Neural Computing and Applications (2020) 32:11111–11127
https://doi.org/10.1007/s00521-018-3925-z

COMPUTER AIDED MEDICAL DIAGNOSIS

A new EEG software that supports emotion recognition by using an autonomous approach

Roberto Munoz[1,2] · Rodrigo Olivares[1] · Carla Taramasco[1,2] · Rodolfo Villarroel[3] · Ricardo Soto[3] · María Francisca Alonso-Sánchez[4] · Erick Merino[1] · Victor Hugo C. de Albuquerque[5]

Received: 6 October 2018 / Accepted: 29 November 2018 / Published online: 6 December 2018

Abstract

Human behavior is manly addressed by emotions. One of the most accepted models that represent emotions is known as the circumplex model. This model organizes emotions into points on a bidimensional plane: valence and arousal. Despite the importance of the emotion recognition, there are limited initiatives that seek to classify emotions easily in an uncontrolled environment. In this work, we present the architecture and the design of an extensible software which allows recognizing and classifying emotions by using a low-cost EEG. The proposed software implements an emotion classifier although a support vector machines (SVM) are boosted with an autonomous bio-inspired approach. The contribution was experimentally evaluated by taking a set of well-known validated EEG Databases for Emotion Recognition. Computational experiments show promising results. Using our proposal for EEG emotion classification, we reach an accuracy close to 95%. The results obtained confirm that our approach is able to overcome to a commonly used SVM classifier and that the proposed software can be useful in real environments.

Keywords Emotion recognition · Software architecture · Support vector machine · Autonomous bat algorithm

1 Introduction

A clear emotional influence on the learning process is demonstrated, this positively or negatively affecting certain processes, such as cognitive mechanisms of storage and recovery of information, information processing, and motivation mechanisms, among others [1–3].

Many forms of detecting emotional states exist. However, research into how emotions affect the learning process is principally conducted through standardized questionnaires and methods of low precision [1]. In spite of this, the research and development of easy-to-use tools that permit studying emotions within diverse environments is still scarce.

✉ Rodrigo Olivares
rodrigo.olivares@uv.cl

Roberto Munoz
roberto.munoz@uv.cl

Carla Taramasco
carla.taramasco@uv.cl

Rodolfo Villarroel
rodolfo.villarroel@pucv.cl

Ricardo Soto
ricardo.soto@pucv.cl

María Francisca Alonso-Sánchez
mariaalonsosa@santotomas.cl

Erick Merino
erick.merino@postgrado.uv.cl

Victor Hugo C. de Albuquerque
victor.albuquerque@unifor.br

[1] Escuela de Ingeniería Civil Informática, Universidad de Valparaíso, Valparaíso, Chile

[2] Centro de Investigación y Desarrollo en Ingeniería en Salud, Universidad de Valparaíso, Valparaíso, Chile

[3] Pontificia Universidad Católica de Valparaíso, Valparaíso, Chile

[4] Escuela de Fonoaudiología, Facultad de Salud, Universidad Santo Tomás, Viña del Mar, Chile

[5] Graduate Program in Applied Informatics, Laboratory of Bioinformatics, University of Fortaleza, Fortaleza, CE, Brazil

Currently, more than 150 proposed theories exist in psychology literature that explain the how and the why behind human emotions. In spite of this large quantity, and in spite of the numerous emotional models that exist, they are all constructed under five focuses: (1) dimensional, (2) anatomical, (3) rational, (4) communicational, and (5) evaluative [4]. Dimensional models (also called constructive models) do not categorize emotions of discrete states (sadness, happiness, etc.); rather, they identify a determined emotional state as a point in a space constructed by central affections (i.e., positive, negative, encourage, etc.) [5]. One of the most popular dimensional models is the circumplex model created in 1980 by James Russell. This model organizes emotions as point in a plane composed of the dimensions "valence" (pleasurable or not pleasurable) and "agitation" (tension, relaxation). According to his findings, emotions are organized in a circular form in this plane [6].

Among the forms commonly used in researching the detection of emotions are those that utilize electroencephalogram (EEG) signals [7, 8]. EEG is the measurement of activation of neuronal populations involved in various levels of processing (from sensory input to emotion processing) by recording voltage differences or frequency over time. The frequency analysis of the electrophysiological signals allows estimating the oscillating patterns of specific neuronal groups and their implication in cognitive processes. However, the use of EEG signal manipulation software is complex for non-expert users. Furthermore, this type of analysis is usually done off-line because of the need for noise elimination preprocessing, which can be done manually or automatically [9].

Currently, there is a lack of software to classify emotional data in a simple way. The absence of this type of initiatives is a additional difficulty for the public without expertise in the area, for example, educators.

In this work, we present the design and implementation of software that allows to detect and to classify emotions based on EEG. For that, we designed an extensible architecture to develop a software for the recognition of emotions. The proposed software uses a classification model based on an approximate technique known as bat algorithm. We use an improved version of this algorithm that allows us adjusting its search space during the run. The main idea is to use this approximate method to optimize the input parameters of the SVM multiclass in order to reduce the time it takes to find a good solution. The software was validated using three different emotion databases: DEAP, MAHNOB HCI Tagging, and, eNTERFACE, achieving a precision of 95%, 95%, and 97%, respectively.

The paper continues as follows. Section 2 presents the theoretical support. In Sect. 3, we expose some relevant works. Section 4 presents the proposed software for emotion recognition. We present in Sect. 5 our autonomous approach for emotion recognition based on bat algorithm. In Sect. 6, we present and discuss the main results of our experiments. Finally, Sect. 7 exposes our conclusions and future work.

2 Theoretical support

We present, in Sect. 2.1, a theoretical framework in respect of emotions, and Sect. 2.2 presents an overview about EEG.

2.1 Emotions

Anatomical models link emotions to certain sectors/circuits in the brain [10]. These models are governed by biological and brain theories [5]. On the other hand, rational models seek to identify the functions or roles that emotions facilitate, and are used mainly in areas related to artificial intelligence [5].

Models of communication are inspired by social theories and describe emotions such as behaviors in relation to approaching or getting away from other people or agents [5]. The evaluative models of the emotions describe these as a consequence of the evaluation of the context, the situation, or the surroundings. These situations generate behaviors that in turn configure different situations, creating a recursive cycle of feedback [5].

One of the most accepted models that represent emotions is known as the circumplex model [6]. This model organizes emotions into points on a bidimensional plane made up of the following dimensions: "valence" (pleasurable or not pleasurable) and "arousal" (tension, relaxation); in this way, emotions are organized in a circular manner within this plane [6]. A representation of the circumplex model is shown in Fig. 1.

There are variants of the circumplex model in which extra dimensions are added, such as domination or freedom within a situation. However, it has been proven that this dimension captures the consequences believed by the person regarding the emotion, and not the emotion itself [11].

Emotions can cognitively affect in three ways. These are the storage and recuperation of information, strategies of information processing, and the limit to the attention of resources [1].

Positive and negative emotions are conducive to dependent learning and congruent recuperation of the emotional state [12]. This signifies that the material to be learned is learned in conjunction with the emotional state that a person exhibits in that moment. Therefore, the learned material will be remembered in the most efficient

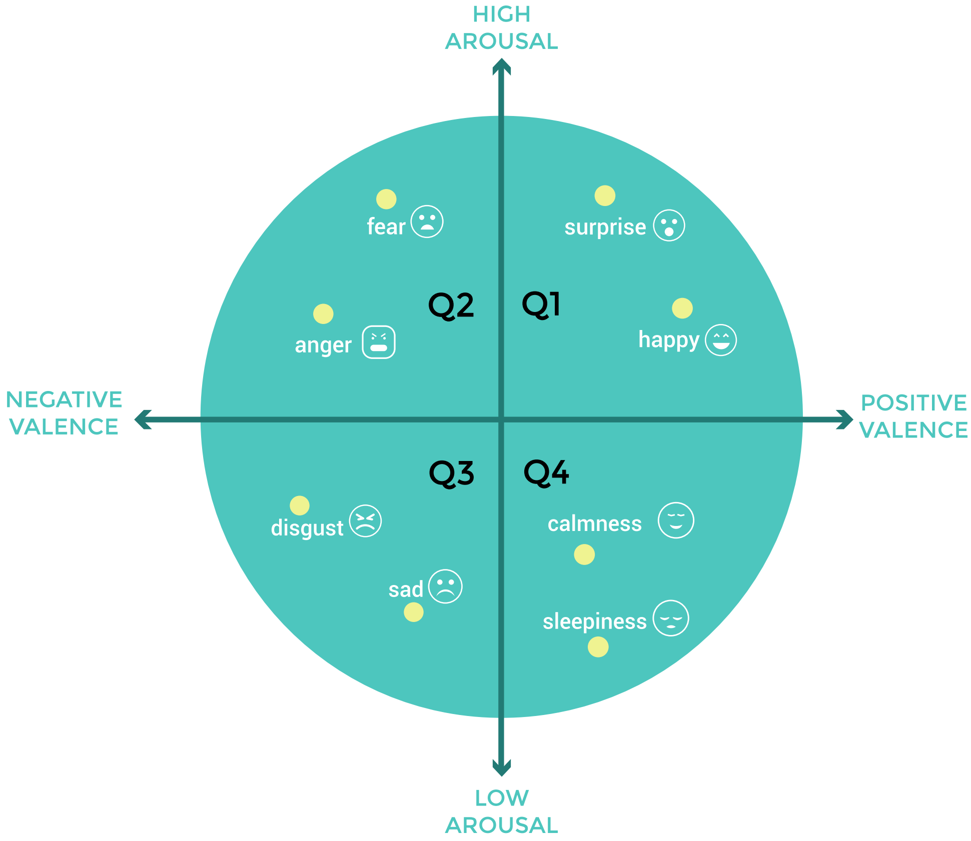


Fig. 1 Russells classification model

manner if the emotion that occurred at the moment of storage of said material is also present [1].

