Predicting Affective States of Programming using Keyboard Data and Mouse Behaviors*

Hualin Liu, Owen Noel Newton Fernando and Jagath C. Rajapakse, *Fellow IEEE*

***Abstract*— This study aims at predicting affective states during programming using keyboard and mouse data. The article proposes and evaluates a novel set of features under programming context to predict affective states. Fourteen undergraduate participants performed three programming tasks of varying difficulties. At the completion of each task, participants reported their affective states by viewing webcam videos and screen recordings. Features extracted from keyboard and mouse logs were used to train multiple classifiers. Among trained classifiers, feedforward neural network recognized positive, neutral and negative states with 52.9% accuracy. The overall Cohen's Kappa reached 0.27. Without neutral states, the classifiers were able to differentiate positive and negative states with 74.1% accuracy and 0.48 Kappa. Our approach demonstrates improved ability of predicting self-labelled affective states of programmers from keyboard and mouse data, without using specialized sensors, and potential of emotional feedback to programmers during learning to deliver better experience.**

I. Introduction

It is now a common practice for people to learn programming skills through various e-learning platforms. Typically, these platforms have videos and exercises for users to watch and practice. However, most e-learning platforms lack the awareness of users' affective states (emotion, stress, etc.) and do not respond to their emotions[1].

Many researchers have found that emotions are related to learning process and academic achievement [1, 2]. In terms of programming learning, a study shows that affect such as flow [3] is positively related to programming achievement and affects like boredom and confusion are negatively related to programming outcome [4]. Therefore, if emotions can be detected by the platform, it can give corresponding feedbacks to improve learning experience. Research has demonstrated the effectiveness of emotional feedback. Shen et al. proposed an affective e-learning model and built an experimental prototype that provided customized learning material based on emotion predicted. Their results showed a significant 91% performance increase with emotion-aware over non-emotion-aware platforms [5]. Other good examples of affect-sensitive intelligent tutoring systems include AutoTutor integrated with emotion sensors [6] and JavaTutor [7].

Others have previously used various methods to predict emotions of programmers. Muller et al. utilized biometric features such as electroencephalography (EEG) data, skin temperature, heart rate and eye tracking data [8]. Unfortunately, it is not practical to use biometric sensors in software platforms since they are mostly intrusive devices. Bosch et al. used Computer Expression Recognition Toolbox (CERT) to detect facial expressions from video and predict emotions [9]. Facial expressions are good indicators of emotions, video features however may be subject to environmental factors like lighting, video quality and recognizable faces. Drosos et al. extracted AST nodes, stylometric features and wordgram from code snippets to predict frustration [10]. While it yielded a high prediction accuracy, it cannot reflect emotion changes occurred during programming. For example, user can delete and rewrite part of the code which may change prediction completely. In this paper, we specifically analyze keyboard and mouse data and derive features that can predict self-labelled affective states from programmers. Choosing keyboard and mouse over other methods has several advantages. Firstly, keyboard and mouse are inexpensive and easily available. Secondly, keyboard and mouse are non-intrusive devices. In our experiment, participants labeled their affective states in each minute of mouse-key log by reviewing webcam videos and screen recordings. With collected data, we conducted feature engineering and evaluated feature set in predicting emotions with several classifiers.

II. Related Works

An experiment was conducted to discover emotions experienced during programming learning. In a computerized learning environment, Bosch et al. [11] studied what emotions novice programmers experienced during their first computer programming class. Researchers extracted key presses, "Run", "Stop", "Submit", "Show Hint" button presses, code snapshots and videos of participants' facial expressions during experiment and emotion labels were reported by participants who reviewed recorded videos. The results showed that five affective states: flow/engaged, neutral, confusion/uncertainty, boredom, frustration comprised 83% of the overall emotions experienced by novice programmers. Our study used these five affective states as labels.

A few literatures have discussed the detection of affective states during programming using keyboard and mouse data. Some studies investigated predicting emotion with mouse. Hernandez et al. [12] demonstrated a pressure-sensitive keyboard and capacitive mouse could be used to sense user's stress while performing designed tasks that required the use of

*Research supported by Nanyang Technological University.

Hualin Liu was with Nanyang Technological University, Singapore, Singapore (e-mail: liuhualin333@gmail.com).

Owen Newton Fernando, is with School of Computer Science and Engineering, Nanyang Technological University, Singapore (e-mail: ofernando@ntu.edu.sg).

