Modeling Negative Affect Detector of Novice Programming Students Using Keyboard Dynamics and Mouse Behavior

Larry Vea[1,2](✉) and Ma. Mercedes Rodrigo[2]

[1] Mapua Institute of Technology, Makati City, Philippines
lavea@mapua.edu.ph
[2] Ateneo de Manila University, Quezon City, Philippines
mrodrigo@ateneo.edu

Abstract. We developed affective models for detecting negative affective states, particularly boredom, confusion, and frustration, among novice programming students learning C++, using keyboard dynamics and/or mouse behavior. The keystroke dynamics are already sufficient to model negative affect detector. However, adding mouse behavior, specifically the distance it travelled along the x-axis, slightly improved the model's performance. The idle time and typing error are the most notable features that predominantly influence the detection of negative affect. The idle time has the greatest influence in detecting high and fair boredom, while typing error comes before the idle time for low boredom. Conversely, typing error has the highest influence in detecting high and fair confusion, while idle time comes before typing error for low confusion. Though typing error is also the primary indicator of high and fair frustrations, other features are still needed before it is acknowledged as such. Lastly, there is a very slim chance to detect low frustration.

Keywords: Affect · Model · Novice programmer · Keyboard dynamics · Mouse behavior

1 Introduction

Affect is an observable expression of some emotional state [1–3]. It influences the ability of an individual to process information, to accurately understand and to absorb new knowledge [4].

In novice programmer studies, the negative affective states, particularly boredom and confusion, are negatively correlated with the student achievement while positive affect such as flow is positively correlated with achievement [5].

Affect detectors are built based on data acquired by sensors, human observations, or other peripherals. Several studies make use of keyboard dynamics as a source of affective data. These data include: typing speed, number of keystrokes, total time taken for typing, typing errors (the number of hits on the backspace key, delete key, or other unrelated keys), keyboard idleness [6, 7], keystroke latency time (dwell time) and keystroke duration time (flight time) between two-key (digraph or 2G) or three-key (trigraph or 3G)

M. Numao et al. (Eds.): PRICAI 2016 Workshops, LNAI 10004, pp. 127–138, 2017.
DOI: 10.1007/978-3-319-60675-0_11

combinations [8, 9]. These studies examined further how these keystroke data are related to a generally described as positive and negative affective states.

A few studies also examined how mouse movements are related to irritation, annoyance, reflectiveness [10], and boredom [11]. Other studies also make use of the combined keyboard and mouse data to examine how these are related to affective states in terms of valence and arousal [7].

There are only a few studies that detect the affective states of novice programmers [e.g. 12, 13]. Also, there is no literature yet that uses the combined keyboard and mouse data to detect such states. This study hopes to contribute to the literature by building and validating a detector for negative affect of novice programming students using both the keyboard and the mouse data. We also attempt to answer the following research questions: (1) what are the notable features from keyboard dynamics and/or mouse behavior that help out in the recognition of negative affective states of novice programming students? (2) how is student's affect related to keyboard dynamics and/or mouse behavior; (3) are the notable features "stable" or "consistent" over student's programming time period? (4) how do these features differ or similar among high/medium/low incidences of boredom, confusion, and frustration? and (5) what is the effect of combining mouse behavior with the keystroke dynamic features in predicting student's affect compared when using keystroke features alone or mouse features alone?

This study hopes to contribute to the development of formal models of recognizing affective states of novice programmers, using the most common, low cost, non-intrusive computer devices such as the keyboard and the mouse. The discovered models or patterns to recognize negative affective states in this study may be used by computer scientists in developing computational systems that may automatically provide feedback to both teachers and students.

2 Related Works

Though there are different devices for affective states detection when using a computer, the keyboard and the mouse are the most commonly available, low-cost, and non-intrusive devices that could obtain affect indicators.

There were several studies that use only the keyboard as data source for affect detection. For example, Khanna et al. [6] extracted keystroke features: typing speed, four statistics (mode, standard deviation, variance and range) from the number of typed characters for a defined time interval, total time taken for typing, number of backspace hits and idle times from recorded key logs to detect positive, negative, and neutral state of a computer user. These keystroke data were gathered from participants who were asked to retype some fixed texts in different time in order to acquire keystroke information under different affect states. The corresponding affect is collected by asking the participants to describe and report their affective state while doing the task. The resulting dataset was then analyzed through some data mining algorithms such as SMO, MLP, and J48, They found out that the increase in the user typing speed relative to neutral state is an indicator of positive affect state while the decrease in the typing speed relative to neutral state is an indicator of negative affect.

