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

Larry A. Vea
Mapua Institute of Technology
333 Sen. Gil Puyat Avenue
Makati City

lavea@mapua.edu.ph

Ma. Mercedes T. Rodrigo
Ateneo de Manila University
Katipunan Avenue, Loyola Heights
Quezon City

mrodrigo@ateneo.edu

## ABSTRACT
Learning to program is vital to novice programming students. During their learning process, particularly when they are making a program, affect plays a significant role. Affect may either motivate them to logically think and effectively respond to the programming activities or, it may make them to disengage or even withdraw from the programming task. Negative affect detection in the context of novice education can cue an intervention. When negative affect is detected, it opens an opportunity for either the teacher or an automated system to change the novice's disposition. Hence, this study aims to develop affective models for detecting negative affective states, particularly boredom, confusion, and frustration, of novice programming students through keyboard dynamics and mouse behavior. It attempts to discover patterns that reflect the relationship of student affect with keystrokes and/or mouse features. The features were extracted from a customized mouse-key logs gathered from 55 novice C++ students and were labeled with the affective state observed from the corresponding video logs, which were gathered simultaneously with the mouse-key logs. Features that are highly correlated to affect detection were selected through a data mining tool and these were used to train well known classifiers. The results were analyzed in terms of some measures such as accuracy rate and kappa statistic to determine the acceptable models and to identify notable patterns that reflect the recognition of negative affect in terms of the selected features. Lastly, the models were tested using a pre-labeled test set.

## CCS Concepts
• Human-centered computing→ Human computer interaction (HCI) → HCI design and evaluation methods→ User models

## Keywords
Affect; novice programmer; keyboard dynamics; mouse behavior; digraph; trigraph.

## 1. INTRODUCTION
Affect is an observable expression of a state of feeling [1][2][3]. It is the outward appearance of some emotional state [4].

Affect or emotion influences individual cognition, perception, and everyday tasks such as communication, learning, and even rational decision-making [5]. It influences the ability of an individual to process information, to accurately understand and to absorb new knowledge [6]. In other words, affect play an important role in learning [7][8]. This is also true in the field of computer programming. Studies show that negative affect is correlated to learning performance of programmers, especially among novices [9][10].

To extend studies in recognizing learner's affect, this study aims to develop affective models for detecting negative affective states of novice programming students through keyboard dynamics and mouse behavior. Specifically, this study's research objectives are: (1) to define notable features from keyboard dynamics and/or mouse behaviors that influence the detection of novice programmer's negative affect; (2) to discover significant patterns that reflect the relationship of student affect with keystroke dynamics and/or mouse behavior; and (3) to validate the affective models in order to identify which of these models provide the most acceptable result.

This study also tries 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 affect states of novice programming students? (2) how is student affect related to keyboard dynamics? (3) how is student affect related to mouse behaviors? (4) will the combined features from keystrokes and mouse movements provide better predicting model than using keystroke features alone, or mouse behaviors alone? (5) to what extent do the models correctly predict novice programmer's affect?

Since the keyboard and the mouse are the most commonly used input devices in computer programming, this study focuses in developing affective model that can detect negative affect states of novice C++ programming students through keyboard dynamics and mouse behaviors. The model centers in the detection of negative affect that may cause students to disengage from the activities. Such affect are boredom, confusion, and frustration [9][11]. These affect, particularly boredom and confusion, are not only be possible causes of students to stop working on their program but also found to be negatively related to their achievement in the class [9][10]. On the other hand, though frustration was not found to be a predictor of student's achievement [9][10], this affect is still a concern since this may cause a student to disengage [12] or ultimately give up [13] from the programming task.

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 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. Tsihrintzis et al [14] uses keyboard-stroke information to complement their visual-facial emotion recognition method. They examined the user typing speed (normal, below normal, above normal typing speed), the number of times the backspace is used, the number of unrelated keys hit, and keyboard idleness as parameters to recognize the six basic emotions, namely surprise, anger, happiness, sadness, disgust, as well as the neutral states. The affect was labeled based on the user's self-assessed questionnaire and was found out that the aforementioned keyboard parameters are indicators of happiness, anger and sadness but not for surprised and disgust.

Khanna et al [15] 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 [16] extracted the same keystroke features used by Khanna et al [15]. 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 less than the idle time.

Another study tried to determine what emotions do novice programmers experience during their first computer programming learning session was conducted by Bosch et al [9]. The participants (29 novice programming students) were tested in a computerized learning environment, then the authors recorded the participant's key presses, the Run, Stop, Submit, Show Hint button presses, the code snapshots and the video of the participant's faces during the learning session.. The affective data were labeled through the participant's retrospective affect self-judgment after viewing his/her videos and the corresponding computer screens while doing the activities. Results shows that flow/engaged, confusion, frustration and boredom are the most commonly present affective states in novices during programming activities.

Instead of using the keystroke features indicated above, Epp et al [17] used another approach. They collected keyboard and affective data by prompting selected participants to type a randomly selected fixed text and report his current affective state through a questionnaire that contains 15 5-point Likert scale questions. The authors computed the keystroke latency features (dwell time) and the keystroke duration features (flight time) between two-key combination (digraph) and three-key combination (trigraph), and used these features to identify the state of the user from a long list affective states. The top affective states that these features could determine are: confidence, hesitance, nervousness, relaxation, sadness, and tired.

Tsui et al [18] 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.