Jagath C. Rajapakse is with the School of Computer Science and Engineering, Nanyang Technological University, Singapore (e-mail: asjagath@ntu.edu.sg).

[1] We use "Affective State" and "Emotion" interchangeably in this literature

keyboard/mouse under stressed/relaxed condition. Analysis showed that increased levels of stress greatly influenced typing pressure and amount of mouse contact. Most people tend to show forceful typing pressure and larger contact with mouse. Sun et al. [13] showed muscle stiffness in arm/hand, which are

stance) to infer negative emotions while performing a number ordering task and a task interacting with an e-commerce website. The negative emotions were induced by introducing unfair tasks on the computer. Experiments were to test the attentional control theory [15] that negative emotions decrease people's ability to control their attention. The result showed that negative emotion may affect mouse cursor distance and speed, resulting in greater cursor distance and slower average cursor speed. Some researchers demonstrated prediction using keystroke data. Epp et al. [16] extracted typing rhythms on the keyboard to detect users' emotional states when typing a free text paragraph. Keyboard features including keystroke duration features (dwell) such as duration of 1st key of all digraphs, keystroke latency features (flight) such as time elapsed from one key release to another key press, and features that combines dwell and flight features were extracted from the keyboard. There were also studies combining both mouse and keyboard data for prediction. Zimmermann et al. [17] used both mouse and keyboard features such as number of mouse clicks per minute, average duration of mouse clicks, total and average distance of mouse movements in pixels, keystroke per second, average duration of one keystroke, etc., to recognize induced emotions. Rodrigues et al. [18] carried out a study using both keyboard and mouse data to detect stress and used a feature set including click duration, click accuracy, amount of mouse movement, mouse clicks and keystrokes. It was found out that substantially greater usage of mouse and keyboard, high frequency of backspace key pressed, mouse clicks and scroll usage were indicators of stress.

Our work follows the recent study by Vea et al. [19] who used keyboard and mouse data to detect negative emotions of novice programming students learning C++. They used a combination of keyboard and mouse features and extracted features defined by a set of rules proposed in literature. The feature set was divided into three sets: (i) keystroke verbosity features such as keys pressed and typing speed; (ii) keystroke durations and latency features of digraph and trigraph; and (iii) mouse features derived from the number of left/right clicks. Extracted features were used to recognize three emotional labels namely frustration, confusion and boredom. Emotional labels were derived from viewing the webcam videos by researchers, rather than by self-judged reports by participants. In this study, we used self-judged approach to collect emotion labels since we believed participant reported labels were more realistic. We built on previously discovered features and introduced more programming-related features, which achieved higher accuracies in predicting self-labelled positive and negative emotions of programmers than other studies [8, 9].

proved to be effective in detecting stress, can be captured from common mouse using mass-spring-damper system parameters with the help of a linear predictive coding model. Hibbeln et al. [14] used mouse cursor movements (specifically mouse move speed and di

III. Methodology

This section explains three processes in our methods: data collection, feature extraction and selection, and predictive model building.

A. Data Collection

A mouse-keyboard logger was written in Python, inspired by Selfspy written by Gurgeh[2]. The logger ran in background and collected keyboard and mouse data while programming. The logger used PyHook[3] module to listen to low-level input device events such as key down, key up, and mouse move. It then stored collected data to SQLite[4] files. Logitech C922 Pro Stream Webcam was used to record facial video. A video and screen recorder were implemented using FFmpeg[5]. Data collection Graphic User Interface (GUI) was implemented using Python and contained instructions to guide participants through programming tasks. The execution of the programming task was synchronized with the mouse-keyboard logger, FFmpeg recording, and Sublime Text Integrated Development Environment (IDE).

In our experiment, we specifically made the following assumptions.

1. We assumed someone familiar with Python Programming once programmed in projects using Python before and is familiar with basic syntax.
2. We assumed year 3 and year 4 undergraduate students with programming experience had some knowledge about algorithm.

