Proceedings of the Federated Conference on Computer Science and Information Systems pp. 1621–1626
DOI: 10.15439/2016F263
ACSIS, Vol. 8. ISSN 2300-5963

Towards detecting programmers' stress on the basis of keystroke dynamics

Agata Kołakowska
Gdańsk University of Technology
Faculty of Electronics, Telecommunications and Informatics
Email: agatakol@eti.pg.gda.pl

***Abstract*—The article describes the idea of detecting stress among programmers on the basis of keystroke dynamics. An experiment with a group of students of artificial intelligence classes was performed. Two samples of keystroke data were recorded for each case, the first while programming without stress, the second under time pressure. A number of timing and frequency parameters were calculated for each sample. Then statistical analysis was performed to evaluate the significance of keystroke parameters changes. It turned out that some of the defined features might be indicators of being stressed.**

I. Introduction

STRESS is nowadays present in most occupations. IT professionals is one of the groups that is exposed to stress the most. There are many reasons for such situation, e.g. deadlines, pressure from clients, high workload [1]. Some research has been made to show that emotions have impact on software developers' productivity [2]. Moreover, negative emotions such as stress may also influence employers' physical health and their mental state. Therefore it is worth detecting them and, if possible, reacting adequately to alleviate negative effects.

Affective computing, a domain intensively explored in recent times, meets the needs of the above problem. It "relates to, arises from, and influences emotions" [3]. Affective applications implement not only emotion recognition methods, but also interpret and react to the recognized affective states. One of possible areas of applying affective methods is software engineering, where emotion recognition may be used to improve software development process [4].

Various input channels may be considered while designing an emotion recognition tool, i.e. visual [5], depth [6], audio [7], textual [8], physiological [9], standard input devices [10]-[15] or multi-modal input [16]. Not all of them seem practical in any situation, e.g. physiological signals not only require specialized devices and disturb computer users, but these measurements are also disturbed by motions typical for human-computer interaction [17].

Analyzing keystroke dynamics and mouse movements is completely non-intrusive, as it does not require any special hardware and may be invisible for users. The aim of this study is to answer a question whether or not stress caused by time pressure influences programmers' keystroke dynamics.

II. Related work

There is a number of research studies on recognizing emotions on the basis of keystroke dynamics [10]. They deal with a number of problems, i.e. inducing emotions, collecting and labeling data samples, defining and calculating characteristic features and finally training and testing the models. Various solutions are designed to be applied for different emotional states. In some cases a number of emotions are recognized. Other works focus on detecting one selected emotional state.

In [11] for example an experiment on recognizing stress on the basis of keystroke and linguistic features has been presented. The stress was induced by giving some stressful tasks to the participants. Several machine learning techniques were applied to solve that task, i.e. SVM, k-NN, neural networks, decision tress and AdaBoost. It was possible to recognize cognitive and physical stress with accuracies 75% and 62.5% respectively. The authors also showed there was a strong relation between the emotional state and the use of backspace, delete, end, arrow keys and also the time per keystroke and pause length.

Another example of stress recognition was presented in [12]. In this case stress was only one of fifteen emotional states recognized during usual computer activities, e.g. using word processor, sending e-mails. Applying decision trees let recognize some of the emotions with high accuracies 77.4-87.8% (confidence, hesitance, nervousness, relaxation, sadness, tiredness) if the recognition was based on fixed texts. Stress was not one the best recognized emotions.

Detecting stress is an important issue in e-learning systems. A framework for stress detection system applied among Moodle students has been proposed in [13]. The authors of that idea not only chose to analyze keystrokes measured as frequency and intensity of keyboard usage, but

This work was supported in part by Polish-Norwegian Financial Mechanism Small Grant Scheme under the contract no Pol-Nor/209260/108/2015 as well as by DS Funds of ETI Faculty, Gdansk University of Technology.

also information from mouse, webcam, touch screen and accelerometer.