Moreover, strategies of information processing are measured by positive and negative emotions. Positive emotions provide a mode of "holistic" thinking that privileges creative manners of thinking and playing with unusual ideas.

On the other hand, negative emotions promote a mode of analytical, detailed, and focused thinking [1], such as determining the resolution to an arithmetical problem in an algorithmic manner.

Finally, the limit to the attention of resources is a very relevant factor upon analyzing the academic success or failure of a student, since associated emotions promote a high or low attention to resources that an environment provides [1].

Positive emotions promote a high attention to the resources of an environment; however, this can become a problem if too much irrelevant thinking is generated (i.e., falling in love) [7]. Conversely, negative emotions limit the attention to resources. Attentional interpretations postulate that anxiety consumes the resources of memory in the cerebrum and therefore limits the remaining capacity [13]. This is particularly important, evolutionary speaking, in situations of danger, such as escaping from wild beasts or running from an enemy.

On the other hand, the effect of emotions on learning can be categorized into two types: (1) according to their orientation and (2) according to their experienced subjective value. The orientation of an emotion in a learning context can be task-oriented or related to the social context [1].

Emotions related to the task can occur at three different times: in the execution of the task (enjoyment, boredom), before the task (hope, anxiety), or after the task (pride, shame). Regarding the subjective value of emotions, these are classified as the valence and can be either positive (related to the experience), negative, or unpleasant [1].

2.2 Electroencephalography (EEG)

EEG is a measurement of postsynaptic brain potentials recorded by sensors on the scalp. That is, when multiple pyramidal neurons are activated, this generates a local current product of the excitation of the dendrites. Thus, the amplitude of the EEG is the sum of electrical potentials of large populations of pyramidal neurons that generate a dipole between the soma and the dendrites [14].

The sensors are electrodes located on the surface of the pericranium that, with the aid of a conductive gel, allow detecting and directing the voltage to an amplifier (sometimes the amplifier is integrated into the electrodes). The electrodes are generally positioned in accordance with the international system 10–20, which is distributed with a 10% and 20% separation at its latitude and longitude, respectively. In particular, the equator is defined by the nasion (Nz corresponds to the depression between the eyes), and the inion (Iz corresponds to the bulge of the nape) and the right and left preauricular points. Thus, the length is defined between Nz and Iz and latitude by the ten separations of these. Although there are other systems (i.e., 10–10 or radial), these should always be referenced based on the 10–20 system. Finally, EEG systems present a problem of ground circuit noise and this is resolved using differential amplifiers. In other words, the difference between reference electrodes (in mastoid or earlobe), active electrodes, and Earth electrodes is calculated [15].

Furthermore, several other techniques exist to interpret this measurement, one of them being the event-related potential (ERP), which explores the voltage of temporal segments measured reiteratively after presenting a stimulus of interest. This technique is widely used for perceptual study and cognitive functions such as attention, memory, or language. In this way, a component generated by averaging numerous trials associates a latency and polarity with a cognitive process (i.e., P3 or N400). In effect, there are two components that have been associated with the emotions, the early posterior negativity located in the visual cortex around the 200 ms established mainly by stimuli with positive valence [16] and the late positive potential associated with the emotional relevance of the stimulus with appearance in the parietal zone around 300 ms [17].

Furthermore, other techniques such as event-related oscillations (ERO) or quantitative EEG (QEEG) permit characterizing the dynamics of frequency or voltage over time. The frequency study of brain activation depends on the integration of numerous sensory, perceptual, and cognitive processes. Because of this, it is currently one of the most popular techniques for studying social neuroscience.

The brain waves measured in EEG are integrated by various oscillating subcomponents and their subsequent breakdown, permitting the isolation of the activity of specific neuronal populations that have been recruited for a particular process. These are calculated from peak to peak, are usually between 0.5 and 100 v of amplitude, and are classified according to their frequency. Frequency bands are classified mainly into four groups: delta (0.5–4 Hz), theta (4–8 Hz), alpha (8–13 Hz), beta (> 13 Hz), and gamma (30–80 Hz). The methods to represent the frequency activity, for example, the Fourier transform or wavelet, perform a mathematical equivalence that allows to determine the necessary power of each band to rebuild a signal and only in some cases. For example, the gamma band between 400 and 800 ms presents a physiological equivalence [15].

2.3 Support vector machines (SVMs)

Support vector machines (SVMs) had been considered as a set of supervised learning algorithms based on statistics learning theory [18]. A SVM works in *d*-dimensional space, where *d* is the number of dimensions of the feature vector (8 in this work). This choice allows to adjust the features to a defined kernel space, Gaussian for example. Later, in the create process for a multiclass SVM, we apply the one-against-all method. This method tries to define a set of binary classifiers (hyperplanes), separating each class from another, according to [19].

SVM provides a model that can be generalized principally for nonlinear feature spaces. Finally, the create process ends when the SVM reaches a efficient configuration for their parameters C and γ. This setting enables accommodation to outliers and allows errors on the training set.

3 Related work

Currently, there are several contributions by using electrophysiological signals for the classification of emotions. For example, in [20] the authors propose a method to recognize emotions via EEG signals. In this way, they detect the patient's emotional states in an assisted living environment. The authors decompose the EEG signal into five frequency bands by using the wavelet transform. The average classification accuracy obtained by the SVM classifier was 83.87% for four types of emotions. In [21], works with EEG signals and emotion recognition. This manuscript presents an average accuracy of 73.0% by using EEG features to classify two emotional states. Another relevant contribution was proposed in [22]. The authors use SVM and K-nearest neighbor classifiers to detect the emotional states from the EEG features. The best classification precision achieved by the proposal is 84.05% for arousal and 86.75% for valence. In [23], empirical mode decomposition (EMD) and the sampled entropy for the generation of a classifier are used. These approaches allow to build an initial multiclass SVM classifier only using channels (F3 and C4). In [24], we improve the [23] proposal by using the black hole algorithm to optimize the feature vector of the SVM. Using MAHNOB HCI Tagging database, we obtained an accuracy of 92.56%.

4 Proposed software for emotion recognition

We have defined a highly extensible architecture (following the 4+1 view model proposed by [25]). This is aimed at developing a highly extensible application that can integrate multiple low-cost sensors. This generation was based on what was developed in [26], and we have extended our work to integrate two new types of EEG (OpenBCI[1] and MindWave[2]). Furthermore, the high-level data flow following the application is illustrated in Fig. 2.

First, data are captured either from the Kinect sensor (given that it inherits the capabilities of [26]), or the EEG OpenBCI or MindWave. Later, in the application, these data are processed through various training models, in this case autonomous search (AS) for the detection of emotions, to then be able to be recorded and classified either in a centralized database (MySQL, PostgreSQL) or in portable files which can be analyzed by other types of tools.

Furthermore, the developed application possesses three types of modules (input modules, training modules, and general modules) (see Fig. 3):

In the case of the input modules that record the entire scene, they can be enabled directly from the menu. However, for the input modules for a specific person, first a person must be associated with a scene, creating one or associating with one that already exists. Then, from the modules menu, it is necessary to specifically activate each person's modules separately, as well as configure them independently if additional configuration is needed.

Once the input sensors are enabled (a process that should be completed prior to initiating recording), the sensor will remain open in "Monitor" mode, in which live data are shown, but are not recorded, stored, or processed. The processing models should also be enabled from the "Modules" menu before starting the recording; no information is collected or processed until recording is initiated.

Once recording is initiated, View brings up the Recorder controller component, which iterates the input modules by activating the Record() method of each of the modules. These types of data are interfaces contained in the project's API package; in this way, each module "knows" the process modules that exist, and, rather, grants access to those via a list of objects through the "DataAccessFacade," a property that all of the modules exhibit that are assigned using dependency injection.

In general terms, upon initiating the recording, the Recorder controller component orders the distinct input modules (via the input module facade) to capture the scene files (for reproduction later on) within the corresponding folder. Furthermore, the component also sends the data generated to the processing modules compatible with their corresponding types of data. These modules, which are inherited from a common class, are accessed through the processing module facade and implement an interface for each input module with which they are compatible.

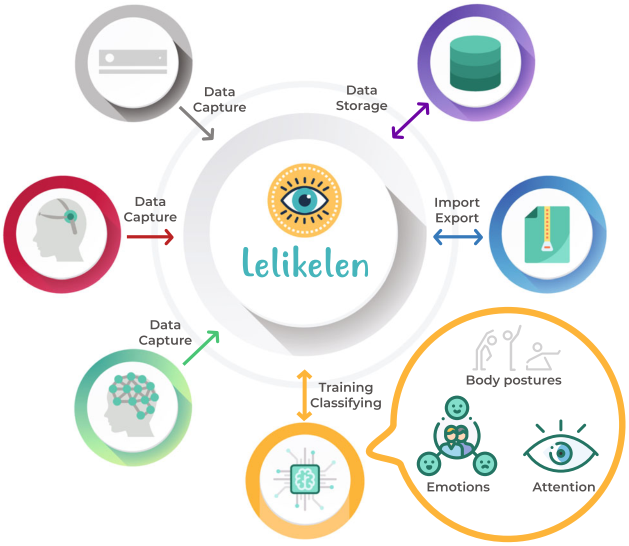


Fig. 2 High-level data flow scheme

These modules complete a specific process and store the data in main memory via data management and validations, which, after stopping recording, store the data in the database through a Data Access Object (DAO) component. We designed this in order to optimize the execution given the large quantity of data generated in real time.

After recording is stopped, the scene is stored in a common database, remaining in playback mode. Upon initiating playback through *PlayerController*, the data from the scene are obtained from the *PlayerController* of each input module through *InputModuleFacade*; these data are sent to View. View supports a video player and images (independent of the technology) of two overlapping layers, and to which each input module can add specific tabs to show their data types; this is known as injecting views.

View can also invoke the *FileController* driver to export and import external files, which communicates with the data validation and management module for obtaining and inserting from or to the database.

Alternatively, a third type of module exists. These are general modules that can be active at any given time of execution, regardless of the current state of the application.

