Acquiring and Processing Data Using Simplified EEG-based Brain-Computer Interface for the Purpose of Detecting Emotions

Rafal Chalupnik, Katarzyna Bialas, Ireneusz Jozwiak, Michal Kedziora
Faculty of Computer Science and Management
Wroclaw University of Science and Technology
Wroclaw, Poland
rafal.chalupnik@pwr.edu.pl

***Abstract*—The aim of the paper was to analyze how to acquire and process EEG data with a simplified, commercially applicable EEG interface and to check whether it is possible to recognize human emotions with it. The EEG data gathering station was built and the data was gathered from the subjects. Then, the data was processed to apply it to the classifier training. The AutoML software was used to find the best ML model, and it was also built manually to prove the output accuracy was reliable and there was no overfitting. The AutoML experiment has shown that the best classifier was the boosted decision tree algorithm, and building it manually resulted in an accuracy of recognizing four distinct emotions equal to 99.80%.**

***Index Terms*—EEG; emotion recognition; machine learning; data acquiring**

I. Introduction

There are various studies regarding detecting emotions using electroencephalography (EEG). However, these studies use full-scale EEG devices, which can be difficult to use for commercial purposes [1] [2]. As the market is being filled with more and more devices that are smaller and more convenient [3] than the traditional EEG interface [4], such as NeuroSky MindWave Mobile 2 [5] [6], it is worth trying to acquire the data necessary for building the classifier with such devices and checking the achievable accuracy.

In this paper, we have planned the data acquisition and processing. The data acquisition process was defined and executed to gather the data for further processing, which included filtering by attention level above the preset threshold and signal smoothing by applying Simple Moving Average technique, which replaces the signal value at any given point in time with the average of the neighbors. Such experiment was completed successfully with the achieved accuracy of the built classifier of 99.80%.

The remainder of this paper is structured as follows: first, the available multiclass classification trainers are presented. Then, the stimulus set is prepared and the data acquisition process is described. After that, we described the methods used for signal processing, and finally, the experiment was performed, along with the results and conclusions drawn from them.

II. Related works

Detecting emotions using EEG was part of several research papers; however, they used the full-scale EEG interface instead of the simplified one. In the research done in [7], it was decided to use the emotion model used in [8], which assumes a division into two groups: positive and negative. The group of subjects consisted of three women and three men around the age of 22. They have viewed 12 video clips of length around four minutes. The authors decided to use audiovisual stimuli, as they stimulate more than one sense of the subject. Recording of brain activity in such a scenario gives 310 characteristics in each sample (62 electrodes * 5 channels). Samples were obtained this way and they were preanalyzed. Results with a dominance value lower than 3 were rejected because it implies an insufficient stimulant effect on the subject.

In the research by Chi et al. [9], emotion recognition was done while listening to music. They decided to use a 2D emotion model used in [10] consisting of two factors: arousal and valence of the emotion [11]. The EEG interface had 32 electrodes, distributed evenly through all head surfaces. The authors have tested three approaches to this problem. The first was one multiclass Support Vector Machine (SVM) classifier directly returning the predicted emotion. The second was the SVM classifier per each emotion and selecting the one with the highest score. The third was the tree of SVM classifiers recognizing valence on the first level, then arousal on the second one. The results of the experiments lead to the conclusion that the best approach to be used is to build the classifier for each emotion separately and then aggregate their outputs. The authors achieved an accuracy of 92.57%. In this paper, we conducted a similar experiment, but with the usage of the simplified, more convenient, and commercially available EEG interface to see whether it could achieve similar results.

III. Multi-class classification trainers

In the following section, the multiclass classification trainers are explained.

A. Averaged Perceptron Trainer

The single perceptron predicts the value by estimating the separating hyperplane. Let us say that there is a sample represented by a feature value vector, as shown in Equation 1.

$$F = [f_0, f_1, ..., f_{D-1}] \tag{1}$$

The perceptron simply determines which side of the hyperplane is the feature vector located. It is described by the sign of the weighted sum of the feature vectors, as shown in Equation 2, where $w*$ values are the weights of the perceptron, and the $b*$ are the biases.

$$y = \sum_{0}^{D-1} (w_i * f_i) + b_i \tag{2}$$

$$y - weighted\, sum$$

The learning process starts with initial weights (the best approach is to set them randomly). For each training sample, the weighted sum is calculated. If the sign of the predicted value is the opposite of the real one, the weights are updated by adding or subtracting the current sample features, multiplied by the learning rate and by the gradient of the loss function (Equation 3).

$$w_{t+1} = w_t \pm F * \alpha * l \tag{3}$$

$$w_{t+1} - new\, weights$$

$$w_t - old\, weights$$

$$F - feature\, vector$$

$$\alpha - learning\, rate$$

$$l - loss\, function\, value$$

The Averaged Perceptron model is based on a set of perceptrons. Each sample is processed with every perceptron, and the final prediction is based on the sign of the average output from all perceptrons.