An attempt to detect confusion and boredom states of novice programming students, Felipe et al. [12] extracted the same keystroke features used by Khanna et al. [6]. They also wanted to determine which of the extracted features could be indicators of the said affective states. The authors were permitted to collect video and key logs from students having programming activities. They reviewed every 20-second segment of the collected video logs and observe the student's behavior. They label affect by matching the corresponding observations from a checklist that describes affective states in terms of student's behavior. Results show that in a 20-second interval, keyboard inactivity in that time interval is the indicator of boredom state while confusion state was observed when the number of backspaces is greater than the idle time.

Tsui et al. [9] also used key duration time (key press to key release) and key latency time (from one key release event to the next key press) features to examine the difference between positive and negative affect states. The keystroke data were collected by asking each participant to type a fixed number sequence with a pen on the mouth. The affect is labeled based on the teeth condition (positive) and the lip condition (negative) of the participant while typing. They found out that the duration time significantly show the difference between the two opposite states.

The features used by Bixler and D'mello [16] to discriminate between natural occurrences of boredom, engagement, and neutral states are divided into four keystroke and timing features: relative timing (session and essay timings), keystroke verbosity (number of keys and backspaces), keystroke timing (latency measures) and pausing behaviors. These features were extracted from the key logs of participants who were asked to write an essay about some selected topics using a computer. Likewise, the affect was labeled by asking the participant to view every 15-second segment of his video log and has to make self-judgment on what affective state was present in him during each time segment. Results show that when the identified keystroke and timing features were combined with task appraisal and stable traits features, it yields to a higher accuracy rate in classifying emotions, specifically, between boredom and engagement.

There were also studies that explored mouse as data sources in affect detection. For example, Tsoulouhas et al. [11] extracted seven mouse movement features to detect emotional state, specifically boredom, of students who attend a lesson online. The said features are: total average movement speed, latest average movement speed, mouse inactivity occurrences, average duration of mouse inactivity, horizontal movements to total movements ratio, vertical movements to total movements ratio, diagonal movements to total movements' ratio, and the average movement speed per movement direction. They found out that the primary indicators of boredom are the average movement speed per movement direction and the mouse inactivity occurrences.

A more comprehensive study on affect detection in terms of its two dimensions was presented by Salmeron-Majadas et al. [7]. They evaluated the keyboard and mouse affective data to identify participant's affective states in terms of valence and arousal. They combined some previously presented keyboard indicators such as the keystroke indicators used by Khanna [6] and Bixler and D'Mello [16], and the digraph and trigraph used by Epp et al. [8]. Their mouse indicators were generated from the participant's mouse clicks, cursor movements and scroll movements. These include: the number of button presses (left, right and both), overall distance, distance the cursor has been moved (covered distance) between two button press events, between a button press and the following button release event, between two button release events and between a button release and the following button press events, the Euclidean distance in the previous described cases, the difference between the covered and the Euclidean distance between the events described before, and the time elapsed between the mentioned events. After the participants finished the given task, they were asked to evaluate and score their affective state using the SAM scale. They computed the correlation between the extracted mouse/keyboard indicators and the reported affective states and found out that the mouse indicators that are correlated to the valence dimension of affect are: the mean time between two consecutive mouse button press events; the mean time between two consecutive mouse button release events; the standard deviation of the difference between the covered and the Euclidean distance between two consecutive mouse button press events; the standard deviation of the difference between the covered and the Euclidean distance between a mouse button release and the following mouse button press events; and the mean time between a mouse button release and the following mouse press button event; while the keyboard indicators are: the standard deviation of the time between two key press events; the mean duration of the digraph; the mean duration between the first key up and the next key down of the digraph; the duration between two key press events when grouped in digraphs; and the mean time between two key press events. On the other hand, the mouse indicators that identify the arousal dimension of affect are: the mean of the difference between the covered and the Euclidean distance between a mouse button release and the following mouse button press events; the mean of the difference between the covered; and the Euclidean distance between two consecutive mouse button press events; while the keyboard indicators are: number of keys pressed; the numbers of alphabetical characters pressed; the mean of the duration of the second key of the digraphs; the duration of the third key of the trigraphs; and the standard deviation of the duration of the digraph. Finally, they used these mouse and/or keyboard indicators in training some classifiers in order for them to know the prediction rates in recognizing positive and negative valence dimension of the participants. Results show that for some well-known classifiers such as C4.5 and Naïve Bayes, keyboard indicators alone provided the higher prediction rates than the mouse data alone, and even the combination of the data sources. However, for some more complex classifiers such as Random Forest and AdaBoost, the combined mouse and keyboard indicators provided the highest prediction rates among all the results.