In [14] an intelligent tutoring system, which recognizes boredom and frustration on the basis of keystroke and mouse dynamics, is presented. The data in this study were gathered from students learning a programming language by performing programming tasks and then filling a short questionnaire about the level of the two mentioned emotional states. The obtained accuracies for boredom and frustration were over 83% and 74% respectively.

Another approach to stress detection has been presented in [15] where the possibility of using a pressure-sensitive keyboard and a capacitive mouse to discriminate between stressful and relaxed conditions was investigated. It turned out that under stress the typing pressure increased and more contact with the surface of mouse was observed.

This study focuses on the possibility of detecting stress among programmers by analyzing their keystroke dynamics. The essential assumption of the presented experiment was to perform it in a real-world stressful situation.

III. Experiment Design and Methodology

A. Research objective

A hypothesis stated in this research is that a programmer's keystroke dynamics change depending on whether or not he or she works under stress caused by time pressure. To verify this idea a proper experiment among programmers should be performed. This study might be treated as a preliminary experiment enabling to reveal all issues necessary to be taken into account before starting to cooperate with a group of programmers in their real life working environment.

B. Participants

Two groups of computer science students have been asked to take part in the experiment in order to collect data. They attended a course on artificial intelligence, where different tasks were solved using Matlab. Each group took part in three sessions performing three different tasks. Only those students, who agreed to participate in the experiment, were collecting the data.

C. Data collection procedure

To collect data coming from keyboard, an application running in background was used [18]. The students started the application themselves at the beginning of a session. The application did not disturb them in any way. Each session was organized in a similar way, as it always happened during that course. The fact, that some keystroke data was gathered, did not cause any changes in the standard way of class conducting, already experienced by the students. During the first part they were supposed to write a piece of code in Matlab script to implement a fragment of an artificial intelligence method. They were given instructions and prompts, both verbal and on the blackboard. They were given clues, when they asked for. Then another task from the same domain, was given to them. This time the students were supposed to solve it individually. They were given specified amount of time for this and they were evaluated at the end. All keystrokes were recorded by the application running in the background. Although the students knew about data recording, they did not know the details of the experiment, especially the stated hypothesis. They only knew the keystrokes would be analyzed later. This was to prevent from intentional change of keystroke behaviors.

16 students took part in the data collection phase, some of them in all three sessions, some in two, and some in one session only. Eventually, 36 data samples were collected, each consisting of two parts. The first part of a sample contained keystroke data from the first part of a session, when stress should not have appeared. The second subsample contained data from the second part of a session, i.e. when the students were working under time pressure.

D. Data preprocessing

The first stage of data processing was segmentation. The whole sequence of keystrokes was split into many shorter sequences depending on the presence of pauses. No one types continually. Every programmer types and stops for a while or a longer time. To identify the limits of typing sequences an idle threshold has been introduced. If the time between depressing a key and pressing the next one exceeded the idle threshold, then the split was made. The greater the value of the threshold the longer keystroke sequences were extracted. All timing characteristics described later in this section were calculated regarding the extracted partial sequences. The segmentation was performed for different values of the idle threshold: 0.3 s, 0.5 s, 0.7 s, 1 s. It lead to creating four sets of data to be analyzed.

E. Feature extraction

After segmenting the data, feature extraction procedure was performed. A number of parameters was calculated from raw data. They may be divided into the following groups: digraph features, trigraph features, special digraph features, frequency features and typing speed.

Digraph and trigraph features are timing characteristics for two-key and three-key sequences. They are all based on parameters commonly used in keystroke dynamics analysis, i.e. dwell time (the time a key is pressed), the time between releasing a key and pressing the next one, the duration of key sequences (the time between pressing the first and depressing the last key in a sequence) and the times between subsequent key presses. Moreover, the number of events for a digraph or trigraph was also calculated. These are the numbers of all key down an key up events in a graph, so it is usually 4 for a digraph and 6 for a trigraph. Sometimes, especially when a user types quickly, it happens that a user presses the next key before depressing one. In such cases additional events may appear between those coming from a graph and then the values for these attributes may differ from 4 or 6. A data sample contains many digraphs and trigraphs. The parameters were calculated for all of them and then their mean values and standard deviations were saved. The detailed list of digraph and trigraph features is presented in Table I.