B. Fast Forest Binary Trainer

Decision trees are models that are based on simple tests executed in sequence. The prediction is made by finding a similar input in the training dataset and returning their output label. Each node of the binary tree is representing a simple test to perform on the input data, and the output decision is reached by traversing the tree and finding the leaf node representing the output.

There are several advantages of decision trees. They are efficient in terms of computation and memory usage, both during the training phase and using the trained classifier. Moreover, they can represent the boundaries that cannot be resolved by linear decision (e.g., perceptron).

This particular trainer is a random forest implementation - it builds an ensemble of decision trees and then aggregates the output to find a Gaussian distribution that is the closest one to the combined distribution of aggregated trees. Such an approach provides better coverage and accuracy than single decision trees.

C. Fast Tree Binary Trainer

This trainer uses the efficient implementation of Multiple Additive Regression Trees (MART) gradient-boosting algorithm. It is building every decision tree using a step-by-step approach and using a predefined loss function to measure the error and correct for each step.

MART algorithm uses an ensemble of regression trees, which is a decision tree that contains scalar values in each leaf. The decision can be presented as a binary tree-like flow, where every node decides which of the two children should be used based on one of the features from the input.

The tree ensemble is constructed by computing a regression tree for each step that is an approximation of the loss function gradient and then adding it to the previous tree to minimize the loss function value of the new tree.

D. LBFGS Logistic Regression Binary Trainer

This trainer is using the optimization technique based on the Limited memory Broyden-Fletcher-Goldfarb-Shanno method (L-BFGS). It is a quasi-Newtonian method that is used to replace the Hessian matrix, which is computation-expensive, with an approximation.

Linear logistic regression is a variant of the linear model. It assumes the mapping of the feature vector into a scalar via the scoring function:

$$y(x) = w^T x + b = \sum_{j=1}^{n} w_j x_j + b \tag{4}$$

Since the approximation uses a limited number of states in history to designate the direction of the next step, it is convenient to solve problems having high-dimensional features. The user can set the number of stored historical steps and thus balance between a better approximation and lower cost per step.

E. LBFGS Maximum Entropy Multiclass Trainer

This model is a generalization of linear logistic regression. It can, however, be used in multiclass classification problems, while the regression can only solve binary ones.

This trainer assigns to each class a coefficient vector:

$$w_c \in R^n \tag{5}$$

and bias:

$$b_c \in R \tag{6}$$

Next, each class's score is calculated:

$$y_c = w_c^T x + b_c \tag{7}$$

The probability of the sample belonging to a given class can be defined in the following way:

$$P(c|x) = \frac{e^{y_c}}{\sum^m e^{y_c}} \tag{8}$$

F. Light GBM Multiclass Trainer

This Gradient Boosting Machine (GBM) trainer is an implementation of a gradient boosting framework that uses tree-based learning algorithms [12]. The major advantages of this trainer are that it achieves higher efficiency in shorter training time. Furthermore, it is using less memory and provides higher accuracy than similar algorithms.

G. Linear SVM Trainer

Linear SVM is another trainer that relies on finding a hyperplane in the feature space to perform binary classification. As in previous examples, the side of a hyperplane is defined by sign of the equation:

$$y = \sum_{1}^{N}(w_i * x_i) + b_i \tag{9}$$

However, the SVM model builds a representation of training samples as points in the space and the objective is to create a wide gap between points representing particular classes as possible.

H. One Versus All Trainer

One Versus All strategy assumes having a binary classification algorithm for each class. Such a classifier predicts the output class by evaluating all binary classifiers and selecting the result which has the highest score.

In ML.NET [13], such trainer can be used to concatenate binary classifiers to perform multiclass classification. This way, the developer can create a complex model, e.g., using 4 Fast Tree algorithms to achieve 4-class classification.