It is important to emphasize that the input modules do not insert data into the database and only read data concerning the configuration directories in order to maintain a highly consistent structure in terms of roles and responsibilities. This means that input modules do not interfere with the data model or the rest of the application; rather, they only capture data and provide them. Processing

[1] https://openbci.com/.

[2] http://neurosky.com/biosensors/eeg-sensor/biosensors/.

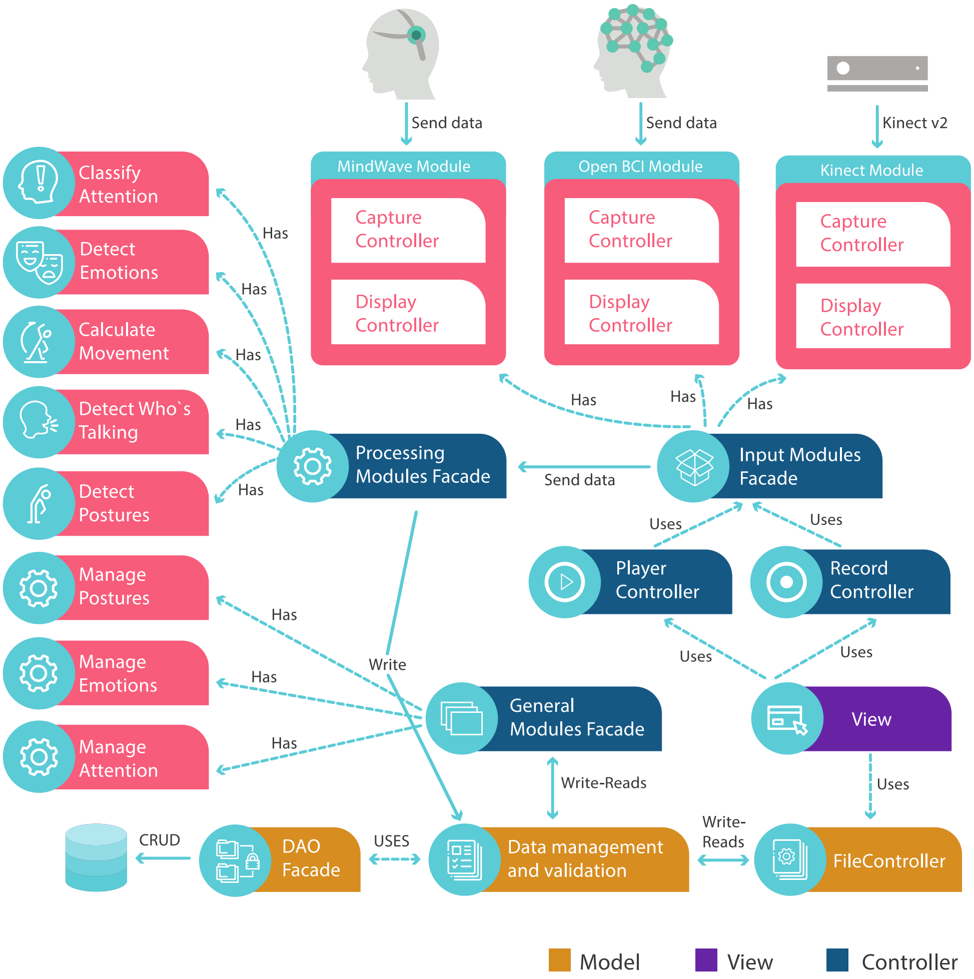


Fig. 3 Orchestration of main services

modules write modal data (intervals, charts, and fixes), but do not delete or modify data. This was done to avoid corrupted data, because as mentioned above, these data will be stored in main memory while they are being captured and will only be stored in the chosen database chosen after the recording is complete.

The general modules can perform four basic operations to the database (create, read, update, and delete). These modules serve to extend the application and do not depend on the state of it, but can always run. These are useful for tasks such as training models and other ancillary works. This way of accessing the data is assured because each class of the data management and validation module implements a certain interface, which is accessible to different modules in a differentiated way.

To finish, it is important to emphasize that a public application programming interface (API) has been generated and the application becomes available in a Git repository that more students could have green, so that it is used and modified by the community in general.

4.1 Main features of the proposed application

In terms of features, the developed application follows the schematic presented in Fig. 4.

In particular, the developed application exhibits the following features:

- Principal: main window that opens upon initiating the program: contains four tabs and a top menu that accesses all program options.

  1. Graph: main window tab where graphics in two dimensions (points) of the events saved by each person (for a scene in playback mode) can be viewed.
  2. Player: main window tab that has a player for video and images and buttons to record, pause, and stop, in addition to displaying the total and current time (in recording mode, the total time is–:–:–while recording).
  3. Summary: main window tab showing two tables with the summary of each of the events and intervals saved by each person.

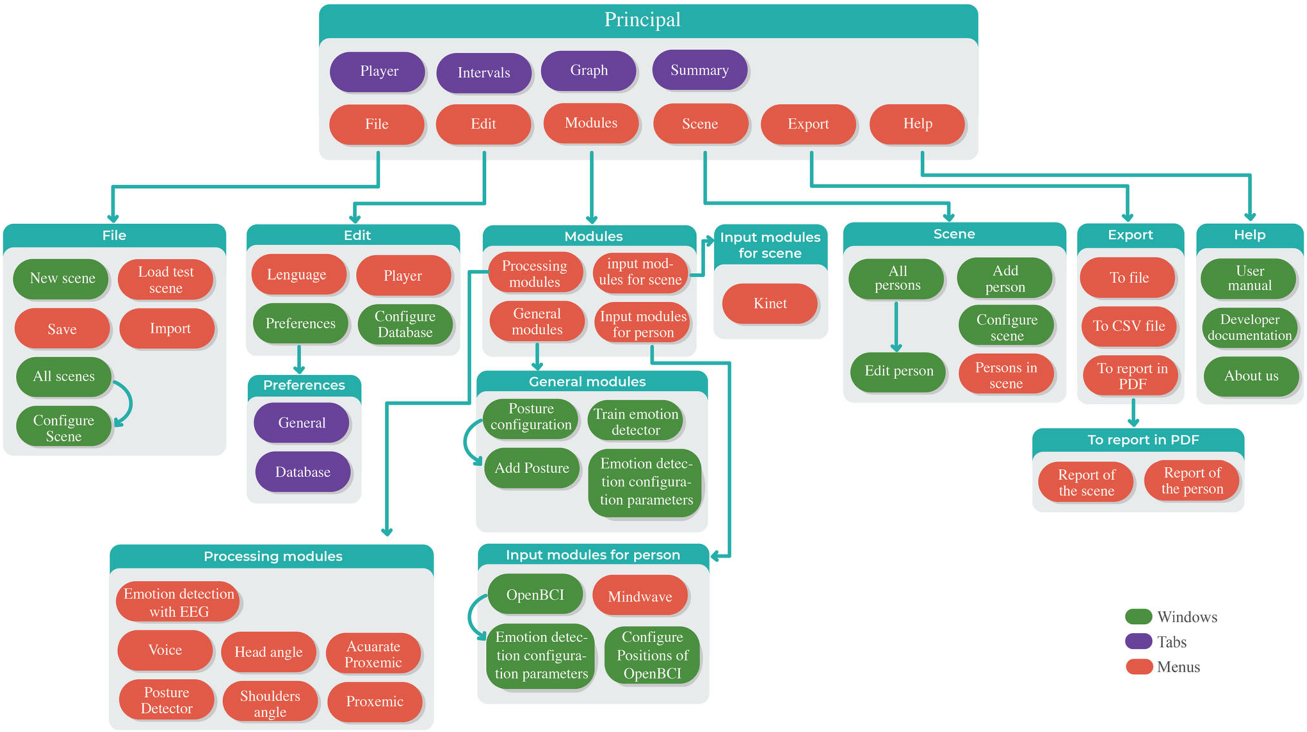


Fig. 4 Application features

4. Intervals: main window tab displaying graphs of intervals saved by each person (being able to add more in addition to selecting person, type, and subtype).

– Menu: upper menu in the app

  – Help: menu with help items.
  – User manual: opens the user manual in PDF format.

    - Developer documentation: opens the developer documentation (auto-generated reference) in HTML format in the default browser.
    - About us: window that shows the logotype and authors, together with a copyright notice.

  – Export: menu with options for exporting the scene or scene data.
  – To file: window for exporting the scene, being able to export the database only or all data associated with the scene, in addition to selecting the route and name of the file.

    - To CSV: sub-menus that allow saving three predefined queries in CSV format: summary of intervals (similar to the summary table in the main window tab), detail of all intervals, and detail of all events.
    - To report in PDF: sub-menu for generating and saving reports in PDF format with graphs and events summary.
    - Report of the scene: generates and saves in a file (one selects path and name) the data for the whole scene, with a general initial section of summary of the whole scene, and then broken down by each participating person.
    - Report of the person: generates and saves a PDF file (one selects path and name), with the report data of a specific person from among the participants in the scene (one selects the person).

  – Scene: menu with options relative to the scene currently open/in use.

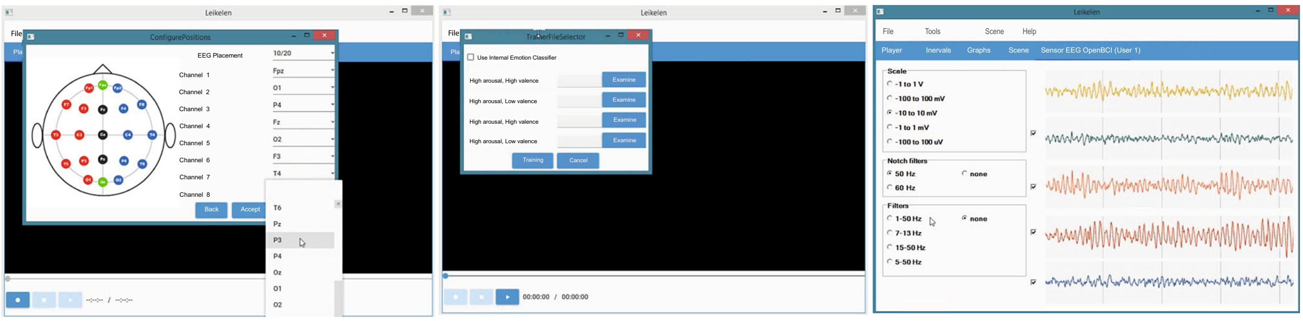
    - All person: window in which a list of all the people that exist in all the scenes saved in the database appears, with options to create a new one (only available if there is a scene in use in monitor mode), to edit the data of a person, and to assign an already existing person to the current scene (same as above, available in monitor mode).
    - Edit person: window for editing the data of a person, with a reminder showing the photograph, and options for name, sex, date of birth, etc.