Fourteen (9 male and 5 female) year 3 and year 4 undergraduate students who claimed Python programming familiar to them participated in the experiment and performed three programming tasks of varying difficulties. At the start of the experiment, participants were given guidelines and reminded the definitions of emotion labels. The participants would carry out three programming tasks (at easy, medium, and hard difficulty levels) in an experimental session. These tasks were carefully chosen from Leetcode[6], an online programming learning website. We estimated and chose easy problems with finish time estimation of 15-20 mins, medium problems with 30 mins estimation and hard problems with more than 30 mins estimation to induce negative emotions. Participants needed to solve each algorithm problem within a 30-minute limit. To be more like real life scenario, participants could search for help in the internet but not for solutions. Web search was not separated from programming work in this study for following reasons: (i) programmers search in the internet all the time when learning and working; and (ii) activities such as web search may reflect emotion of a programmer. To record as much keyboard/mouse data as possible, pens and draft

[2] https://github.com/selfspy/selfspy
[3] https://sourceforge.net/p/pyhook/wiki
[4] https://www.sqlite.org/index.html
[5] https://www.ffmpeg.org/
[6] https://leetcode.com/

paper were not provided. The participants could choose to end one task in advance if he/she successfully finished the assignment.

Upon completion of each task, the participant was prompted with labeling GUI window. During labeling process, participants were shown with screen recording and video of facial expressions at 1.5x speed to avoid long labeling process. The videos were divided into clips corresponding to one minute of real time. As opposed to 15s used in previous studies [9, 19], we used 1 min time interval because emotions were observed not to change frequently and one-minute interval provided more data to train the classifiers. Participants were asked to report the most appropriate affective state (Flow/Engagement, Neutral, Boredom, Confusion, or Frustration) to each video clip.

B. Feature Extraction

The raw mouse-key log cannot be used for classification immediately. Some features need to be extracted. In our study, we carefully devised our feature set to be programming-related. Primary features were first constructed from raw mouse-key log. Table 1 lists seven primary features collected from mouse-key logs. We discarded the first- and last-minute data of each programming task to avoid initial and final settling effects. After this process, a mouse-key log of primary features was obtained.

Inspired by earlier studies and our own experience, we calculated the secondary features given in Table 2 from primary feature log. These secondary features are usually counts and average values in one minute. Each minute of primary feature log was therefore condensed into one data point. Emotion labels acquired in data collection process were assigned to corresponding data points.

Secondary features were divided into four categories: keystroke verbosity features, mouse features, keyboard/mouse usage features, and keystroke dynamics. Keystroke verbosity features measured number of keystrokes/keys hold/combination keys that have special functionality in the IDE or text editor for programming work. These features are closely related to programmers' behaviors. For example, we measured number of backspace and delete keys to monitor whether participants encounter mistakes in the code. Large values of backspace keys typed in one minute indicate the programmer is not in flow. Same for combination keys, we counted the number of undo, copy, save and build keystrokes. These combination keys are frequently used in programming. Take build keys for example, it is usually an indicator of confusion because there are usually bugs or syntax errors after building the code, which tend to make programmers confused. We also counted the keystroke that happens outside the IDE to explore if out-of-editor activities such as web search helped in emotion prediction.

Mouse behaviors captured all possible mouse actions, including mouse left/right clicks, mouse wheel action, mouse drag, number of mouse move, mouse move average speed. We design them because earlier study showed negative affects resulted in slower mouse speed and greater distance [14], and mouse click, scroll could be seen as indicators of stress [18], which may lead to other negative emotions.

TABLE I. PRIMARY FEATURES EXTRACTED FROM MOUSE-KEY LOGS

Feature	Description
Mouse Click	Left/Right/Middle mouse clicks, position, whether the event is in IDE
Mouse Scroll	Scroll up/down and position, whether the event is in IDE
Mouse Move	Mouse move with move length (Euclidean distance in pixels) and duration
Mouse Drag	Mouse drag behaviors
Mouse Idle	Record mouse idle time
Keystroke	Keycode/Combination, whether user holds the key, key hold time, whether the event is in IDE
Keyboard Idle	Record keyboard idle time

Inspired by work of stress detection from Roduigues et al. [18], we added keyboard/mouse usage features which were a set of time-related features including keyboard idle time, number of keyboard idle events, mouse idle time and number of mouse idle events.

Keystroke dynamics were comprised of key press and release time. Inspired by Epp et al. [16], we designed new keystroke dynamic features under programming context. It included the average dwell time between the first and second keystrokes in Python keywords, consecutive arrow keys, and consecutive backspace keys. Programming keyword, arrow keys and delete keys occur frequently in keyboard log of a code snippet just like digraph and trigraph in a free text. We would like to explore if the keystroke dynamics of them can be used to predict emotion as well. We performed a feature scaling on average time duration between two pressed keys to eliminate individual differences in typing speeds. We did not scale intervals between consecutive arrow keys and backspace keys because they were not seen as words.