Though there are some few studies on the affective states of novice programmers (e.g. [3, 5, 12, 13]), to date, there is no literature yet that uses the combined keyboard and mouse data to detect some negative affective states of these novices.

3 Methodology

3.1 Participants

The participants in this study were 55 volunteers from first year students of a higher educational institution in Makati City. All of them were given waivers to parents or guardians, asking permission to let their child participate in the study. Hence, only those students with parent's/guardian's consent were allowed to participate.

At the time of the study, the students were enrolled in CS126 - Programming 1 with no or minimal background in C++. CS126 is a first year introduction to programming course using structured programming approach. Topics include: simple C++ syntax; program flow description; variables and data types; C++ operators; C++ control structures such as sequential, selection, and iterative structures; and functions.

3.2 Data Collection Methods and Instruments

With the consent of the school, we used a customized mouse-key logger, web cam, the MS Movie Maker, and the Dev-C++ Integrated Development Environment.

Before the student works on its programming activity, the web cam is already properly in place and turned-on. The mouse-key logger and the Movie Maker were set and running in the background and hidden from the student in order not to bother him/her while he/she is doing the programming activity.

The mouse-key logger captured the mouse motion, mouse clicks, and mouse scrolls and the key event logs while the web cam captured the facial expressions and body movements of the student (video logs). The Dev-C++ was used as the programming environment in doing the programming activities.

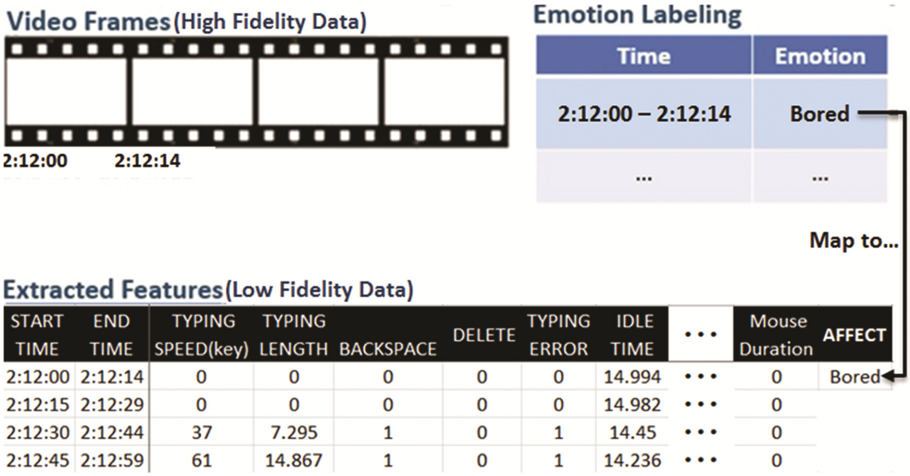
Data was collected from the participants where the problem is about selection constructs and loop constructs, respectively. Data recording took almost 3 h.

3.3 Data Processing

We mapped the mouse-key logs with the video logs in several steps: We first cleaned the data by removing segments in the mouse-key logs that had no corresponding video logs; then we extracted potential keystroke and mouse dynamic features identified in some previous works, plus other features that may influence affect detection, from the mouse-key logs. The result was a comma separated value (csv) file containing the keyboard and mouse dynamic features at every 15-second interval. This file was called the "incomplete dataset" since the affect labels were not yet attached. We also divided the video logs into 15-second video time segments that corresponded to mouse-key time segments in the incomplete dataset. Then, affect labeling on each video segment was done by three trained labelers, one was a graduate student serving as lead and the other two were college seniors with strong background in computer programming. They watched the video together and came to a consensus regarding the student's affective state based on the coding scheme in Table 1. If there were disagreements, they played the segment until they agreed. Video segments where the participants showed curiosities about being monitored through the camera or not seen in the video were marked "X". Finally, we mapped each label of the video segment in the incomplete dataset (Fig. 1), and the instances labeled with "X" were deleted.