  - Add person: window to create a person, with a reminder showing their photograph, and fields for name, sex, and date of birth.
  - Configure scene: window for editing the fields of the scene (only available in monitor mode)
  - Persons in scene: sub-menu with people that are currently in the scene, and rapid access to be able to edit and/or see their personal data.

- File: menu with options for opening or managing the scene currently in use.
  - New scene: window for editing the fields of a new scene to be created.
  - Import: open a dialogue box for selecting a scene file (with a database only or with all data in format .leikelen)
  - Save: save the current scene with all its data.
  - Load test scene: load a scene with random dummy data.
  - All scenes: window that shows all the scenes and permits selecting them, opening them, and eliminating them (one or multiple at a time).
  - Configure scene: window for editing the fields of a scene.

- Edit: menu with options and general configurations of the application:
  - Language: sub-menu for selecting language. By default, language is set to auto, which detects the language of the system and selects the most adequate among the languages in the application (or the by default, which is English).
  - Player: sub-menu for activating and deactivating the layers of the player (layer 1 for image and layer 2 for typical framework).
  - Preferences: window with tabs for changing the general options of the application.
    - General: tab with generic options of the application such as name of data directory or maximum number of points in graphs of events to implement decimation.
    - Database: tab with options for connection to the distinct databases that the applications support.
  - Configure database: opens window of preferences with the tab of the database opened simultaneously.

- Modules: menu with all modules displayed.
  - General modules: sub-menu with general modules as extension of the application.
    - Posture configuration: window that lists all existing postures and permits adding, editing, and eliminating. Add posture: window for adding a posture with its name, description, type, and route.
    - Train emotion detector: window for selecting files for training.
    - Emotion detection configuration parameters: window for configuring the configuration parameters using the training algorithm for detection of emotions.
  - Input modules for scene: sub-menu with modules that deliver data from the entire scene (or various people at the same time):
    - Kinect v2: input sensor that uses Kinect v2 (from Xbox One). Option for enabling this sensor.
    - Input modules for person: sub-menu with input modules that deliver data from a single person specifically.
    - MindWave: low-cost EEG MindWave that delivers data of attention and focus, in addition to raw signal.
    - OpenBCI: EEG used for investigation with eight channels and high-quality electrodes (although they can cause pain). Emotion detection configuration parameters: window for configuring the parameters of configuration used for the training algorithm for the detection of emotions. Configure positions of OpenBCI: window to configure and assign the positions in which the electrodes are positioned, using the 20–10 or 10–10 system.
  - Processing modules: sub-menu with processing modules that generate calculations of sensor data and save events or intervals.
    - Emotion detection with EEG: classifier of emotions with EEG that classifies in one of four quadrants (uses EEG).
    - Voice: Detects when a person is talking (uses kinect).
    - Posture detector: classifiers that detect events and samples intervals of the postures of each person.
    - Head angle: calculates the angle in which the head looks in relation to the neck and back
    - Shoulder angle: calculates the angle that the shoulders form with the sternum.
    - Accurate proxemic: calculates the proxemic (distance between participants) using the total smallest distance between their bodies.



EEG Placement: 10/20
Channel 1: Fpz
Channel 2: O1
Channel 3: P4
Channel 4: Fz
Channel 5: O2
Channel 6: F3
Channel 7: T4
Channel 8

□ Use Internal Emotion Classifier
High arousal, High valence
High arousal, Low valence
High arousal, High valence
High arousal, Low valence

Scale: ○ -1 to 1 V ○ -100 to 100 mV ● -10 to 10 mV ○ -1 to 1 mV ○ -100 to 100 uV
Notch filters: ● 50 Hz ○ 60 Hz ○ none
Filters: ○ 1-50 Hz ○ 7-13 Hz ○ 15-50 Hz ○ 5-50 Hz ● none

Fig. 5 Proposed software that supports the emotion recognition

- Proxemic: calculates the proxemic (distance between participants) using the distance between the central points of their torsos.

In Fig. 5 we present some interfaces of the proposed software with AS for the emotion recognition.[3]

5 Autonomous approach for emotion recognition

In this section, we present the autonomous search approach and its integration with the bat algorithm.

5.1 Autonomous search

The metaheuristics are not a an exact science that given the same input, will have the same output. Generally, their behavior is controlled by random occurrences and by set of adjustable parameters, whose modification makes uncertain changes in the quality of the solution. Finding the optimal value for these parameters can be an optimization problem by itself that normally involves tedious and lengthy tests of different values. For this reason, researchers are impulsed to develop automated ways to tune the parameters in an optimal way, setting the foundations of autonomous search [27].

Autonomous search (AS) [28, 29] is a system that has the ability to modify the internal components of the algorithm when it is exposed to external forces and opportunities, with the objective of improving the performance of the algorithm in the resolution of problems, and to achieve this, adapts the search strategies according to the characteristics of the problem.

The main purpose of this technique is to improve the performance of the algorithm. This improvement can be measured in terms of the reduction of solving time, the number of iterations needed to converge to the best solution, or by making the algorithm converge more consistently to the global optimum. Successful works of this approach solving different constraint satisfaction problems can be seen in [30] and more recently in [31, 32].

5.2 Original bat algorithm

Bat algorithm has widely been used for solving optimization and selection problems. When we studied the literature, we found successful works that apply this technique in classification problems. For instance, in [33] the authors describe how to use bat algorithm for finding the best vector of features. Other work that applies this method can be seen in [34]. In this manuscript, we can observe how to select the ideal features on an unsupervised classification process. In this work, we propose a variation of the bat algorithm that integrates the ability to auto-adjust its behavior during the run. The bat algorithm was proposed by Xin-She Yang in 2010 and has had several applications to date. For example, in [35], the authors use this method to extract the best features that feed a super vector machine model. In the area of wireless sensor networks, the bat algorithm has been effective for the geographic allocation of [36].

The original procedure of the algorithm is inspired by the fascinating capacity of echolocation, a behavior that certain species of bats possess. This behavior allows them to differentiate their food or avoid obstacles in complete darkness. To use echolocation, the micro-bats species that possess this ability emit a signal through their sonar to recognize positions of objects in their flight zone in order to avoid possible collisions and to be able to feed on insects or any type of detected prey. Figure 6 illustrates the behavior of micro-bats utilizing their sonar.

To design the procedure, Yang defined three simple rules that should be followed for the sake of the generalization of the method:

1. All species of bats possess the characteristics of echolocation for measuring distance and determining the difference between prey and an obstacle.
2. The flight of a bat is random. The velocity is controlled by v_i and the position is represented by x_i. The

[3] https://github.com/leikelen-team/Leikelen/releases/tag/v1.2.0.

frequency of sonar f_i is defined initially A_0 and varies in accordance with the wavelength and loudness. The value of wavelength (frequency) can be adjusted through the emission of pulse (sonar) by the bat. This ratio of emission of pulse r_i varies between 0 and 1, depending on the proximity of the object.

3. Although the loudness can vary in many ways, it is assumed that it changes from a very high positive value A_0 to a low constant value $A_{\min}$.

The procedure that describes this method is shown in Fig. 7.

Initially, the population of bats is positioned in a random manner in the search space after randomly defining their velocities. The input parameters are then initialized, defining the values for population size, the maximum and minimum frequencies, and the stop criterion, in this case, the maximum quantity of generations (iterations). The next step is to evaluate the solutions generated and choose the best of these. In the iterative step, the algorithm evaluates the stop criterion. As long as it is not fulfilled, the method attempts to improve the generated solutions by adjusting the frequency then updating the speed and position of each bat. This process is guided by Eqs. (1), (2), and (3).

$$f_i = f_{\min} + (f_{\max} - f_{\min})\beta \tag{1}$$

$$v_i(t) = v_1(t-1) + (x^* - x_i(t))f_i \tag{2}$$

$$x_i(t) = x_i(t-1) + v_i(t) \tag{3}$$

In Eq. (1), β is a value randomly generated starting from a random distribution and permits varying the frequency in a non-deterministic manner. In Eq. (2), the unknown variable x^* represents the best position (solution) encountered to this point and is utilized for measuring the difference in position toward the solution $x_i(t)$ in the iteration t. The new velocity $v_i(t)$ is updated starting from the difference in position and the previous velocity $v_i(t-1)$. Finally, the new position in Eq. (4) is obtained starting from the previous position $x_i(t-1)$ and the updated velocity $v_i(t)$.

In the next step, the pulse radius of each bat is evaluated. If there is a higher intensity sonar than that of the current bat, it will try to move toward it. This is known as exploration. To do this, the algorithm evaluates whether the pulse r_i is overcome by a random value and thus generates a new local solution close to the best solution. This mechanism operates under the random walk scheme described in Eq. (4), where ϵ is a random value and $A(\bar{t})$ represents the average volume of all the solutions in the instance t.

$$x_{\text{new}} = x_{\text{old}} + \epsilon A(\bar{t}) \tag{4}$$

Finally, if the new solution overcomes the best global solution, it is accepted as the best solution, the pulse radius is increased, and the volume is decreased. Formally, these actions are carried out through Eqs. (5) and (6).