To improve features for classification, we performed correlation analysis and recursive feature elimination. The correlated features were eliminated and features were ranked using recursive feature elimination (RFE). Least effective features were identified and removed in RFE using cross validation until an optimal set of features was obtained. The optimal feature set is then used for classification.

C. Classification

Since building individualized models require multiple experiments from the same subject, we build subject independent models to predict emotions from keyboard and mouse data. We tested with a few classifiers. Among them, SoftMax classifier, feedforward neural network with one hidden layer and SoftMax output layer, random forest, a gradient boosting decision tree called XGBoost[7] showed good performance.

We received only a few responses for boredom and frustration labels. This could be due to tasks being not bored and participants were not given huge pressure to finish the task in time. Because of the lack of data, we aggregated the labels into three affective states, namely positive, neutral and negative. The positive emotion includes engagement.

[7] http://xgboost.readthedocs.io/en/latest/

TABLE II. SECONDARY FEATURES EXTRACTED FROM PRIMARY MOUSE-KEY LOG

Feature type	Feature number	Feature code	Description
Content Features	1	Total_keys	No of keystrokes typed
	2	Hold_keys	No of keystrokes user held
	3	Backspace_keys	No of backspace keys typed
	4	Delete_keys	No of delete keys typed
	5	Undo_keys	No of "Ctrl+Z" combination typed
	6	Copy_keys	No of "Ctrl+C" combination typed
	7	Deletion_held_keys	No of "backspace, delete, Ctrl+Z" keys held
	8	Combo_keys	No of overall combination key
	9	Save_keys	No of "Ctrl+S" typed
	10	Alt_tab_keys	No of "Alt+Tab" typed
	11	Build_keys	No of "Ctrl+B" typed
	12	Arrow_keys	No of "Up, Down, Left, Right" typed
	13	Home_end keys	No of "Home and End" typed
	14	Not_ide_keys	No of keystrokes typed outside IDE
	28	Not_ide_clicks	No of clicks outside IDE
Keyboard Dynamic Features	15	Duration_keywords	Average duration between 1st and 2nd key down in a keyword in one minute
	16	Duration_arrow	Average duration between 1st and 2nd key down in consecutive arrow keys typed in one minute
	17	Duration_deletion	Average duration between 1st and 2nd key down in consecutive backspace keys typed in one minute
Mouse Behaviors	20	Mouse_move_avg_speed	Average speed of mouse movement
	21	Mouse_move_times	No of mouse moves
	22	Mouse_move_total_length	Total length of mouse move in one minute
	23	Mouse_move_total_time	Total time of mouse move in one minute
	24	Mouse_left_click	No of left clicks
	25	Mouse_right_click	No of right clicks
	26	Mouse_wheel_action	No of scroll events
	27	Mouse_drag	No of mouse drags
Keyboard/Mouse Usage Features	18	Keyboard_idle_time	Total keyboard idle time
	19	Keyboard_idle_events	No of keyboard idle events
	29	Mouse_idle_time	Total mouse idle time
	30	Mouse_idle_events	No of keyboard idle events

The negative emotions combined frustration, confusion and boredom states. All participants contribute data to positive and neutral states and over 78.5% of participants contribute to all affective states.

IV. RESULTS

Fourteen undergraduates with prior experience in Python programming participated in three programming tasks each lasting for 30 minutes. After each task, participants provided emotional labels for each minute of the task by going through facial videos and screen recording. We divided dataset into train set and test set with an 8:2 ratio. Classifiers' performances were evaluated.

A. *Feature Selection and Ranking*

Correlation map between features, visualized with Seaborn[8], is given in Figure 1. Numbers in the graph are feature numbers in Table 2. We identified two highly negatively correlated (smaller than -0.8) features (18: keyboard idle time and 29: mouse idle time), and two highly positive correlated (larger than 0.8) features (26: mouse wheel action and 28: not_ide_clicks). Deleting one of the highly correlated features improved the accuracy of the classifiers.

The RFE was applied to the feature set with all classifiers for ranking features and identifying an optimal set of features. Usually the dimension of the final set is around 20 features. Features were ranked based on the aggregate ranking score. The top 10 features are shown in Table 3. As seen, novel features such as arrow_keys (Keystroke Verbosity), duration_deletion and duration_keyword (Keystroke Dynamics) were ranked high and shown to be highly effective in predicting emotions from keyboard and mouse data.