Table 1. Affective state criteria

Affective states	Description
Boredom	Slouching and resting the chin in his/her palm; Yawning; Zoned out within the software; Looks uninterested/unfocused; Barely uses the mouse/keyboard; Slouching; Eyes wandering
Confusion	Scratching his/her head; Repeatedly looking at the same interface elements consulting with a classmate or a teacher; Flipping through lecture slides or note; Statements such as "Why didn't it work?"; Still engage with the software; Cannot grasp/experiencing difficulty with the material; On-task conversation; Pouts/Frowns/wrinkles brows/forehead; Nail biting; Lip biting; Lip slightly ajar
Frustration	Banging on the keyboard or pulling at his/her hair; cursing; statements such as "What's going on?!"; Scratching the back of his head; Rubbing his neck from behind; Scratching any part from his body; Changing his sitting position; Lips pulled inward; Raising the arms lifts sometimes up (or two arms- like throwing something in the air); Deep breath


Video Frames (High Fidelity Data)
2:12:00
2:12:14
Emotion Labeling
Map to...
Extracted Features (Low Fidelity Data)

Time	Emotion
2:12:00 – 2:12:14	Bored
...	...

START TIME	END TIME	TYPING SPEED(key)	TYPING LENGTH	BACKSPACE	DELETE	TYPING ERROR	IDLE TIME	• • •	Mouse Duration	AFFECT
2:12:00	2:12:14	0	0	0	0	0	14.994	• • •	0	Bored
2:12:15	2:12:29	0	0	0	0	0	14.982	• • •	0	
2:12:30	2:12:44	37	7.295	1	0	1	14.45	• • •	0	
2:12:45	2:12:59	61	14.867	1	0	1	14.236	• • •	0	

Fig. 1. Mapping of high fidelity data with the low fidelity data.

Determining of student's affect from the video segment was based on the modified coding scheme adopted from [3, 5, 14] and is presented in Table 1. The scheme was modified to find the state of confusion (negative valence, positive arousal), boredom (negative valence, negative arousal), frustrated, and a special affective state labeled as "others" [3, 6] in which the emotion with respect to the time frame was found to be neither confused, bored, nor frustrated.

The resulting complete dataset was then further divided into training and test set. Every fifth participant from the list was chosen as part of the test set while the rest were part of the training set.

3.4 Model Development and Data Analysis Methods

We used these datasets to develop several affective models for detecting confusion, frustration and boredom by training some well-known tree classifiers that could handle datasets with nominal class such J48, Decision Tree, and Random Forest using RapidMiner. Each classifier were trained and validated using different feature set, such as: keystroke verbosity features alone (KV); keystroke time duration and latency features of the digraph and trigraph alone (KT); all keystroke features - the combined verbosity, time duration and latency features (KF); mouse features alone (MF); and, the combined all keystroke and mouse features (KM). The gini index attribute criterion was used for feature selection and batch-X-validation to validate the model. The depth of the tree in each tree classifier was also explored in order to determine the model that has the highest performance in terms of accuracy rate and/or kappa statistic.

It was observed during the experiment that using keystroke time duration and latency features on the digraph and trigraph alone (KT), as well as mouse features alone (MF) do not provide a good model to detect negative affect since the kappa statistic is very low (less than 0.2) which implies a slight agreement [15]. It was also observed that the decision tree classifier consistently provide the highest kappa statistic and accuracy rate. It also implies that decision tree classifier gave the most acceptable model.

Lastly, the kappa and accuracy of the other feature sets are statistically tied (Table 2, columns 2 and 3). And since the kappa is in moderate agreement [15], it implies that these feature sets can be used to model negative affect detector. The models generated by the decision tree classifier for the said feature sets were tested using a pre-labeled test set for further investigation. The result of the tests is also presented in Table 2, columns 4 and 5. The table shows that the kappa and the accuracy significantly increased but are still statistically tied. This confirms that the three (3) feature sets can be used to model negative affect detectors of novice programming students.