TABLE I.
PARAMETERS CALCULATED FROM RAW DATA

Feature subsets	Description (feature identifier)
12 digraph features (mean and standard deviation calculated for each parameter)	dwell time for the first key (di_01, di_02)
	dwell time for the second key (di_03, di_04)
	time between pressing the first and the second key in a digraph (di_05, di_06)
	time between depressing the first and pressing the second key (di_07, di_08)
	digraph duration (time between pressing the first and releasing the second key) (di_09, di_10)
	number of events for a digraph (di_11, di_12)
18 trigraph features (mean and standard deviation calculated for each parameter)	dwell time for the first key (tri_01, tri_02)
	dwell time for the second key (tri_03, tri_04)
	dwell time for the third key (tri_05, tri_06)
	time between pressing the first and the second key in a trigraph (tri_07, tri_08)
	time between pressing the second and the third key in a trigraph (tri_09, tri_10)
	time between depressing the first and pressing the second key (tri_11, tri_12)
	time between depressing the second and pressing the third key (tri_13, tri_14)
	trigraph duration (time between pressing the first and releasing the third key) (tri_15, tri_16)
	number of events for a trigraph (tri_17, tri_18)
10 special digraph features (mean and standard deviation calculated for the timing parameters)	time between pressing the first and the second key in a digraph starting from the left shift (di_L_01, di_L_02)
	duration of digraph starting from the left shift (time between pressing the first and releasing the second key) (di_L_03, di_L_04)
	percentage of times when the left shift starting a digraph is released before releasing the second key (di_L_05)
	time between pressing the first and the second key in a digraph starting from the right shift (di_R_01, di_R_02)
	duration of digraph starting from the right shift (time between pressing the first and releasing the second key) (di_R_03, di_R_04)
	percentage of times when the right shift starting a digraph is released before releasing the second key (di_R_05)
17 frequency features	frequency of using the following keys: enter (freq_ENTER), spacebar (freq_SPACE), tab (freq_TAB), backspace (freq_BCKSPC), delete (freq_DEL), up (freq_UP), down (freq_DOWN), left (freq_LEFT), right (freq_RIGHT), left shift (freq_LSHIFT), right shift (freq_RSHIFT), home (freq_HOME), end (freq_END), pgup (freq_PGUP), pgdn (freq_PGDN), percent (freq_PERC)
	number of capital letters to the total number of letters (freq_CAPS)
typing speed	average number of keystrokes per second (speed)

Some digraphs have been treated as special sequences in the case of this applications. These are digraphs containing either left or right shift key as the first one. Digits, operators, brackets, which require using shift to enter them, are common characters while programming. Therefore some digraph parameters were calculated for digraphs starting from the left and the right shift. The detailed list of these characteristics is presented in Table I.

Another group of features are frequency parameters. In contrast to digraphs and trigraphs they do not describe keystroke rhythm. Some of them may indicate the way users make corrections (backspace, delete), move across the code (pgup, pgdn, home, end, up, down, left, right), take care of programming style issues. For example the percent (%) symbol is used in Matlab to start a comment. The frequency was calculated as the number of a selected symbol to the total number of keystrokes. One of the frequency features was calculated in a different way, i.e. the number of capital letters to the total number of letters. The complete list of frequency features is shown in Table I.

Finally, the typing speed, which indicates the number of keystrokes per second, was calculated.

The total number of parameters calculated was 58, which is rather high. Some of the features are redundant. e.g. digraph duration is the sum of dwell times for the two keystrokes and the time between depressing the first and pressing the second key. However, in this study, this set of features is not going to be used for classification purposes for example. That is why dimension reduction is not priority. The aim of this research is to find out whether some of the parameter changes might be caused by stress. Therefore statistical analysis was performed for each of the proposed 58 features.