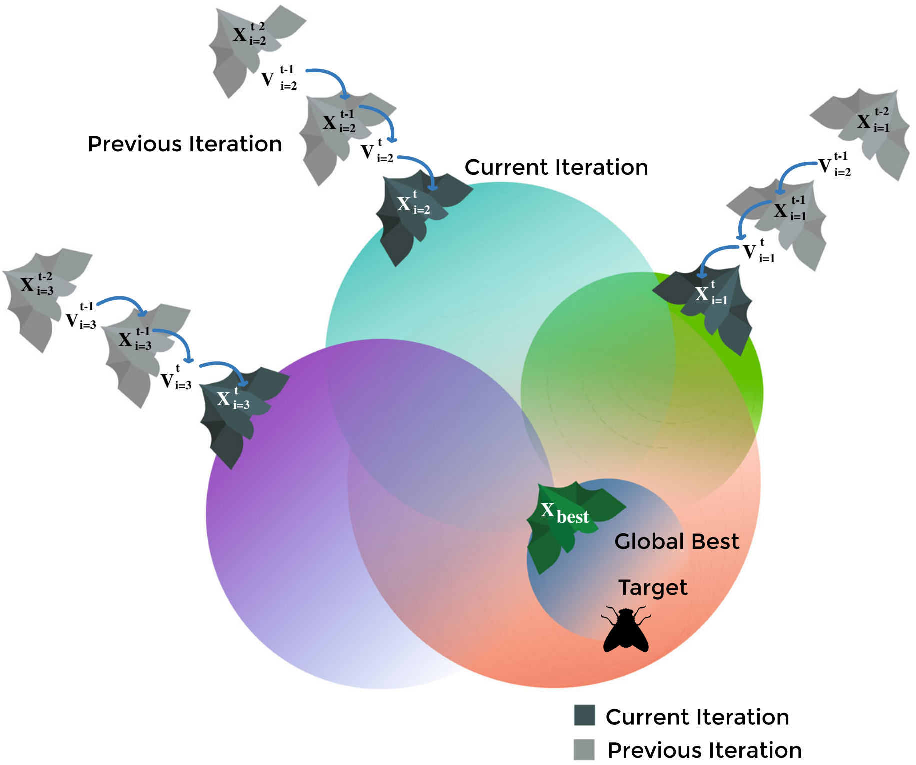


Fig. 6 Echolocation features of micro-bats

$$A_i(t) = A_i(t-1) + \alpha A_i(t) \quad (5)$$

$$r_i(t) = r_i(0)[1 - \exp^{\gamma t}] \quad (6)$$

In spite of being used frequently, the bat algorithm presents the problem of finding the best configuration of values for its input parameters. This task is complex and can be seen as an optimization problem in itself. Some works present external tools for the [37] algorithm to determine the best configuration. However, the majority of the time is spent in running and collecting enough information to define the best values. This activity is costly and requires an expert user to run and adjust each value. Although it is viable, it is inefficient because the values found do not guarantee to be useful for all instances of the same problem.

5.3 Improved bat algorithm

Autonomous search is a particular case of self-adaptive systems. Its main objective is to improve the process of solving a problem, modifying its internal components when exposed to external changes or opportunities [29].

The autonomous search technique can be analyzed through performance indicators that measure the behavior of its internal components. These correspond mainly to algorithms involved in the search process: heuristics, inferences, exploration process, and exploitation process, among others. External components drive changes through information evolution. This information is taken directly from the problem, for example the exact or estimated search space size, number of subproblems to be resolved, quantity of enumeration attempts, or indirectly through the perceived efficiency of the individual components, such as the ability to exploit or explore initially unpromising regions.

Taking advantage of the flexibility provided by autonomous search, we propose an improvement to the bat algorithm to enhance the exploration process of promising regions when it is quickly converging or trapped in local

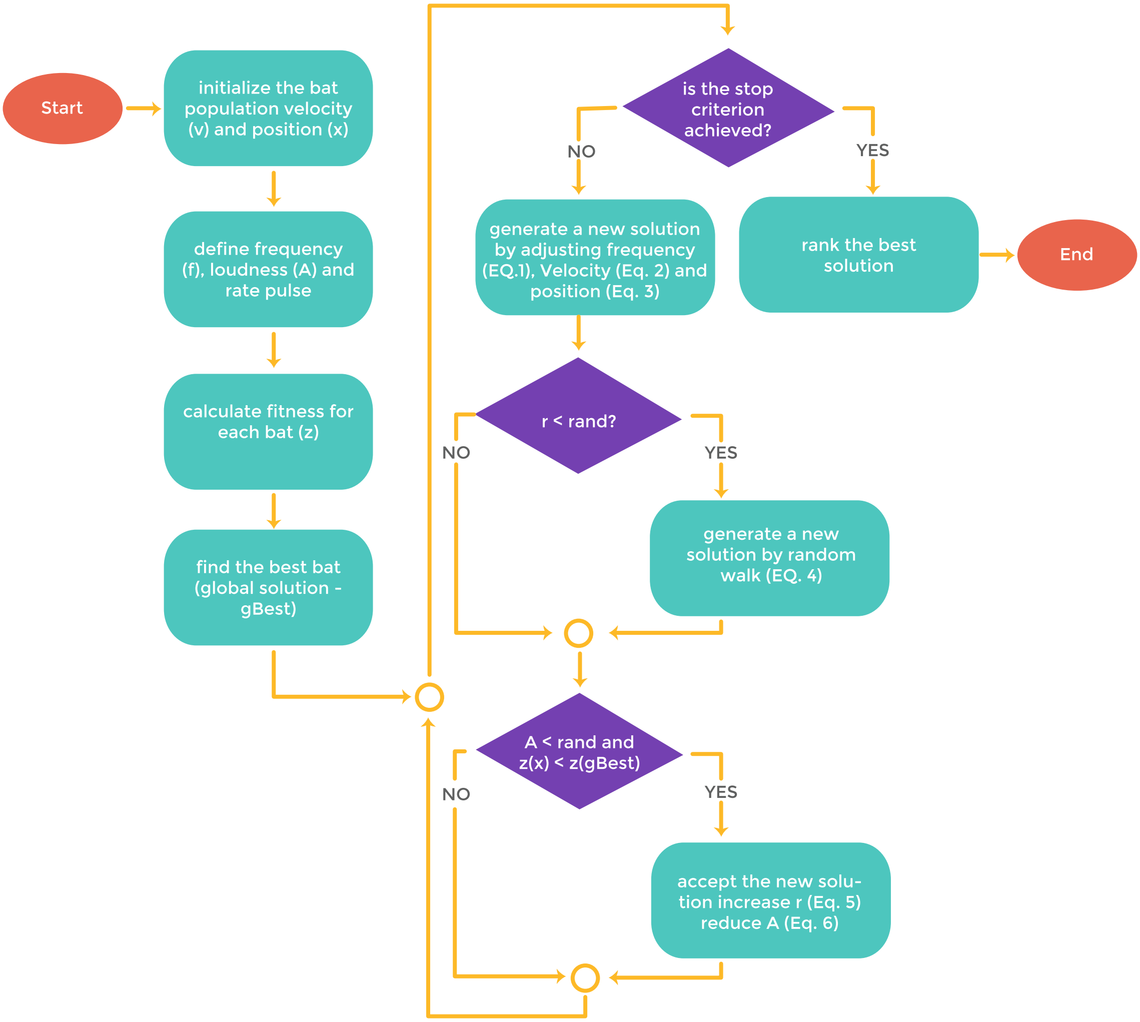

Start
initialize the bat population velocity (v) and position (x)
define frequency (f), loudness (A) and rate pulse
calculate fitness for each bat (z)
find the best bat (global solution - gBest)
is the stop criterion achieved?
NO
YES
generate a new solution by adjusting frequency (EQ.1), Velocity (Eq. 2) and position (Eq. 3)
rank the best solution
End
r < rand?
NO
YES
generate a new solution by random walk (EQ. 4)
A < rand and z(x) < z(gBest)
NO
YES
accept the new solution increase r (Eq. 5) reduce A (Eq. 6)

Fig. 7 Process flow for the bat algorithm

Algorithm 1 Adaptive approach for the population size parameter $(popSize)$

```
objective function $z(x), x = \langle x_1, \ldots, x_{popSize} \rangle$.
initialize the bat population: velocity $v_i$ and position $x_i$, $(\forall\, i = 1, 2, \ldots, popSize)$.
define pulse frequency $f_i$ at $x_i$, $(\forall\, i = 1, 2, \ldots, popSize)$.
initialize pulse rates $r_i$ and the loudness $A_i$, $(\forall\, i = 1, 2, \ldots, popSize)$.
calculate the fitness each bat (solutions) by using the objective function.
find the global best solution.
while the termination conditions are not met. do
  if original bat algorithm is trapped in a local optimum (stagnation) then
    calculate the probability of change $prob \leftarrow \frac{cost(gBest) - cost(x_{worst})}{\sum cost(x_i)}$
    $nBats \leftarrow roundup(popSize \times prob)$
    $rand \sim U[0, 1]$
    if $rand > prob$ then
      for all bat $x_i, (\forall\, i = 1, 2, \ldots, nBat)$ do
        $x_i \leftarrow gBest + \epsilon \bar{A}(t)$
      end for
      $popSize \leftarrow popSize + nBats$
    else
      {remove $nBats$ latest solutions.}
      {generate $nBats$ random solutions.}
    end if
  end if
  for all bat $x_i, (\forall\, i = 1, \ldots, popSize)$ do
    generate new solutions through the equations (1), (2) and (3).
    if $r_i < rand$ then
      {exploration process}
      randomly select a solution among the best solutions.
      generate a local solution around the best solution by Eq. (4).
    end if
    if $rand < A_i$ and $z(x_i) < z(\hat{x})$ then
      {exploitation process}
      accept the new solutions.
      increase $r_i$ and reduce $A_i$, via equations (6) and (5), respectively.
    end if
  end for
end while
return ranking of solutions and show the current best.
```

optimums. In particular, we include a routine that adjusts at run time the bat population (number of solutions) according to the performance displayed by the algorithm. The idea is inspired by the probability of change of chromosomes into the genetic algorithm [38, 39].