Table 2. Model performance using decision tree classifier

Feature set	Training phase		Testing phase	
	Kappa	Accuracy (%)	Kappa	Accuracy (%)
KV	0.493	70.80	0.564	74.08
KF	0.489	71.03	0.568	74.28
KM	0.490	71.06	0.567	74.23

The tree models were further analyzed to find the significant features that help out in the recognition of negative affective states of novice programming students and how these features are related to student's affect. This was done by listing the unique inner nodes of the decision tree models generated by the classifier.

Using correlations in RapidMiner, it was observed that some of the notable features in the tree are strongly correlated. For example: typing error is highly correlated with backspace; total keyevents is also highly correlated with typing speeds and total time for typing; the sum of all time durations the student acted on the 1st key of the digraph (SUM_2G_1Dur) is fairly correlated with the maximum value in the set of the durations of the 1st key of the trigraph (MAX_3G_1Dur); and the total distance travelled by the mouse along the x-axis (MM_Total_X) is highly correlated with mouse activity duration. Thus, to achieve a more parsimonious model, we tried iteratively removing some features that are highly correlated to other features. Results show that the kappa and accuracy slightly improved (see Table 3). The table shows that kappa and accuracy in all the feature sets are almost equal. It implies that the notable features from the keystroke verbosity feature set alone (KV) or the combined verbosity, duration, and latency keystroke features (KF) are already enough to model a negative affect detector of novice C++ programming students. However, adding MM_Total_X (total distance travelled by the mouse along the x-axis) mouse feature with the keyboard features (KF) slightly improved the recognition rate of the model (Table 3).

Table 3. Model performance when some features correlated to other features were removed.

Feature set	Kappa		Accuracy (%)		Remaining significant features
	Before removal	After removal	Before removal	After removal	
KV	0.564	0.569	74.08	74.23	Typing error, typing variance, idle time, total key events, F9
KF	0.568	0.568	74.28	74.28	Typing error, typing variance, idle time, total key events, F9, MAX_3G_1Dur, AVE_3G_2D3D, and SUM_2G_1Dur
KM	0.567	**0.572**	74.23	**74.37**	Typing error, typing variance, idle time, total key events, F9, SUM_2G_1Dur, AVE_3G_2D3D, and MM_Total_X

To specifically determine how student's affect related to keyboard and mouse dynamics, the unique paths from the root of the decision tree of the KM feature set, to the its leaves were analyzed and then transformed into rules. The result is shown in Table 4.

We examined if the features were stable over time since it is possible that student keyboard and mouse dynamics change as the student develops and completes a program. Also, a student may type more in the beginning of the development process, when he is still writing code, and less so when he is debugging. We therefore divided the dataset into the first 1/3, the second 1/3, and the last 1/3 of the observation period and re-processed each subsets.

Results show that when students are just starting with their programming activity (first 1/3 of the period), the most notable features that determines student‘s negative affect in all feature sets are the typing error, and idle time. Though typing error and the idle time are also the dominant features on the second 1/3 of the period, other keystroke verbosity features such as typing variance, total keyevents, and the number of times the student presses F9 (shortcut to compile and run the program) are included. Also, adding the average duration time of the first key in the trigraph (AVE_3G_1Dur) or the total distance travelled by the mouse along the x-axis (MM_Total_X) improves the recognition rate. Lastly, at the time the programming period is almost toward its end (last 1/3 of the period), the typing error and idle time are still the dominant features, but adding the typing variance and total mouse movement along the x-axis increases student's negative affect detection. It was also observed that the total keyevents and the average duration time of the first key in the trigraph (AVE_3G_1Dur) that represent the

Table 4. How student affect related to keystroke dynamics and mouse behaviors.

Affect (most likely)	Pattern (based on a 15-second observation)
Boredom	IF ((Typine error ≤ 0.50) and (Idle time > 14.98) and (Total keyevents ≤ 0.50) and (MM_Total_X ≤ 243.50))
Frustration	IF ((Typing error ≤ 0.50) and (Idle time > 14.98) and (Total Keyevents > 4.50))