F. *Statistical apparatus for data analysis*

To verify the stated research hypothesis a statistical test should be applied. In this case dependent t-test was used. This is a proper test when the data from a person are gathered several times and their changes are investigated. The advantage of analyzing the changes, not the values themselves, is also the fact that in this way differences between the subjects may not be taken into account. In the case of this experiment the difference between the participants and between keyboard types are hidden by analyzing the values changes only.

The keystroke statistics of a person performing a task are measured twice, i.e. while coding without being evaluated and then while writing under time pressure a piece of code to be evaluated. The question is whether the changes of the values of keystroke parameters are significant. To answer this question the dependent t-test of the following form may be used:

$$t = \frac{\bar{d}}{s_d}\sqrt{n-1}$$

where $\overline{d}$ is the mean difference between the values of two measures obtained in two situations; s_d is the standard deviation of the differences; n is the number of degrees of freedom, i.e. the number of pairs of samples, for which the difference is calculated. Because of the fact that no assumption is made on the direction of the observed changes, i.e. keystroke parameters may either increase or decrease, the applied t-test should be two-tailed.

IV. Analysis of Results

A. Results

The values of t-test were calculated for all defined keystroke parameters and the corresponding levels of significance (p-values) have been presented in Table II. Only the features, for which t-statistic exceeded critical value for p=0.05 have been shown.

The results are divided into four sections obtained for different values of the idle threshold. Moreover, each section (table column) is divided into three parts depending on the observed level of significance of parameters' changes. The three subsections correspond to $p \leq 0.001$, $0.001 < p \leq 0.01$ and $0.01 < p \leq 0.05$ respectively. As it can be seen from Table II, about half of the parameters change significantly. The differences among the results obtained for different values of the idle threshold are not very clear. The number of significantly changed features is lower for the lowest threshold of 0.3 s. It can be noted that the subsets of parameters are quite similar. Most features appear in all four sections (columns). It is possible to indicate the parameters which seem to be the best indicators of keystroke dynamics changes. These parameters are potential candidates to be analyzed in a stress detection system.

Table II.
Significance level for feature changes obtained for four sets of data generated for different idle threshold values

idle threshold = 0.3 s		idle threshold = 0.5 s		idle threshold = 0.7 s		idle threshold = 1 s	
Feature	**p-value**	**Feature**	**p-value**	**Feature**	**p-value**	**Feature**	**p-value**
di_09	0.00002	di_09	0.00000	di_09	0.00000	tri_09	0.00001
Speed	0.00029	tri_09	0.00000	tri_09	0.00003	di_07	0.00007
tri_09	0.00067	tri_10	0.00004	tri_13	0.00009	di_09	0.00013
di_07	0.00091	di_07	0.00005	di_07	0.00013	tri_13	0.00013
tri_13	0.00102	tri_12	0.00007	Speed	0.00033	di_03	0.00018
tri_15	0.00133	tri_13	0.00013	tri_03	0.00041	tri_03	0.00022
di_03	0.00140	Speed	0.00030	tri_15	0.00067	tri_15	0.00048
di_L_04	0.00155	tri_15	0.00031	di_03	0.00080	di_L_04	0.00067
tri_03	0.00215	di_03	0.00045	di_01	0.00117	tri_10	0.00086
di_L_01	0.00244	di_01	0.00085	tri_10	0.00140	di_L_01	0.00166
di_R_03	0.00559	di_L_01	0.00419	di_L_01	0.00309	speed	0.00253
tri_10	0.00684	di_12	0.00517	di_L_04	0.00329	di_L_02	0.00477
di_L_02	0.00713	di_L_04	0.00770	freq_BCKSPC	0.00850	tri_12	0.00563
di_R_04	0.00849	di_11	0.00849	di_11	0.00853	di_11	0.00738
freq_BCKSPC	0.00850	freq_BCKSPC	0.00850	freq_RIGHT	0.01230	freq_BCKSPC	0.00850
di_R_01	0.00954	di_10	0.01120	di_12	0.01284	di_12	0.00996
freq_RIGHT	0.01230	tri_03	0.01198	tri_12	0.01471	tri_07	0.01028
di_R_02	0.01236	freq_RIGHT	0.01230	di_R_04	0.01509	freq_RIGHT	0.01230
tri_12	0.01727	tri_07	0.01346	di_L_02	0.01558	freq_LSHIFT	0.01734
freq_LSHIFT	0.01734	tri_18	0.01449	di_R_02	0.01567	tri_16	0.01839
di_01	0.01898	freq_LSHIFT	0.01734	freq_LSHIFT	0.01734	freq_CAPS	0.02016
freq_CAPS	0.02035	di_R_04	0.02000	tri_07	0.01748	tri_01	0.02154
di_05	0.02939	freq_CAPS	0.02013	freq_CAPS	0.02020	di_08	0.02166
freq_ENTER	0.03468	di_08	0.02463	tri_17	0.02610	di_L_03	0.02285
di_12	0.03816	tri_17	0.03151	tri_14	0.02651	tri_18	0.02444
di_R_05	0.04214	di_R_03	0.03301	di_10	0.03150	tri_17	0.02740
tri_04	0.04584	di_R_02	0.03353	tri_16	0.03293	tri_14	0.03079
		freq_ENTER	0.03468	freq_ENTER	0.03468	di_10	0.03454
		tri_14	0.03511	di_08	0.03483	freq_ENTER	0.03468
		di_L_02	0.03945	di_L_03	0.03896	di_R_02	0.03804
		tri_01	0.04220	tri_18	0.03910	di_R_04	0.03905
		di_02	0.04960			di_01	0.04539

