162.
[39] D. Berndt, J. Clifford, Using dynamic time warping to find patterns in time series, in: AAAIWS'94 Proceedings of the Third International Conference on Knowledge Discovery and Data Mining, 1994, pp. 359–370.
[40] M. Vlachos, D. Gunopulos, G. Kollios, Discovering similar multidimensional trajectories, in: Proceedings of the ICDE, 2002, pp. 673–684.
[41] R.A. Wagner, M.J. Fischer, The string-to-string correction problem, J. ACM 21 (1) (1974) 168–173.
[42] M. Viola, M. Sangiovanni, G. Toraldo, M.R. Guarracino, A generalized eigenvalues classifier with embedded feature selection, Optim. Lett. 11 (2) (2017) 299–311.
[43] M. Belkin, P. Niyogi, V. Sindhwani, Manifold regularization: a geometric framework for learning from labeled and unlabeled examples, J. Mach. Learn. Res. 7 (2006) 2399–2434.
[44] M. Viola, M. Sangiovanni, G. Toraldo, M.R. Guarracino, Semi-supervised generalized eigenvalues classification, Ann. Oper. Res. (2017) 1–18.
[45] F. Amato, V. Moscato, A. Picariello, F. Piccialli, SOS: a multimedia recommender System for Online Social networks, Future Gener. Comput. Systems 93 (2019) 914–923.
[46] F. Piccialli, A. Chianese, The internet of things supporting context-aware computing: a cultural heritage case study, Mobile Netw. Appl. 22 (2) (2017) 332–343.
[47] M. Zaharia, An architecture for fast and general data processing on large clusters, Morgan & Claypool, 2013.
[48] P. Carbone, S. Ewen, S. Haridi, A. Katsifodimos, V. Markl, K. Tzoumas, Apache Flink: stream and batch processing in a single engine, Bull. IEEE Comput. Soc. Tech. Comm. Data Eng. 36 (4) (2015) 28–38.