B. *Classifier Performance*

We evaluated prediction accuracies with a set of classifiers. SoftMax classifier, feedforward neural network with one hidden layer, random forest and XGBoost had better performance and their performance were shown in this paper.

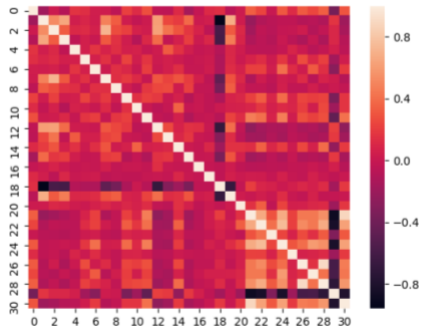

Figure 1. Correlation Heatmap Between Features

[8] https://seaborn.pydata.org

TABLE III. TOP 10 RANKED FEATURES

Feature Name	Ranking
Total_keys	1
Mouse_move_total_length	2
Keyboard_idle_events	3
Arrow_keys	4
Duration_deletion	5
Mouse_move_total_time	6
Mouse_move_times	7
Backspace_typed	7
Duration_keyword	9
Mouse_move_average_speed	10

Since the dataset was unbalanced, we used Synthetic Minority Over-sampling Technique (SMOTE) technique [20] which generates minority labels with replacement using K-Nearest Neighbor method, to rebalance the dataset and not to overfit classifiers due to the lack of data. To avoid possible overfitting, we used L2 regularization to penalize large weights in training neural network classifiers. For tree-based methods, the maximum depth and the number of estimators were specified.

Table 4 shows the performances of different classifiers in predicting positive, neutral and negative affective states. The best overall accuracy of 52.93% was achieved by feedforward neural network. Since classifiers were sensitive to initializations, performance measures were reported as averages of 100 prediction attempts. In each attempt, the training and test set were randomly subsampled. F1 score and Cohen's Kappa statistic [21] were also used to measure the performances due to the imbalance of class labels. According to Landis and Koch [22], the Kappa score value of 0.2788 indicates fair agreement between data points.

As seen from Table 4, the model differentiates positive and negative emotions well but struggles to predict neutral emotions. This may be because the neutral state does not carry any features that are drastically different from positive or negative emotions. For example, participants might have reported their affective states as neutral though they had mild positive or negative emotions. Another reason could be they marked idle periods with no keyboard/mouse inputs as neutral. To see the confounding effects of neutral state, we built a classifier to differentiate only the positive and negative affective states. The results listed in Table 5 shows an improvement of performance: 74.1% predicting positive and negative emotions with improved Kappa of 0.481.

V. DISCUSSION

The aim of our work is to explore useful features and methods that can detect self-labelled programmer's affective states with non-intrusive input devices such as keyboard and mouse. Compared to some of previous works, our set of features showed better performance in predicting positive and negative affective states. Bosch et al. [9] tracked facial features and achieved Cohen's Kappa of 0.22 and 0.23 for confusion and frustration and 0.04, 0.11 and 0.07, for boredom, flow/engagement and neutral states, respectively. Müller et al. [8] distinguished positive and negative emotions with an accuracy of 71.36% using features collected by biometric sensors. In contrast, our method using keyboard and mouse data distinguished the two labels with an accuracy of 74.11%.

There are several ways that our current method can be improved in the future. As for data collected, we had a limited dataset, so more data is needed. What's more, there were some time periods in which participants were thinking and not interacting input devices. These data points should be excluded from the analysis or we could add the "thinking" state in future study. Speaking of experiment design, we observed that given definitions of affective states, different volunteers may have different interpretations. Hence, in future work we will measure arousal and valence levels alongside the self-labelled affective states as the ground truth using biometric sensors. Besides, there were labels with very few responses. This could be due to the flaw in our experiment design or the chosen affective states cannot fully model the participants' feelings. In future study, we will design other emotion inducing techniques and apply unsupervised clustering to the ground truth data to obtain an accurate set of emotion labels under programming context. Regarding the current feature set, we didn't measure key-up time in mouse-keyboard logger. With key-up time measured, researchers can design more keystroke dynamics features in the future such as duration. In addition, although we didn't use distance-based in classifiers, we did apply KNN method in the dataset rebalancing step. As the size of dataset grows, it is necessary to apply other dimensionality reduction methodologies like Principle Component Analysis. and feedback to programmers