TABLE III.
NUMBER OF FEATURES CHANGED SIGNIFICANTLY

p-value	Idle threshold			
	0.3 s	0.5 s	0.7 s	1 s
Task 1				
$p \leq 0.001$	0	0	0	0
$0.001 < p \leq 0.01$	2	2	2	2
$0.01 < p \leq 0.05$	0	3	0	0
Task 2				
$p \leq 0.001$	9	14	10	9
$0.001 < p \leq 0.01$	11	9	7	7
$0.01 < p \leq 0.05$	13	7	11	11
Task 3				
$p \leq 0.001$	3	3	7	6
$0.001 < p \leq 0.01$	4	8	6	8
$0.01 < p \leq 0.05$	7	5	6	5

Most of the top parameters are digraph and trigraph characteristics, e.g. mean digraph duration (di_09), mean time between depressing the first and pressing the second key in a digraph (di_07), mean dwell time for the second key (di_03), mean time between pressing the second and the third key in a trigraph (tri_09), mean dwell time for the second key in a trigraph (tri_03), mean trigraph duration (tri_15). Typing speed also turned out to change significantly. Some parameters calculated for digraphs starting from the left shift, i.e. mean time between pressing the left shift and the subsequent key (di_L_01) and standard deviation for digraph duration (di_L_04). Regarding the frequency features, it may be observed that only five keys seem to be worth taking into account. These are: backspace, right arrow, left arrow, enter. It confirmed some observations made in other studies [11].

The mentioned observations have been made on the basis of data coming from different people. It is possible that analyzing the results individually would let draw different conclusions. First of all the idle threshold could be adjusted regarding one's typing speed. Then the subsets of significantly changed parameters could be also found individually. However, it would require gathering more samples from one person performing different tasks.

B. Limitations

Although significant changes for some of the defined characteristics have been observed, these results should be analyzed with precaution. The experiment was not free from limitations.