Confusion	IF (Typing error > 3.50); OR
	IF ((1.50 < Typing error ≤ 3.50) and (Typing variance > 0.815) (MM_Total_X > 17383.50)); OR
	IF ((Typing variance > 0.815) and (Typing error ≤ 1.50) and (SUM_2G_1Dur ≤ 0.83)); OR
	IF((1.5 ≤ Typing error ≤ 3.50) and (Typing variance ≤ 0.815)); OR
	IF ((Typing variance ≤ 0.815) and (Typing error ≤ 1.50) and (AVE_3G_2D3D ≤ 5.170)); OR
	IF ((Typing error ≤ 0.50) and (Idle time > 14.98) and (Total Keyevents ≤ 0.50) and (MM_Total_X > 850)); OR
	IF ((Typing error ≤ 0.50) and (Idle time ≤ 14.98) and (Total Keyevents > 9.50) and (SUM_2G_1Dur > 0.924) and (F9 > 27)); OR
	IF ((Typing error ≤ 0.50) and (Idle time ≤ 13.636) and (Total Keyevents ≤ 9.50) and (MM_Total_X > 872956.5))

Table 5. Differences or similarities among high/medium/low incidences of boredom, confusion, and frustration.

Affect		Keystroke dynamic features	Combined mouse and keystroke dynamics features
Boredom	High	Idle time, Total keyevents	Idle time, mouse activity duration
	Fair	Idle time, Total keyevents	Idle time, total keyevents
	Low	Typing error, Idle time	Typing error, idle time, numbers, mouse activity duration
Confusion	High	Typing error	Typing error
	Fair	Typing error, Typing speed(char)	Typing error, typing speed(char)
	Low	Idle time, Typing error	Idle time, typing error
Frustration	High	Typing error, Typing variance, Idle time, Control	Typing error, typing variance, idle time, control
	Fair	Typing error, Idle time, Keypress_DeltaX, F9, etc.	Typing error, idle time, CMM_time, keypress, etc.
	Low	(no sign of low frustration)	(the tree is very deep where there is a very minimal sign of frustration)

movements of the keys, including F9 which represents running the program were gone towards the end of the programming period. This may indicate that there were only few monitored keyboard activities. Probably, some of the students may have stopped working; either they are already finished with the activity or they have abandoned their work.

Finally, to determine how the notable features differ or similar among high/medium/low incidences of boredom, confusion, and frustration, the original dataset was divided into other subsets by computing the percentage of the time each student was observed to be bored, confused or frustrated and then segregate the data into the top 1/3 of those who are bored, confused or frustrated, the middle 1/3, and the lowest 1/3, and then re-process each subsets. The result is shown in Table 5.

4 Conclusion

This study was conducted to address the following research questions: (1) what are the notable features from keyboard dynamics and/or mouse behaviors that help out in the recognition of negative affective states of novice programming students? (2) how is student's affect related to keyboard dynamics and/or mouse behaviors; (3) are the notable features "stable" or "consistent" over student's programming time period? (4) how do these features differ or similar among high/medium/low incidences of boredom, confusion, and frustration? and (5) what is the effect of combining mouse features with the keystroke features in predicting student's affect compared when using keystroke dynamic features alone or mouse behaviors alone? These questions are answered as follows:

(1) The notable features from keyboard dynamics and/or mouse behaviors that help out in the recognition of negative affective states of novice programming students are presented in Table 3. These include: the student's typing errors incurred (the number times the backspace and delete keys were pressed); the length of time the student is idle (not pressing any key in the keyboard); the student's typing variance (his/her typing varies with time); the number of key events (keydown + keypress + keyup) he/she executed in the keyboard; total distance the student moved the mouse along the x-axis (MM_Total_X); the sum of all time durations the student acted on the 1st key of the digraph (SUM_2G_1Dur); the average time duration between the 2nd and 3rd keydown of the trigraph (AVE_3G_2D3D); and, the number of times F9 key (shortcut to compile and run the program) was pressed.

(2) As shown in Table 4, student's boredom is related to both keystroke dynamics and mouse behavior. The keyboard has almost no activity while the mouse has a very minimal movement along the x-axis. On the other hand, student's frustration is similar to boredom, except for the mouse features, since for this affect, students tend to release the mouse and scratch their head or do some other hand gestures. There is almost no keyboard activity too since when a student get frustrated, he/she usually pause for a while and do nothing. Lastly, student's confusion is both related

to keystroke dynamics and mouse behavior. The table shows that there are several indicators when a student is confused.

(3) After analyzing the data at first 1/3, the second 1/3, and the last 1/3 of the observation period, it was observed that the features are not stable since there are many features needed in detecting negative affect at the middle (second 1/3) of the observation period. It was also observed that the typing error has the greatest influence during the first 1/3 and second 1/3 of the observation period followed by the idle time, while the idle time has the greatest influence during the last 1/3 followed by the typing error.