The procedure that describes the autonomous search behavior is shown in Fig. 8.

6 Results and discussion

Firstly, the software implements a run-time classifier to the emotion recognition. The proposed architecture supports the integration of new components in plug and play mode. The main idea is using this tool in real learning environment in order to facilitate the skill's development. This is because the emotional aspects play a critical role in learning processes. Initially, the implemented software receives an EEG signal to be processed. Then, we decompose the signal by using the empirical model decomposition (EMD) [40]. We have used this technique because it allows softening the signals and decreasing noise, which is especially useful in physiological signals. Next, we calculate features via the autonomous bat algorithm as shown in Fig. 9. However, when the emotions are analyzed is in off-line mode, it is possible to add passband filter. For the reconstruction of the beta band, we used low-pass and high-pass Butterworth filters [24]. Signals were filtered using a third-order bandpass Butterworth filter with a cutoff frequency of 12.5 and 30 Hz and the resonant frequency equal to 0.1 Hz [41].

According to data acquisition, this paper mainly takes four categories of emotions into account. Consider the Russell's circumplex model of emotions dividing into four quadrants (high and low present values greater or equal to 5 and lower than 5, respectively).

In the following, we design and implement a variation of the bat algorithm which includes an autonomous parameter

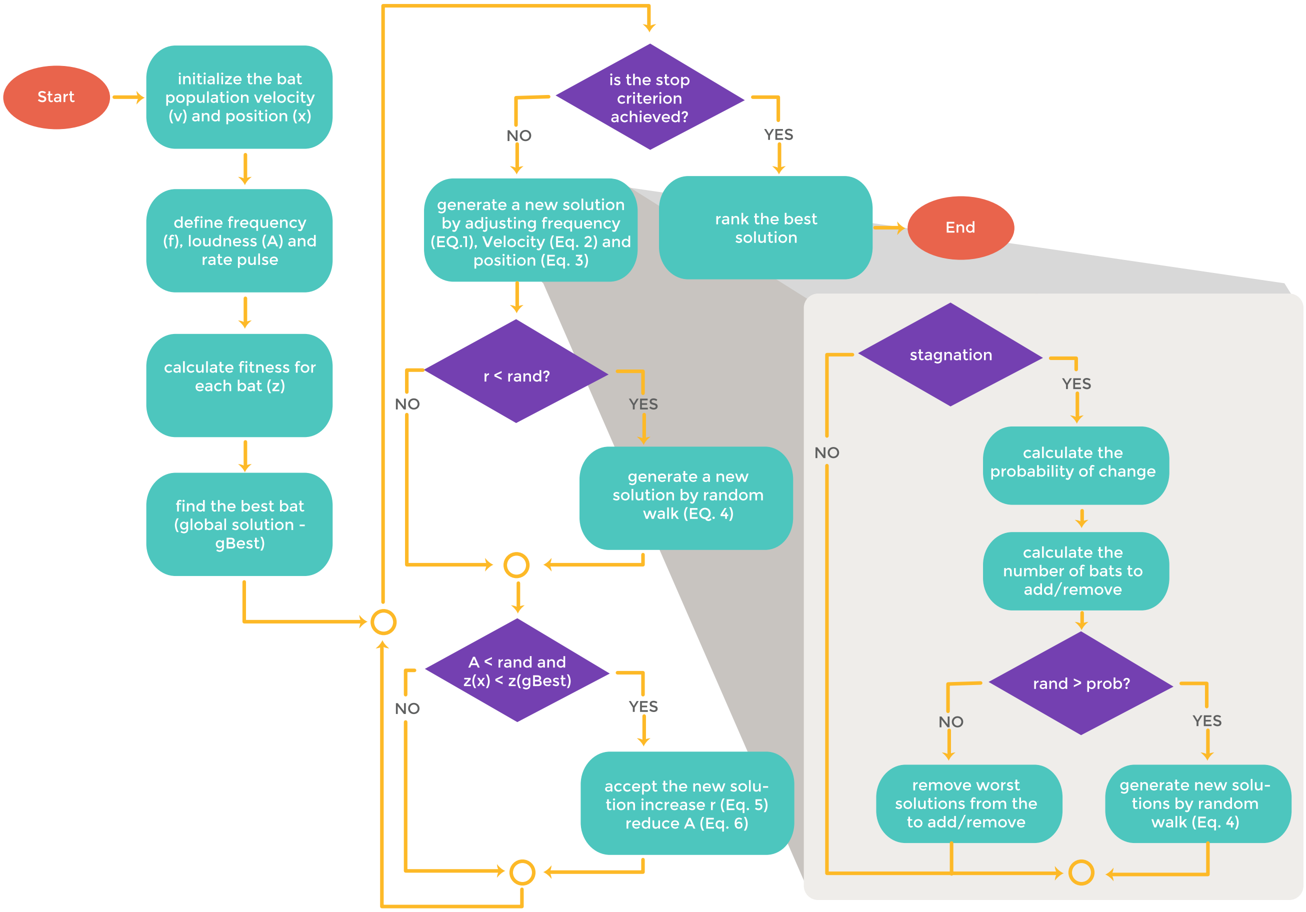

Start
initialize the bat population velocity (v) and position (x)
define frequency (f), loudness (A) and rate pulse
calculate fitness for each bat (z)
find the best bat (global solution - gBest)
is the stop criterion achieved?
NO
YES
generate a new solution by adjusting frequency (EQ.1), Velocity (Eq. 2) and position (Eq. 3)
rank the best solution
End
r < rand?
NO
YES
generate a new solution by random walk (EQ. 4)
A < rand and z(x) < z(gBest)
NO
YES
accept the new solution increase r (Eq. 5) reduce A (Eq. 6)
stagnation
NO
YES
calculate the probability of change
calculate the number of bats to add/remove
rand > prob?
NO
YES
remove worst solutions from the to add/remove
generate new solutions by random walk (Eq. 4)

Fig. 8 Process flow for the autonomous bat algorithm

setting during the run for improving the multiclass classification performance. The defined classes for the classifier are: high arousal/high valence, high arousal/low valence, low arousal/low valence, and low arousal/high valence.

For our computational experiments, we use tenfold cross-validation for testing the algorithm. The weight SVM with Gaussian kernel involves the penalty parameter C (complexity) as well as the kernel parameter to be optimized. The main idea is finding the best values for both parameters. Then, we compute our gold standard sample entropy (SampEn) for comparing our proposal. To calculate SampEn values, we only consider the first four calls to the intrinsic mode functions (IMF). Thus, we seen that the embedded dimension m varies between 1 and 2, and the probability of similarity on two simultaneous datasets is between 0.1 and 0.25. Table 1 summarizes the used initial parameter for all experiments.

Finally, the performance of the proposed solution was experimentally evaluated by using three different emotion databases: DEAP, MAHNOB HCI Tagging, and eNTERFACE [42]. Their sample sizes are 458, 266, 260, and 296, respectively.

After running 31 times with 100 iterations each one, we can guarantee that the improved version of the approximate method is technique robustness, and it is appropriated to be used as an optimizer for the super vector machine. For instance, SampEn for DEAP database generates an error close to 25%. For self-database, the accuracy given by autonomous proposal is close to 95%. The original version of the bat algorithm exhibits a good performance also with a accuracy of 93%. Now, if we consider the classification results for MAHNOB HCI Tagging database, we can again see that our autonomous approach is more effective than the original version and far surpasses SampEn. Here, for SampEn we obtained an variable error between 20 and 25%, an error close to 6% given by the original bat algorithm. For the autonomous bat, we obtained an accuracy overcomes 95%. Finally, if the exhibited performance of the methods is analyzed for eNTERFACE database, we can observe that the autonomous approach is again better than SampEn and it exhibits a similar behavior with the original bat algorithm. SampEn achieves an error close to 25%, and the original bat algorithm and its autonomous version, both reach a 96% of accuracy. This shows that using a