I. SDCA Maximum Entropy Multiclass Trainer

The Stochastic Dual Coordinate Ascent (SDCA) trainer is dedicated to multiclass classification usage. Assuming that there are C classes and N features in a particular sample, this algorithm assigns to every class a coefficient vector $w_c \in R^n$ and bias $b_c \in R$. For the feature vector, the value y_c is calculated for each class.

$$y_c = w_c^T x + b_c \tag{10}$$

Then, the probability of the feature vector belonging to a particular class is calculated in the following way:

$$P(c|x) = \frac{e^{y_c}}{\sum_1^C e^{y_{c_i}}} \tag{11}$$

J. Symbolic SGD Logistic Regression Binary Trainer

This trainer, also in its core is using the hyperplane to divide the samples represented as points in space. However, it has one fundamental difference.

While most of the algorithms that are using Stochastic Gradient Descent (SGD) are sequential, which means they are using the result of the previous step to process the current one, this algorithm is training the local models on separate threads and then, the probabilistic model combiner is trained to aggregate the models and provide the same output that the sequential algorithms would produce.

IV. Data acquisition and processing

As it is a common approach in related works, it has been decided to use audiovisual stimuli to invoke the particular emotion of a subject while stimulating more than one sense. This approach allows us to stimulate different parts of the brain. The best approach seems to be using music videos [14], as:

- they are fulfilling the audiovisual stimulus criterium,
- music is known to invoke human emotions effectively,
- videos are often well synchronized with music.

As the experiment is conducted based on the two-dimensional emotion model [15] [16], there is a need to find four music videos that would be related to each of the emotions: anger, depression, relaxation and happiness.

To build the classifier, the EEG data needed to be acquired and processed. NeuroSky MindWave Mobile 2 EEG interface is returning the raw data in its own metrics [17], and such data is already split into EEG spectres, This section is describing the two parts of the gathering data problem: how the data gathering station was built, and how the process looked like.

A. Data gathering station

Fig. 1. Data gathering station

To collect the EEG interface data necessary to conduct this experiment, the data gathering station was assembled, as visible in Figure 1. It consists of four most important factors: video stimuli, displayed on the monitor placed directly before the subject; audio stimuli, played through the headphones to reduce the noise around the station and improve the quality of the gathered data; NeuroSky MindWave Mobile 2 EEG interface [18], which is recording the EEG data; data gathering software, running on a separate computer facing away from the subject to not distract the user (for the station presentation, the photo is taken where it is visible for the subject).

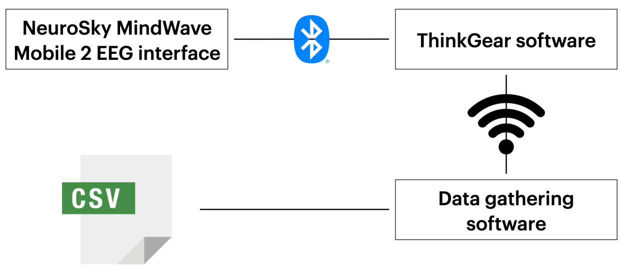

Fig. 2. Data gathering connection diagram

To collect the recorded EEG data and write it to the CSV file, the software needs to be used as presented in Figure 2. NeuroSky ThinkGear software is responsible for handling the Bluetooth connection to the EEG interface and provides this data over the Internet connection (during this experiment, the connection was used only in the localhost scope). Then, the dedicated data gathering software was written that connects to the ThinkGear API, downloading the EEG data and saving it to the CSV file.

B. Data gathering process

The study group consisted of 6 people, both women and men, of age from 15 to 45 years. Each subject had the EEG MindWave interface put on, along with headphones to reduce the outside stimuli that could negatively affect the experiment results.

For each of the stimuli, the subject watched the video for about five minutes, then was allowed to rest for about one minute to clear the mind, so the next recording is not affected by the previous one.

The subject is not told about the emotion that is currently recorded. In a related work, Nie et al [7] were using self-assessment manikin (SAM) to make the subject describe the emotion he was feeling, which later was classified as positive or negative. Such manikin consists of valence and arousal measures, which in our experiment are part of the two-dimensional emotion model. The third measure is the dominance measure, which was used to indicate whether the emotion was felt precisely and deeply or too slightly.

The idea of using SAM was deliberately discarded [19], as three measures are already included, and the third one will be achieved in another way, which will be described later in this work. Moreover, such an approach makes it possible to check in this experiment whether the emotions felt by different people have something in common, even if we are giving it different names. The proposed two-dimensional model has the advantage of not giving name to each of the emotions, we can describe them as valence/arousal positivity/negativity.