(4) Table 5 shows that idle time has the greatest influence in detecting high and fair boredom but it is just secondary with the typing error for low boredom. On the other hand, typing error has the greatest influence in detecting high and fair confusion but it is just secondary with the idle time for low confusion. Though typing error is also the primary indicator of high and fair frustrations, it requires other features before it is acknowledged as such.

As shown in the last row of Table 3, adding a mouse feature, particularly with the distance it travelled along the x-axis, with the keystroke features improve the detection of student's affect compared when using keystroke dynamic features alone or mouse behaviors alone.

References

1. Affect. Encyclopedia of Mental Disorders. http://www.minddisorders.com/A-Br/Affect.html
2. Psychiatry Clerkship. http://depts.washington.edu/psyclerk/glossary.html
3. Carlos, C.M., Delos Santos, J.E., Fournier, G., Vea, L.: Towards the development of an intelligent agent for novice programmers through face expression recognition. In: Proceedings of the 13th Philippine Computing Science Congress, pp. 101–106 (2013)
4. Picard R.W., The Medial Lab – Affective Computing Group: Affective computing. http://affect.media.mit.edu/
5. Rodrigo, M.M.T., Baker, R.S., Jadud, M.C., Amarra, A.C.M., Dy, T., Espejo-Lahoz, M.B.V., Lim, S.A., Pascua, S.A., Sugay, J.O., Tabanao, E.S.: Affective and behavioral predictors of novice programmer achievement. In: Proceedings of the 14th Annual ACM SIGCSE Conference on Innovation and Technology in Computer Science Education (ITiCSE 2009), vol. 41, no. 3, pp. 156–160 (2009). http://doi.acm.org/10.1145/1562877.1562929
6. Khanna, P., Sasikumar, M.: Recognising emotions from keyboard stroke pattern. Int. J. Comput. Appl. **11**(9), 1–5 (2010)
7. Salmeron-Majadas, S., Santos, O.C., Boticario, J.G.: Exploring indicators from keyboard and mouse interactions to predict the user affective state. In: Educational Data Mining (2014)
8. Epp, C., Lippold, M., Mandryk, R.L.: Identifying emotional states using keystroke dynamics. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 715–724 (2011)
9. Tsui, W.H., Lee, P., Hsiao, T.C.: The effect of emotion on keystroke: an experimental study using facial feedback hypothesis. In: 2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), pp. 2870–2873 (2013)

10. Schuller, B., Rigoll, G., Lang, M.: Emotion recognition in the manual interaction with graphical user interfaces. In: Proceedings of the 2004 IEEE International Conference on Multimedia and Expo (ICME 2004), vol. 2, pp. 1215–1218 (2004). http://dx.doi.org/10.1109/ICME.2004.1394440
11. Tsoulouhas, G., Georgiou, D., Karakos, A.: Detection of learner's affective state based on mouse movements. J. Comput. **3**(11), 9–18 (2011)
12. Felipe, D.A.M., Gutierrez, K.I.N., Quiros, E.C.M., Vea, L.A.: Towards the development of intelligent agent for novice C/C++ programmers through affective analysis of event logs. Proc. Int. MultiConf. Eng. Comput. Sci. **1**, 511–518 (2012)
13. Lee, D.: Detecting confusion among novice programmers using BlueJ compile logs. Master's thesis. Ateneo de Manila University, Quezon City (2011)
14. Dragon, T., Arroyo, I., Woolf, B.P., Burleson, W., el Kaliouby, R., Eydgahi, H.: Viewing student affect and learning through classroom observation and physical sensors. In: Woolf, B.P., Aïmeur, E., Nkambou, R., Lajoie, S. (eds.) ITS 2008. LNCS, vol. 5091, pp. 29–39. Springer, Heidelberg (2008). doi:10.1007/978-3-540-69132-7_8
15. Viera, A.J., Garrett, J.M.: Understanding interobserver agreement: the kappa statistic. Fam. Med. **37**(5), 360–363 (2005)
16. Bixler, R., D'Mello, S.: Detecting boredom and engagement during writing with keystroke analysis, task appraisals, and stable traits. In: Proceedings of the International Conference on Intelligent User Interfaces, pp. 225–234 (2013)