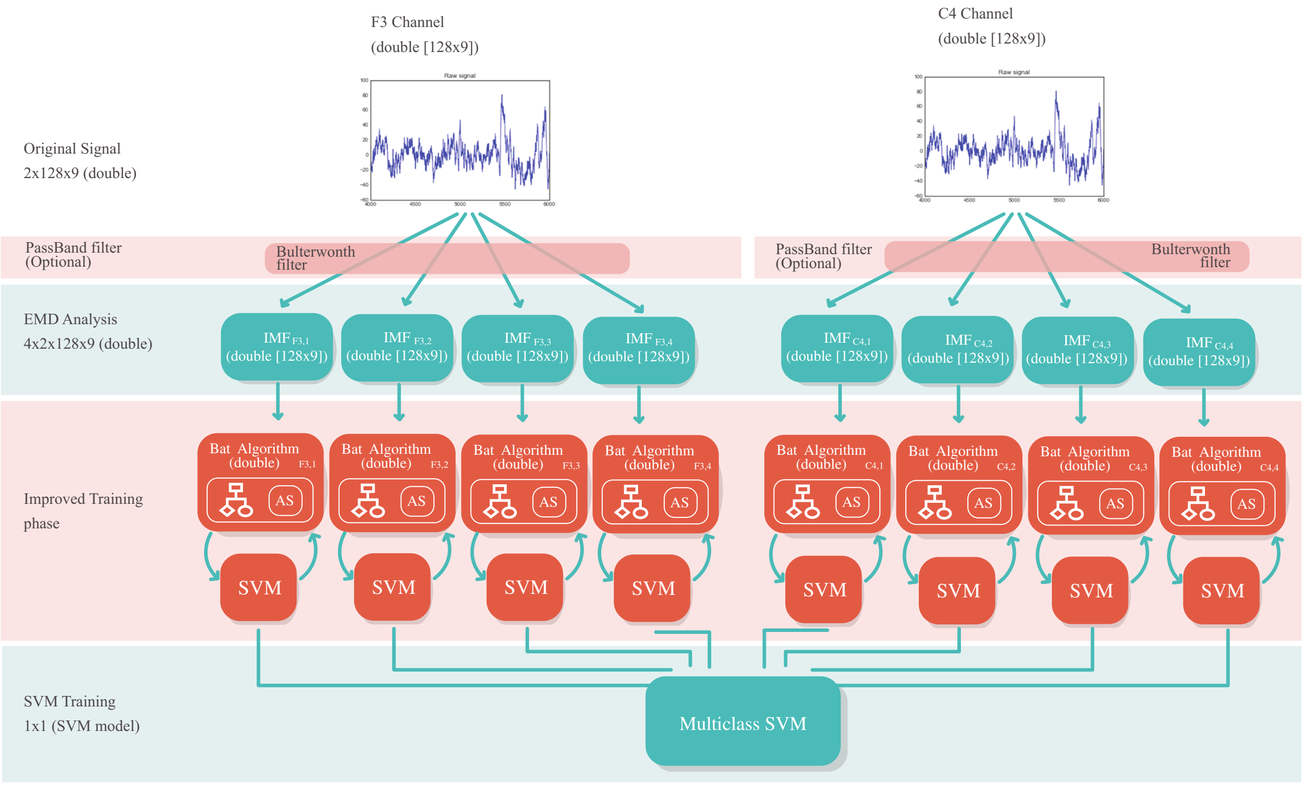

F3 Channel
(double [128x9])
C4 Channel
(double [128x9])
Original Signal
2x128x9 (double)
Raw signal
PassBand filter
(Optional)
Bulterwonth
filter
EMD Analysis
4x2x128x9 (double)
IMF F3,1 (double [128x9])
IMF F3,2 (double [128x9])
IMF F3,3 (double [128x9])
IMF F3,4 (double [128x9])
IMF C4,1 (double [128x9])
IMF C4,2 (double [128x9])
IMF C4,3 (double [128x9])
IMF C4,4 (double [128x9])
Improved Training
phase
Bat Algorithm (double) F3,1
Bat Algorithm (double) F3,2
Bat Algorithm (double) F3,3
Bat Algorithm (double) F3,4
Bat Algorithm (double) C4,1
Bat Algorithm (double) C4,2
Bat Algorithm (double) C4,3
Bat Algorithm (double) C4,4
AS
SVM
SVM Training
1x1 (SVM model)
Multiclass SVM

Fig. 9 Proposed method using autonomous search approach integrated to the bat algorithm

classification approach improved by a bio-inspired technique is a better alternative than the traditional way of using the sample entropy calculation.

Charts in Fig. 10 show the minimum values, maximum values, in addition to the average and medium values of 31 runs along with 100 iterations, for the three databases. While it is true that there is no significant difference

Table 1 Parameter setting to the entropy and the autonomous search bat algorithm

Section	Component	Description	Value
Data selection	Number of sessions	Emotion elicitation trials	563
	Frequency	Each second has 128 samples or values	128 Hz
	Frame	To classify each frame, it lasts 9 s, without overlapping	9 s
Sample entropy	N	Number of samples	128 samples
	m	Embedded dimension	2
	r	Probability of similarity on two simultaneous datasets	0.15
Empirical mode decomposition	Order	Number of calls to the IMFs [40]	4
Autonomous bat algorithm	n	Number of bats (solutions)	30
	$f_{\min}$	Minimum frequency	0.75
	$f_{\max}$	Maximum frequency	1.25
	α	Positive constant	0.9
	γ	Positive constant	0.9
	ϵ	Positive constant	0.9
	T	Maximum iterations	100
Miscellaneous	–	Runs of the approximate approach	30
	–	Number of used cores (processors)	8

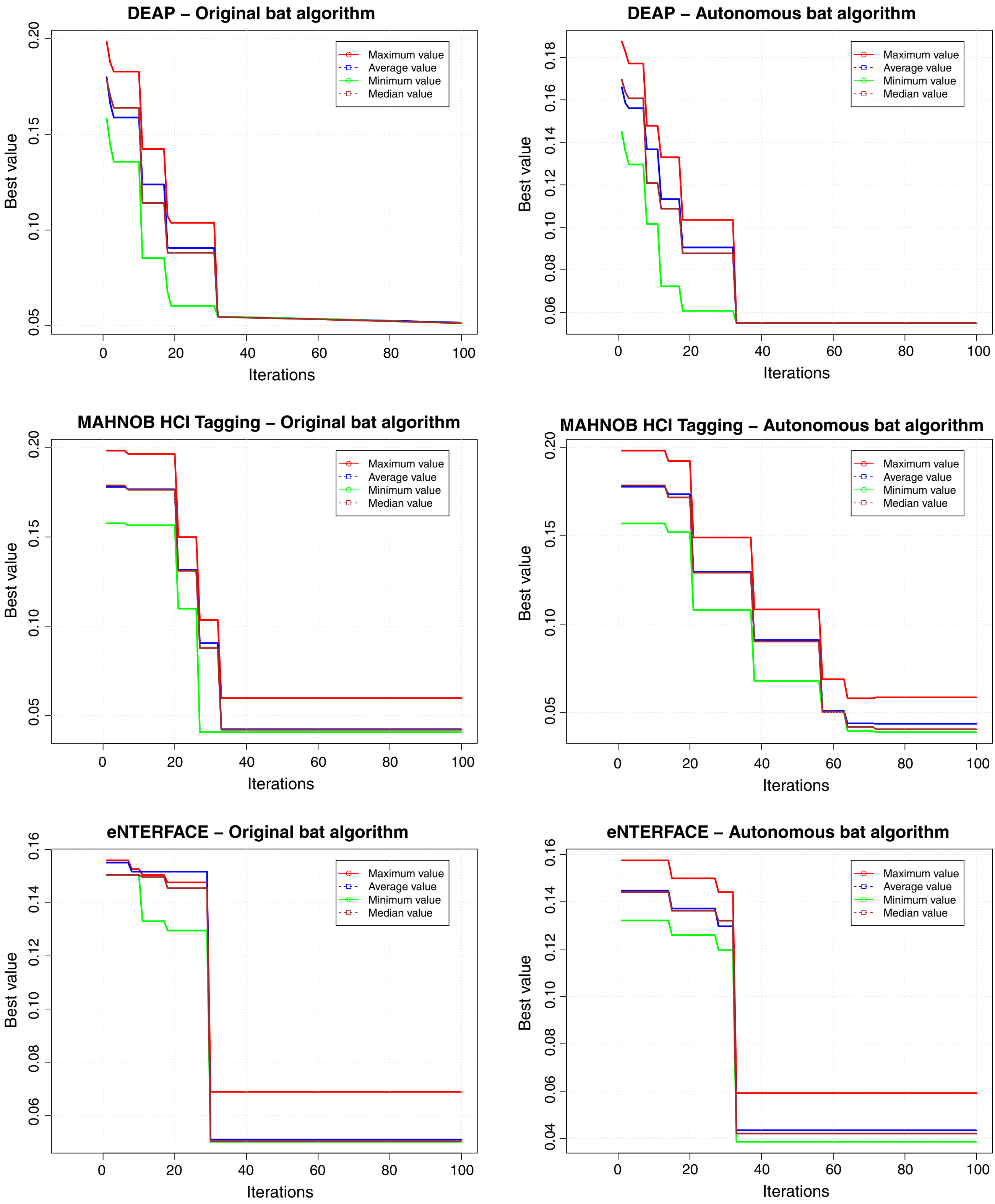


Fig. 10 Algorithm performance

between the original version and the autonomous version for the bat algorithm, in the classification process a distance higher than 1% is very important. In this case, the autonomous bat algorithm works better than its traditional way.

Independent of the databases used, at the beginning the algorithm generates randomly solutions (before the first iteration) and the errors rates for the best solutions vary between 12 and 21%, approximately. The reported average value is very close to 15% and the median value exceeds 16%. At the end of iterations, these values are improved reaching an standard deviation (blue line in the chart) equal to interquartile range (brown line in the chart), and both in zero. The performance of the approximate algorithms is always exhibited after 10% iterations. In this point, interquartile range is equal to 0, and it indicates the best value has been achieved very early. With respect to standard deviation, we can note that the behavior is similar to the previous analysis but begins from 20% of iterations.

Finally, it is possible to conclude that our autonomous approach inspired by the bio-sonar of bats provides the ability to optimize the input parameters of SVM in order to find the best accuracy in low time period. The final result for accuracy was 95.04% that it is dramatically better than the average calculated by using a standard approach although the entropy factor. In this case, the median value is lower than 75%.

7 Conclusions

Emotions are a critical aspect of human behavior. In fact, the emotions have been subject to scientific research for more than a century. One of the areas that have a crucial importance is in the cognitive processes, in particular how they can affect the learning of people.

Given the importance of emotions in cognitive processes, in this work, we present the design and implementation of software to support the emotions recognition by using a low-cost EEG (OpenBCI). For the emotions recognition, a hybrid method was used. This method uses an autonomous approach with the bat algorithm. Autonomous search is a modern approach used in metaheuristics to solve the parameter setting problem. The main idea is using the improved bat algorithm for boosting the classification process led by the SVM.

To validate our approach, we use an open database that had 563 sessions. After several tests, the results show that our approach reaches an 95% accuracy.

Given the results obtained, we believe that our approach contributes to two aspects. The first is related to scalable architecture with usable software that allows the detection of emotional states. On the other hand, we also believe that the development of a bio-inspired method for the emotions recognition contributes to the state of the art. This result shows that our work is a real and trustworthy alternative in the signal classification for the emotion recognition.

Finally, as future work, we plan to carry out tests with several databases as well as use the proposed software in real environments and thus confirm the importance of our contribution [43, 44].