Figures from 3 to 10 represent the example visualization of the gathered data for each emotion or each spectrum in the domain of time. The Y axis represents the ASIC_EEG_POWER custom unit, which can be represented as $\frac{V^2}{Hz}$. [17].

Fig. 3. Visualization of the Alpha High EEG spectrum for anger

Fig. 4. Visualization of the Alpha Low EEG spectrum for anger

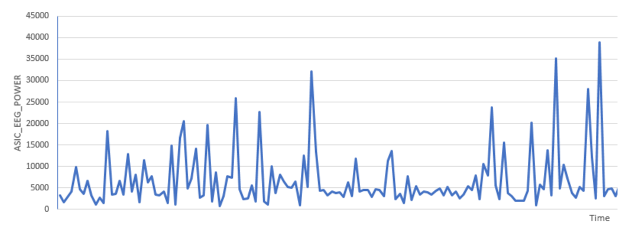

Fig. 5. Visualization of the Beta High EEG spectrum for anger

C. Processing data - filtering

In a related work, Nie et al [7] were using SAM manikins to describe i.a. strength of the felt emotion. As such manikins can be nonintuitive to the subjects, and such a measure does not have to match the reality, it was decided to filter the data using eSense measures provided by the MindWave EEG interface.

The eSense data consists of three measures: Attention, Meditation and Blink. To filter out the not-applicable data,

Fig. 6. Visualization of the Beta Low EEG spectrum for anger

Fig. 7. Visualization of the Gamma High EEG spectrum for anger

Fig. 8. Visualization of the Gamma Low EEG spectrum for anger

all samples with the attention level below 50 (on a scale from 0 to 100) were discarded. Such a filter can discard all data when the subject was not ”concentrated” enough, e.g., there was a noise in the environment that got to the subject through the headphones or there was a fly in the room that distracted the subject.

Using SAM manikins to check the dominance of the emotion has two major disadvantages:

- subject can answer that the emotion was very dominant, but there were times that he was distracted,
- subject can answer that the emotion was not dominant because of distraction, e.g., 3 seconds - then the whole

Fig. 9. Visualization of the Delta EEG spectrum for anger

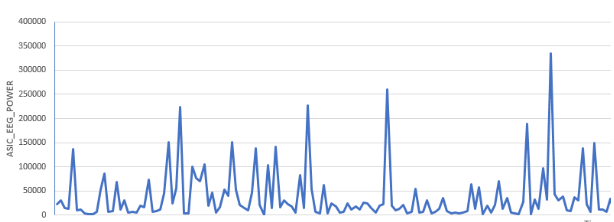

Fig. 10. Visualization of the Theta EEG spectrum for anger

Fig. 11. Original signal

Fig. 12. Smoothed signal

recording was wasted.

Filtering the recorded samples through their attention score allows removing these two disadvantages, as such a measure is independent of the subject consciousness, and therefore cannot be falsified too easily.

D. Processing data - signal smoothing

The data received from the MindWave EEG interface consisted of eight measures:

- low alpha,
- high alpha,
- low beta,
- high beta,
- low gamma,
- high gamma,
- delta,
- theta.

The problem that appeared is that the signal received from the interface is quite noisy, and trying to train the classifier using a singular sample could result in low efficiency. To avoid that, the received output was smoothed, as presented in Figures 11 and 12.

There are numerous ways to smooth the signal [20] [21]. For this experiment, the Simple Moving Average algorithm [22] has been chosen, which uses the sliding window to calculate the average value of the signal.

For example, let us assume the window width $w = 4$ and step $s = 2$ for the signal values:

$$\lambda = [4, 9, 6, 5, 2, 1, 3, 10, 8, 7] \tag{12}$$

then the new signal will consist of four average values:

$$\lambda\prime = [\frac{\sum_1^4 \lambda_n}{4}, \frac{\sum_3^6 \lambda_n}{4}, \frac{\sum_5^8 \lambda_n}{4}, \frac{\sum_7^{10} \lambda_n}{4}] \tag{13}$$

which after calculation, will be

$$\lambda\prime = [6, 3.5, 4, 7] \tag{14}$$

TABLE I
AUTOML: TOP 5 TESTED CLASSIFIERS

	Trainer	MicroAccuracy	MacroAccuracy	Duration
1	FastTreeOva	0.9233	0.9264	19.4
2	LightGbmMulti	0.9000	0.9001	14.3
3	LightGbmMulti	0.8185	0.8199	17.5
4	FastForestOva	0.8027	0.8051	53.1
5	FastTreeOva	0.8006	0.8035	24.3