The first issue, which should be discussed, is the difference between the results obtained for different tasks. As it has been mentioned in Section III C, the 36 samples were collected while performing three different tasks. The number of samples form the three tasks was 15, 12 and 8 respectively. The significance of feature changes was also estimated for each task independently. Table III contains numbers of features found to change significantly in each case. Only a few parameter changes turned out to be significant in the case of task 1, whereas for other two tasks there were more of them. One of the reasons for this difference is the fact that the level of difficulty of the three tasks was not the same. The first one was the easiest although it required more coding than in the second case. The second task was more difficult but in this case the students were supposed to spend some time on designing before starting coding so the amount of code written was smaller. The third task was the most difficult and it required the highest number of code lines to be written. Moreover, in all three cases the students were precisely instructed during the first part of the lesson. The blackboard was used to explain the details of the problems being solved. In the case of the second and the third task more instructions were given on the blackboard and it was possible to make use of it by copying some of the lines to students' code. The amount of rewriting in the first task was much lower. During the second, i.e. the stressful, part of the lesson, no lines to be copied were given on the blackboard. Although few lines could be written by copying from the blackboard, keystroke dynamics with rewritten fragments might be different than without it.

It should be also noted that some of the features turned out to be useless, e.g. the frequency parameters for the pgup, pgdn keys. It was because of the specificity of the three given tasks, which did not require writing many pages of code. However, in real programming environments, these parameters could be worth calculating. Possible different behaviors are worth paying attention, e.g. either pressing pgdn/pgup quickly many times or keeping it pressed for some time to move down/up.

Another limitation of this study is neglecting the fact that people are not equally prone to stress and they react to stress in different ways. There are some factors such as marital status, age, gender, income, experience, which have been found as having influence on individual stress level [1]. Some of these factors (e.g. age, experience, income) were not present in the case of the presented experiment, due to the peculiar study group of students attending the same class, but some of them still remained.

Finally, it has to be highlighted, that although the stressful situation was induced in a way, there is no certainty that the participants were really stressed, because they were not asked for any self-assessment at the end of each session. The only way to make sure that the task given to the students really induced stress would be applying one of numerous questionnaires which cover a wide range of symptoms induced by stress and are used in the field of psychology [19]. Another interesting approach would be incorporating some physiological measurements, which could be indicators of stress. However, this would require usage of special devices and coping with the problem of the sensitiveness of some biometric sensors to finger movements [16][17]. Thus it could not be implemented in an experiment performed in a real life situation as the one described.

I. Conclusions

The results of the presented survey may be treated as the preliminary ones, which could be useful in designing a deliberate experiment to be performed among programmers. Regarding the mentioned issues it can be noticed, that more factors should be taken into account to make sure on the influence of stress on programmers' keystroke dynamics. The presented results give some clues: the tasks performed with and without time pressure should be as similar in the sense of difficulty and length, as possible; effort should be made in order to ensure similar working conditions; the idle threshold should be adjusted individually depending on the typing speed. Finally, the experiment results should be also compared to the results of a proper psychological questionnaire.

Moreover, some other ideas could be explored. One of the most interesting ones is adding to the set of analyzed parameters the timing characteristics specially defined for a given programming language. It could be for example key words and also common sequences of symbols instead of calculating all digraph and trigraph parameters.

Another interesting idea is to incorporate information from mouse as well. There are some known studies on recognizing emotions from mouse movements [10]. Such analysis could be adapted to a given programming environment by tracking the way it is operated, e.g. using menus, moving across various windows etc.

The observed changes in keystroke dynamics are also worth investigating in another application, i.e. intelligent tutoring systems. Analyzing keystroke changes could be applied to detect specific situations, that might reduce the efficiency of the learning process.

References

[1] V. Sreecharan and M. Srinivasa Reddy, "A study on individual and interpersonal stress levels among software employees," *Int. Journal of Information Technology & Computer Sciences Perspectives*, vol. 2(4), pp. 711-716, 2013.

[2] M. R. Wróbel, "Emotions in the software development process," in *Proc. 6th International Conference on Human System Interaction*, Gdańsk, 2013, doi: 10.1109/HSI.2013.6577875.

[3] R. W. Picard, "Affective Computing", MIT Press, Cambridge, 1997.