VI. CONCLUSION

Detecting emotions during programming is of vital importance for efforts on building next generation of e-learning platforms that can infer emotions from input devices

TABLE IV. PERFORMANCES OF DIFFERENT CLASSIFIERS PREDICTING POSITIVE, NEUTRAL AND NEGATIVE AFFECTIVE STATES

Classifier	Affect	F1 score	Precisions	Recall	Accuracies	Cohen's kappa
XGBoost	Positive	0.6158	0.5583	0.6898	0.5223	0.2623
	Neutral	0.1807	0.4078	0.1185		
	Negative	0.5806	0.5107	0.6766		
Random Forest	Positive	0.6126	0.5978	0.6316	0.5161	0.2617
	Neutral	0.2587	0.3550	0.2074		
	Negative	0.5685	0.5075	0.6492		
SoftMax Classifier	Positive	0.6030	0.5930	0.6157	0.5212	0.2701
	Neutral	0.2754	0.3621	0.2252		
	Negative	0.5851	0.5225	0.6677		
Feedforward Neural Network	Positive	0.6212	0.5931	0.6547	0.5293	0.2788
	Neutral	0.2434	0.3910	0.1790		
	Negative	0.5855	0.5119	0.6866		

TABLE V. PERFORMANCES OF DIFFERENT CLASSIFIERS PREDICTING POSITIVE AND NEGATIVE AFFECTIVE STATES

Classifier	Affect	F1 score	Precision	Recall	Accuracies	Cohen's kappa
XGBoost	Positive	0.7495	0.7548	0.7470	0.7274	0.4504
	Negative	0.6995	0.6991	0.7038		
Random Forest	Positive	0.7319	0.7680	0.7017	0.7199	0.4398
	Negative	0.7055	0.6750	0.7419		
SoftMax Classifier	Positive	0.7491	0.7886	0.7158	0.7384	0.4770
	Negative	0.7256	0.6919	0.7656		
Feedforward Neural Network	Positive	0.7552	0.7820	0.7322	0.7411	0.4810
	Negative	0.7244	0.7011	0.7519		

to improve their experience and efficiency.

The input devices for such systems cannot be intrusive and can be deployed without the knowledge of the programmers. We used common input devices like keyboard and mouse, and do not need other sophisticated input sensors.

We conducted experiments to gather mouse-key logs while programming and affective states were given by participants after watching screen recordings and facial videos. From keyboard and mouse data, we constructed features including keystroke verbosity features, mouse behaviors, keyboard/mouse usage features and keystroke dynamics. Our results showed that the programming-related feature set performs well in detecting self-labelled emotions of programmers. Feedforward neural network trained predicting three emotional labels: positive, neutral and negative achieved an overall accuracy of 52.9% and 0.27 Kappa and when trained to predict two emotional labels: positive and negative, classifier reached an overall accuracy of 74.1% and 0.48 Kappa.

The present study demonstrates that keyboard and mouse data can be effectively used to predict emotions of the programmers. Our study can be used to build next generation of affect-sensitive intelligent tutoring system under programming context. Researchers could begin with the set of present features and investigate higher order features to further improve prediction performance. One could also research on using dynamics of raw keyboard and mouse data to design the classifiers.

REFERENCES

[1] R. Pekrun, T. Goetz, A. C. Frenzel, P. Barchfeld and R. P. Perry, "Measuring emotions in students' learning and performance: The Achievement Emotions Questionnaire (AEQ)", *Contemporary Educational Psychology*, vol. 36, no. 1, pp. 36-48, 2011.

[2] R. M. Carini, G. D. Kuh and S. P. Klein, "Student Engagement and Student Learning: Testing the Linkages*", *Research in Higher Education*, vol. 47, no. 1, pp. 1-32, 2006.

[3] M. Csikszentmihalyi, *Flow: The psychology of optimal experience*. Harper and Row, New York, 1990.

[4] M. M. T. Rodrigo, R. S. Baker, M. C. Jadud, A. C. M. Amarra, T. Dy, M. B. V. Espejo-Lahoz, S. A. L. Lim, S. A. M. S. Pascua, J. O. Sugay and E. S. Tabanao, "Affective and behavioral predictors of novice programmer achievement", *Proceedings of the 14th annual ACM SIGCSE conference on Innovation and technology in computer science education - ITiCSE '09*, 2009.