TABLE II
SUMMARY CONFUSION MATRIX FOR TESTED BEST CLASSIFIER

	Anger	Depression	Happiness	Relaxation	Recall
Anger	1326	1	2	0	0.97743
Depression	1	1280	2	0	0.97662
Happiness	0	0	1267	1	0.99211
Relaxation	2	2	0	1024	0.96109
Precision	0.97743	0.97662	0.96853	0.99024	

E. Gathered data from time perspective

While it is natural to consider the gathered data as changing in time, as it is recording brain waves, this experiment was conducted using singular samples from the recordings to check whether the time is needed. The ThinkGear Web API [23] used to collect the samples was returning the data 1 sample per second, and that is the granularity used in this experiment. To reduce the noise available in the data, a smoothing filter was used as described earlier, however, the resulting samples were used one-by-one in building the classifier, without keeping the information about the relationship between them. This way, the experiment can check whether it is more important to check the brain waves fluctuations over time, or to observe the plain values of EEG spectres to correctly recognize and classify emotions.

V. EXPLORING THE CLASSIFIERS

The following section describes the classifier exploration with ML.NET.

A. Exploring classifiers using AutoML

After the data processing, the AutoML framework was used to determine the best ML.NET model for the dataset. AutoML is a feature of ML.NET foundation developed by Microsoft. It is a key advantage is that it allows quickly browsing the available classifiers and their achievable accuracy without writing the code. Given 5 minutes of time, AutoML designated the best classifier and the summary of the tested classifiers (top 5 shown in Table I).

B. Building the best classifier with ML.NET

During the experiment, the best classifier found by AutoML was FastTreeOva. To ensure that the achieved accuracy is reliable (e.g, there was no overfitting, the test data were not the same as the training ones, etc.), the FastTreeOva classifier has been built manually, using ML.NET components.

Firstly, the training and test data were prepared and the cross-validation technique was applied.

1) Load the samples from CSV file.
2) Randomize the samples order.
3) Split data into 10 batches.
4) For each batch:
    a) Form the test data from the selected batch.
    b) Form the training data from the rest.
    c) Run the training and gather output.
5) Gather average values from the output of each batch.

The FastTreeOva model constructed with ML.NET components was used in the cross-validation. The results were presented in the confusion matrix, presented in Table II. The real emotions are represented by rows, and the predicted emotions are represented as columns.

The accuracy (a) of the classifier can be presented as a ratio of correctly classified samples (s) to all the samples(Ω):

$$a = \frac{s}{\Omega} = \frac{4897}{4908} \approx 99.8\% \qquad (15)$$

The FastTreeOva model has been proven to achieve an accuracy of 99.8%. It is worth noticing that the accuracy was achieved without filtering the data by attention level, which might indicate the difficulty of hiding the felt emotions.

VI. CONCLUSION AND FUTURE WORK

Data that was gathered during the experiment was processed by applying two techniques: eliminating samples with the attention level below the preset threshold and signal smoothing using Simple Moving Average algorithm.

Taking into account the purpose of the study, which is to check whether it is possible to recognize emotions using the simplified EEG interface and to see how many emotions it is achievable to distinguish, the purpose was fulfilled.

The conducted experiment has shown that it is possible to predict 4 distinct emotions using NeuroSky MindWave Mobile 2 device with an accuracy of 99.80%. It seems that filtering the gathered data by attention level did not impact the final results, as opposed to the applied signal smoothing technique, which helped to achieve such an accuracy.

It would be promising to research the improvement ratio for each EEG spectrum and compare it with the accuracy of the classifier built solely on this spectrum, showing that there certainly is a noticeable difference in spectrum influence between the two groups: alpha, beta, gamma and delta, theta. The difference between the calculated values could be observed, which could indicate that some EEG spectres have higher influence on emotion recognition. Furthermore, the built FastTreeOva model research could define the range of each EEG spectrum that is correlated with the particular emotion.

REFERENCES

[1] S. Saganowski, A. Dutkowiak, A. Dziadek, M. Dzieżyc, J. Komoszyńska, W. Michalska, A. Polak, M. Ujma, and P. Kazienko, "Emotion recognition using wearables: A systematic literature review-work-in-progress," in *2020 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*. IEEE, 2020, pp. 1–6.