[4] A. Kołakowska, A. Landowska, M. Szwoch, W. Szwoch, M. R. Wróbel, "Emotion Recognition and its Application in Software Engineering", in *Proc. 6th International Conference on Human System Interaction*, Gdańsk, 2013, doi: 10.1007/978-3-319-08491-6_5.

[5] S. V. Ioannou, A. T. Raouzaiou, V. A. Tzouvaras, T. P. Mailis, K. C. Karpouzis, S. D. Kollias, "Emotion recognition through facial expression analysis based on a neurofuzzy network," *Neural Networks*, vol. 18(4), pp. 423-435, 2005, doi: 10.1145/1980022.1980177.

[6] M. Szwoch, P. Pieniążek, "Facial Emotion Recognition Using Depth Data," in *Proc. 8th Int. Conf. Human System Interaction*, pp. 271-277, 2015, doi: 10.1109/HSI.2015.7170679.

[7] B. Schuller, M. Lang, G. Rigoll, "Multimodal emotion recognition in audiovisual communication", in *Proc. IEEE Int. Conference on Multimedia and Expo*, ICME, Lausanne, 2002, doi: 10.1109/ICME.2002.1035889.

[8] A. J. Gill, R. M. French, D. Gergle, J. Oberlander, "Identifying Emotional Characteristics from Short Blog Texts," in *Proc. 30th Annual Conference of the Cognitive Science Society*, 2008, pp. 2237-2242.

[9] W. Szwoch, "Using Physiological Signals for Emotion Recognition," in *Proc. 6th International Conference on Human System Interaction*, Gdańsk, 2013, doi: 10.1109/HSI.2013.6577880.

[10] A. Kołakowska, "A review of emotion recognition methods based on keystroke dynamics and mouse movements," in *Proc. 6th International Conference on Human System Interaction*, Gdańsk, 2013, doi: 10.1109/HSI.2013.6577879.

[11] L. M. Vizer, L. Zhou, A. Sears, "Automated stress detection using keystroke and linguistic features," *Int. Journal of Human-Computer Studies* 67, pp. 870-886, 2009, doi: 10.1016/j.ijhcs.2009.07.005.

[12] C. Epp, M. Lippold, R. L. Mandryk, "Identifying emotional states using keystroke dynamics," in *Proc. Conf. on Human Factors in Computing Systems*, Vancouver, pp 715-724, 2011, doi: 10.1145/1978942.1979046

[13] M. Rodrigues, P. Novais, F. Fdez-Riverola, "An approach to assess stress in e-learning students," in *Proc. 11th European Conf. e-Learning*, pp. 461-467, 2012.

[14] A. Hernandez-Aguila, M. Garcia-Valdez, and A. Mancilla, "Affective States in Software Programming: Classification of individuals based on their Keystroke and Mouse Dynamics," *Research in Computing Science* 87, 2014, pp. 27-34.

[15] J. Hernandez, P. Paredes, A. Roseway, M. Czerwinsky, "Under pressure: sensing stress of computer users," in *Proc. 14th Conf. Human Factors in Computing Systems*, pp. 51-60, 2014, doi: 10.1145/2556288.2557165.

[16] A. Landowska, "Emotion monitor – concept, construction and lessons learned," in *Proc. Federated Conference on Computer Science and Information Systems*, pp. 75-80, 2015, doi: 10.15439/2015F384.

[17] A. Landowska, "Emotion monitoring - verification of physiological characteristics measurement procedures," *Metrology and measurement systems*, vol. 21(4), pp. 381-388, 2014, doi: 10.2478/mms-2014-0049.

[18] T. Mankiewicz, "Emotion recognition based on keystroke dynamics" (in Polish: "Rozpoznawanie emocji na podstawie dynamiki pisania na klawiaturze"), Master's thesis, Gdańsk University of Technology, 2014.

[19] Centre for Studies on Human Stress, "How to measure stress in humans," Fernand-Seguin Research Centre of Louis H. Lafontaine Hospital, Quebec, Canada, 2007.