[5] L. Shen, M. Wang, and R. Shen, "Affective E-Learning: Using "Emotional" Data to Improve Learning in Pervasive Learning Environment." *Journal of Educational Technology & Society*, vol. 12, no. 2, pp. 176-189, 2009.

[6] S. K. D'Mello, S. D. Craig, B. Gholson, S. Franklin, R. Picard and A. C. Graesser "Integrating Affect Sensors in an Intelligent Tutoring System.", *International Conference on Intelligent User Interfaces*, 2005.

[7] J. B. Wiggins, K. E. Boyer, A. Baikadi, A. Ezen-Can, J. F. Grafsgaard, E. Ha, J. C. Lester, C. M. Mitchell and E. N. Wiebe, "JavaTutor: An Intelligent Tutoring System that Adapts to Cognitive and Affective States during Computer Programming", *Proceedings of the 46th ACM Technical Symposium on Computer Science Education - SIGCSE '15*, 2015.

[8] S. C. Muller and T. Fritz, "Stuck and Frustrated or in Flow and Happy: Sensing Developers' Emotions and Progress", *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering*, 2015.

[9] N. Bosch, Y. Chen and S. K. D'Mello, "It's Written on Your Face: Detecting Affective States from Facial Expressions while Learning Computer Programming", *Intelligent Tutoring Systems*, pp. 39-44, 2014.

[10] I. Drosos, P. J. Guo and C. Parnin, "HappyFace: Identifying and predicting frustrating obstacles for learning programming at scale", *2017 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, 2017.

[11] N. Bosch, S. K. D'Mello and C. Mills, "What Emotions Do Novices Experience during Their First Computer Programming Learning Session?", *Lecture Notes in Computer Science*, pp. 11-20, 2013.

[12] J. Hernandez, P. Paredes, A. Roseway and M. Czerwinski, "Under pressure: Sensing Stress of Computer Users", *Proceedings of the 32nd annual ACM conference on Human factors in computing systems - CHI '14*, 2014.

[13] D. Sun, P. Paredes and J. Canny, "MouStress: Detecting Stress from Mouse Motion", *Proceedings of the 32nd annual ACM conference on Human factors in computing systems - CHI '14*, 2014.

[14] M. Hibbeln, J. L. Jenkins, C. Schneider, J. S. Valacich and M. Weinmann, "How Is Your User Feeling? Inferring Emotion Through Human-Computer interaction Devices", *MIS Quarterly*, vol. 41, no. 1, pp. 1-21, 2017.

[15] M. W. Eysenck, N. Derakshan, R. Santos and M. G. Calvo, "Anxiety and cognitive performance: Attentional control theory.", *Emotion*, vol. 7, no. 2, pp. 336-353, 2007.

[16] C. Epp, M. Lippold and R. L. Mandryk, "Identifying emotional states using keystroke dynamics", *Proceedings of the 2011 annual conference on Human factors in computing systems - CHI '11*, 2011.

[17] P. G. Zimmermann, S. Guttormsen, B. Danuser and P. Gomez, "Affective Computing—A Rationale for Measuring Mood With Mouse and Keyboard", *International Journal of Occupational Safety and Ergonomics*, vol. 9, no. 4, pp. 539-551, 2003.

[18] M. Rodrigues, S. Gonçalves, D. Carneiro, P. Novais and F. Fdez-Riverola, "Keystrokes and Clicks: Measuring Stress on E-learning Students", *Advances in Intelligent Systems and Computing*, pp. 119-126, 2013.

[19] L. A. Vea and M. M. T. Rodrigo, "Modeling Negative Affect Detector of Novice Programming Students Using Keyboard Dynamics and Mouse Behavior", *Lecture Notes in Computer Science*, pp. 127-138, 2017.

[20] N. V. Chawla, K. W. Bowyer, L. O. Hall and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique", *Journal of Artificial Intelligence Research*, vol. 16, no. 1, pp. 321-357, 2002.

[21] J. Cohen, "A Coefficient of Agreement for Nominal Scales", *Educational and Psychological Measurement*, vol. 20, no. 1, pp. 37-46, 1960.

[22] J. R. Landis and G. G. Koch, "The Measurement of Observer Agreement for Categorical Data", *Biometrics*, vol. 33, no. 1, p. 159, 1977.