[2] Y. Yin, X. Zheng, B. Hu, Y. Zhang, and X. Cui, "Eeg emotion recognition using fusion model of graph convolutional neural networks and lstm," *Applied Soft Computing*, vol. 100, p. 106954, 2021.

[3] S. Saganowski, A. Dutkowiak, A. Dziadek, M. Dzieżyc, J. Komoszyńska, W. Michalska, A. Polak, M. Ujma, and P. Kazienko, "Emotion recognition using wearables: A systematic literature review - work-in-progress," in *2020 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, 2020, pp. 1–6.

[4] E. Niedermeyer, *Electroencephalography : basic principles, clinical applications, and related fields*. Philadelphia: Lippincott Williams & Wilkins, 2005.

[5] K. Bialas and M. Kedziora, "Analiza mozliwosci sterowania aplikacja mobilna za pomoca interfejsu mozg-komputer," in *XII Ogolnokrajowa Konferencja Naukowa MLODZI NAUKOWCY W POLSCE – BADANIA I ROZWOJ (paper accepted)*, 2020.

[6] "NeuroSky official website," http://neurosky.com, accessed: 2020-06-08.

[7] D. Nie, X. Wang, L.-C. Shi, and B.-L. Lu, "Eeg-based emotion recognition during watching movies," 06 2011, pp. 667 – 670.

[8] P. Ekman and D. Cordaro, "What is meant by calling emotions basic," *Emotion Review*, vol. 3, no. 4, pp. 364–370, 2011.

[9] Y.-P. Lin, C.-H. Wang, T.-L. Wu, S.-K. Jeng, and J. Chen, "Eeg-based emotion recognition in music listening: A comparison of schemes for multiclass support vector machine," 04 2009, pp. 489–492.

[10] M. N. Shiota, "Ekman's theory of basic emotions," *The SAGE Encyclopedia of Theory in Psychology*, 2016.

[11] C. Osgood, G. Suci, and P. Tenenbaum, *The Measurement of meaning*. University of Illinois Press, 1957.

[12] "LightGBM documentation website," https://lightgbm.readthedocs.io/en/latest/index.html, accessed: 2020-06-14.

[13] "ML.NET official webpage," https://dotnet.microsoft.com /apps/machinelearning-ai/ml-dotnet, accessed: 2020-05-21.

[14] D. Plass-Oude Bos, "Eeg-based emotion recognition," *The Influence of Visual and Auditory Stimuli*, 01 2006.

[15] D. L. Schacter, *Psychology*. New York, NY : Worth Publishers, 2011.

[16] P. Ekman, "What scientists who study emotion agree about," *Perspectives on Psychological Science*, vol. 11, no. 1, pp. 31–34, 2016.

[17] "NeuroSky EEG Band Power values meaning," http://support.neurosky.com/kb/development-2/eeg-band-power-values-units-amplitudes-and-meaning, accessed: 2020-06-14.

[18] "NeuroSky MindWave Mobile 2 official product page," https://store.neurosky.com/pages/mindwave, accessed: 2020-06-08.

[19] M. Dzieżyc, J. Komoszyńska, S. Saganowski, M. Boruch, J. Dziwiński, K. Jabłońska, D. Kunc, and P. Kazienko, "How to catch them all? enhanced data collection for emotion recognition in the field," in *2021 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*, 2021, pp. 348–351.

[20] H. He, Y. Tan, J. Ying, and W. Zhang, "Strengthen eeg-based emotion recognition using firefly integrated optimization algorithm," *Applied Soft Computing*, vol. 94, p. 106426, 2020.

[21] C. Wei, L.-l. Chen, Z.-z. Song, X.-g. Lou, and D.-d. Li, "Eeg-based emotion recognition using simple recurrent units network and ensemble learning," *Biomedical Signal Processing and Control*, vol. 58, p. 101756, 2020.

[22] C. Clapham, J. Nicholson, and J. Nicholson, *The Concise Oxford Dictionary of Mathematics*, ser. Oxford Paperback Reference. OUP Oxford, 2014. [Online]. Available: https://books.google.pl/books?id=c69GBAAAQBAJ

[23] "NeuroSky ThinkGear chip technical specifications," http://neurosky.com/biosensors/eeg-sensor/, accessed: 2020-06-08.