Acknowledgements Carla Taramasco has been supported by CORFO—CENS 16CTTS-66390 through the National Center on Health Information Systems. This work is also supported by the National Commission for Scientific and Technological Research (CONICYT) through the Program STIC-AMSUD 17STIC- 03: “MONITORing for ehealth,” FONDEF ID16I10449 “Sistema inteligente para la gestión y análisis de la dotación de camas en la red asistencial del sector público”, and MEC80170097 “Red de colaboración científica entre universidades nacionales e internacionales para la estructuración del doctorado y magister en informática médica en la Universidad de Valparaíso.” Ricardo Soto is supported by Grant CONICYT/FONDECYT/REGULAR/1160455. María Francisca Alonso-Sńchez is supported by CONICYT/FONDECYT/INICIACION/11160212. Victor Hugo C. de Albuquerque appreciates the received support from the Brazilian National Council for Research and Development (CNPq, Grant #304315/2017-6).

Compliance with ethical standards

Conflict of interest The authors declare that there is no conflict of interest regarding the publication of this article.

Data availability The data generated to support the findings of this study have been deposited in the Figshare repository (https://figshare.com/s/1a80d472266b39ed2723).

References

1. Pekrun R (1992) The impact of emotions on learning and achievement: towards a theory of cognitive/motivational mediators. Appl Psychol 41(4):359–376
2. Ibáñez N (2002) Emotions in the classroom. Estudios pedagógicos (Valdivia). https://doi.org/10.4067/s0718-07052002000100002
3. Marchand GC, Gutierrez AP (2012) The role of emotion in the learning process: Comparisons between online and face-to-face learning settings. Internet High Educ 15(3):150–160. https://doi.org/10.1016/j.iheduc.2011.10.001
4. Oatley K, Keltner D, Jenkins JM (2006) Understanding emotions. Blackwell Publishing, Hoboken
5. Lee W, Norman MD (2016) Affective computing as complex systems science. Procedia Comput Sci 95:18–23
6. Russell JA (1980) A circumplex model of affect. J Personal Soc Psychol 39(6):1161
7. Hu X, Yu J, Song M, Yu C, Wang F, Sun P, Wang D, Zhang D (2017) EEG correlates of ten positive emotions. Front Hum Neurosci 11:26. https://doi.org/10.3389/fnhum.2017.00026
8. Chen M, Han J, Guo L, Wang J, Patras I (2015) Identifying valence and arousal levels via connectivity between EEG channels. In: 2015 international conference on affective computing and intelligent interaction (ACII). IEEE. https://doi.org/10.1109/acii.2015.7344552

9. Sanei S, Chambers JA (2007) EEG signal processing. Wiley, Hoboken
10. LeDoux JE (2000) Emotion circuits in the brain. Ann Rev Neurosci 23(1):155–184. https://doi.org/10.1146/annurev.neuro.23.1.155
11. Wang XW, Nie D, Lu BL (2014) Emotional state classification from EEG data using machine learning approach. Neurocomputing 129:94–106. https://doi.org/10.1016/j.neucom.2013.06.046
12. Fredrickson BL (1998) What good are positive emotions? Rev General Psychol 2(3):300–319. https://doi.org/10.1037/1089-2680.2.3.300
13. Schunk DH (2013) Motivation in education: theory, research, and applications, 4th edn. Pearson, London
14. Teplan M et al (2002) Fundamentals of EEG measurement. Meas Sci Rev 2(2):1–11
15. Luck SJ (2014) An introduction to the event-related potential technique. MIT press, Cambridge
16. Weinberg A, Hajcak G (2010) Beyond good and evil: the time-course of neural activity elicited by specific picture content. Emotion 10(6):767–782. https://doi.org/10.1037/a0020242
17. Hajcak G, Olvet DM (2008) The persistence of attention to emotion: brain potentials during and after picture presentation. Emotion 8(2):250–255. https://doi.org/10.1037/1528-3542.8.2.250
18. Wang Z, Xue X (2014) Multi-class support vector machine. Springer, Cham, pp 23–28. https://doi.org/10.1007/978-3-319-02300-7_2
19. Huang NE, Shen Z, Long SR et al (1998) The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis. Proc R Soc A Math 454(1971):903–995. https://doi.org/10.1098/rspa.1998.0193
20. Ali M, Al Machot F, Mosa AH, Kyamakya K (2016) A novel EEG-based emotion recognition approach for e-healthcare applications. In: Proceedings of the 31st annual ACM symposium on applied computing. ACM, pp 162–164
21. Zhang Q, Lee M (2009) Analysis of positive and negative emotions in natural scene using brain activity and GIST. Neurocomputing 72(4–6):1302–1306. https://doi.org/10.1016/j.neucom.2008.11.007
22. Mohammadi Z, Frounchi J, Amiri M (2017) Wavelet-based emotion recognition system using EEG signal. Neural Comput Appl 28(8):1985–1990. https://doi.org/10.1007/s00521-015-2149-8
23. Zhang Y, Ji X, Zhang S (2016) An approach to EEG-based emotion recognition using combined feature extraction method. Neurosci Lett 633:152–157. https://doi.org/10.1016/j.neulet.2016.09.037
24. Munoz R, Olivares R, Taramasco C, Villarroel R, Soto R, Barcelos TS, Merino E, Alonso-Sánchez MF (2018) Using black hole algorithm to improve EEG-based emotion recognition. Comput Intell Neurosci 2018:1–21. https://doi.org/10.1155/2018/3050214
25. Kruchten P (1995) The 4 + 1 view model of architecture. IEEE Softw 12(6):42–50. https://doi.org/10.1109/52.469759
26. Munoz R, Villarroel R, Barcelos TS, Souza A, Merino E, Guiez R, Silva LA (2018) Development of a software that supports multimodal learning analytics: a case study on oral presentations. JUCS 24(2):149–170
27. Gendreau M, Jean-Yves P (2010) Handbook of metaheuristics. Springer, Berlin. https://doi.org/10.1007/978-1-4419-1665-5
28. Hamadi Y, Monfroy E, Saubion F (2010) What is autonomous search? Emotion 10(6):767–782. https://doi.org/10.1007/978-1-4419-1644-0_11
29. Hamadi Y, Monfroy E, Saubion F (2011) An introduction to autonomous search. In: Hamadi Y, Monfroy E, Saubion F (eds) Autonomous search. Springer, Berlin, Heidelberg, pp 1–11. https://doi.org/10.1007/978-3-642-21434-9_1
30. Soto R, Crawford B, Palma W, Monfroy E, Olivares R, Castro C, Paredes F (2015) Top-k based adaptive enumeration in constraint programming. Math Probl Eng 2015:580785. https://doi.org/10.1155/2015/580785
31. Soto R, Crawford B, Olivares R, Galleguillos C, Castro C, Johnson F, Paredes F, Norero E (2016) Using autonomous search for solving constraint satisfaction problems via new modern approaches. Swarm Evolut Comput 30:64–77. https://doi.org/10.1016/j.swevo.2016.04.003
32. Soto R, Crawford B, Palma W, Galleguillos K, Castro C, Monfroy E, Johnson F, Paredes F (2015) Boosting autonomous search for CSPs via skylines. Inf Sci 308:38–48. https://doi.org/10.1016/j.ins.2015.01.035
33. Mizobe R, Martins L, Rodrigues D, Pontara K, Papa JP, Yang XS (2013) Binary bat algorithm for feature selection. Swarm Intell Bio-inspired Comput Theory Appl. https://doi.org/10.1016/B978-0-12-405163-8.00009-0
34. Sylvia A, Rajalaxmi R (2015) Unsupervised feature selection using binary bat algorithm. In: 2nd international conference on electronics and communication systems, pp 451–456. https://doi.org/10.1109/ECS.2015.7124945
35. Rodrigues D, Pereira LA, Nakamura RY, Costa KA, Yang XS, Souza AN, Papa JP (2014) A wrapper approach for feature selection based on bat algorithm and optimum-path forest. Expert Syst Appl 41(5):2250–2258. https://doi.org/10.1016/j.eswa.2013.09.023
36. Goyal S, Patterh MS (2015) Modified bat algorithm for localization of wireless sensor network. Wirel Pers Commun 86(2):657–670. https://doi.org/10.1007/s11277-015-2950-9
37. López-Ibáñez M, Dubois-Lacoste J, Cáceres LP, Birattari M, Stützle T (2016) The irace package: iterated racing for automatic algorithm configuration. Oper Res Perspect 3:43–58. https://doi.org/10.1016/j.orp.2016.09.002
38. Jewajinda Y, Chongstitvatana P (2012) A parallel genetic algorithm for adaptive hardware and its application to ECG signal classification. Neural Comput Appl 22(7–8):1609–1626. https://doi.org/10.1007/s00521-012-0963-9
39. Sun N, Lu Y (2018) A self-adaptive genetic algorithm with improved mutation mode based on measurement of population diversity. Neural Comput Appl. https://doi.org/10.1007/s00521-018-3438-9
40. Huang NE, Shen Z, Long SR, Wu MC, Shih HH, Zheng Q, Yen NC, Tung CC, Liu HH (1998) The empirical mode decomposition and the hilbert spectrum for nonlinear and non-stationary time series analysis. Proc R Soc A Math Phys Eng Sci 454(1971):903–995. https://doi.org/10.1098/rspa.1998.0193
41. Butterworth S (1930) On the theory of filter amplifiers. Wirel Eng 7(6):536–541
42. Soleymani M, Lichtenauer J, Pun T, Pantic M (2012) A multimodal database for affect recognition and implicit tagging. IEEE Trans Affect Comput 3(1):42–55
43. Flandrin P, Rilling G, Goncalves P (2004) Empirical mode decomposition as a filter bank. IEEE Signal Process Lett 11(2):112–114
44. Soto R, Crawford B, Carrasco C, Almonacid B, Reyes V, Araya I, Misra S, Olguín E (2016) Solving manufacturing cell design problems by using a dolphin echolocation algorithm. In: Computational science and its applications—ICCSA 2016. Springer, pp 77–86. https://doi.org/10.1007/978-3-319-42092-9_7