Solanki and Shukla [19][20] used combination of key occurrences (number of characters in the sample, number of mistakes found (backspace + delete keys), number of digits found, number of symbols found, number of letters found, and the total number of key pressed); the ASCII code that represents each key; the dwell time (minimum, maximum, mean, mode, medium, standard deviation, and variance); and the flight time to identify user's affect state such as confidence, sadness, happiness, tiredness, nervousness, anger, and others. Unfortunately, the authors did not properly present and discuss the results of their study.

The features used by Bixler and D'mello [21] 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. Schuller et al [22] used some geometrical (total sum of the contour values, number of zero-crossings, maxima, minima, means of the absolute values, standard deviations, and the variances) and temporal (auto-correlation function of the contour, first order contour derivative, and second order contour derivative) contours for each mouse movement along the x-y plane to recognize affect. This includes irritation, annoyance, reflectiveness, and neutral affect states. Results show that the temporal aspects have less contribution than the geometrical information in recognizing affect state.

Tsoulouhas et al [7] 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.

Some studies have jointly considered both the keyboard and the mouse as the source of data to detect affect. Zimmermann et al [23] extracted 64 parameters such as total number of mouse clicks, single mouse clicks (multiple clicks counted as one click), total distance of the mouse pointer, mouse speed, median click time (time between pressing and releasing a mouse button), number of pauses in the mouse movement, median distance of a single mouse movement, mouse acceleration, angle and direction of mouse movements, number of keystrokes, median length of a keystroke, etc. as parameters to measure affect. Their experiments have not been analyzed fully yet, so they only presented an overview of their preliminary results, which include the mouse and keystroke features that they used.

Rodrigues et al [24] and Lim et al [25] conducted separate studies using the keyboard and the mouse as sources of data to detect stress. The former used keystroke frequency and intensity, mouse click accuracy, mouse click duration, amount of mouse movement, mouse movement, and mouse clicks to detect stress of e-learning students. It was found out that the intensive use of the keyboard, high frequency of backspace usage, mouse clicks and scroll usage are indicators of stress. The latter explored the average key latency, average typing speed per key, backspace key, delete key, average mouse speed, total mouse inactivity duration, total mouse inactivity occurrences, left click rate, and right click rate as measures of stress. The study showed that the average key latency, the average typing speed per key, the average mouse speed, the total mouse inactivity duration, and the left click rate are the significant features in detecting stress.

Khan et al [26] extracted self-reported arousal (strength of the emotion) and valence (pleasantness of the emotion) values from the collected log files, and some keyboard/mouse behavior within 10 minute windows. The basic measures taken for each window were: the self-reported valence and arousal values that a participant provided, the total number of events around a particular mood rating, the average time between events, the total windows switched, the standard deviation of the time between events, the number of backspace and delete key events, the number of alphabetical and numerical key events, the number of mouse clicks, and the number of all other keys. The main focus in this study is on keyboard and mouse click processing and did not include the mouse movements or the distances between clicks. Further, the authors did not find support for a generic measure of mood through user interaction behavior.

A more comprehensive study on affect detection in terms of its two dimensions was presented by Salmeron-Majadas et al [27][28]. 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 [15] and Bixler and D'Mello [21], and the digraph and trigraph used by Epp et al [17]. 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.

# 3. METHODOLOGY

## 3.1 Participants

The participants in this study were around 60 volunteers from three (3) sections of first year or second year 16-18 years old students of a higher educational institution in Makati City. However, due to some technical problems, only 55 participants have both key-mouse logs and video logs that are needed in the study. At the time of the study, the students were enrolled in CS126 - Programming 1 with no or minimal background in C++. All these volunteers were given waivers to parents or guardians, asking their permission to let their child participate in the study. Hence, only those students with consent from their parents or guardians were allowed to participate.

CS126 is a first year introduction to programming course using structured 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

Preparations prior to the data gathering included the securing of administrative approval and the setting up of the student environment. We requested permission from school authorities to use the school's facilities. Upon securing permission, we installed the customized mouse-key logger, the CamStudio screen casting program, web cam drivers, the Microsoft Movie Maker, and Dev-C++ Integrated Development Environment.

Before the student works on its programming activity, the web cam is already properly in place and turned-on (Figure 1), the mouse-key logger, the Movie Maker, and the CamStudio were set and running in the background but is hidden from the student in order not to bother him/her while he/she is doing the programming activity.

**Figure 1. Data gathering setup.**

The mouse-key logger captured the mouse motion, mouse clicks, and mouse scrolls and the key event logs. The web cam captured the facial expressions and body movements of the student. The captured video (video logs) were used in labeling student affect. CamStudio captured the whole screen (screen logs). This was used primarily to match the observation time of the mouse-key logs, and the video logs. Dev-C++ was used as the programming environment in doing the programming activities.

Data was collected from three (3) CS126 classes where the problems are about selection constructs and loop constructs, respectively. Data was gathered simultaneously from around twenty participants per section and it took more than two (2) hours data recording.

## 3.3 Data Processing

Data processing encompasses the conversion of the collected data into a complete mapping of the low fidelity data (mouse-key logs) with the high fidelity data (video logs). The results were used as the dataset of this study. This includes several steps: First step was to clean the data by removing segments in the mouse-key logs that had no corresponding video logs. The second step was to extract potential features from the raw data of the mouse-key logs (Table 1) that may have contributed to the recognition of affect states of the student.

**Table 1. Sample Raw Data of the Mouse-Key Log.**

| | A | B | C | D | E |
|---|---|---|---|---|---|
| 261 | 36979.074 | IdleTime | 0 | 0 | 0 |
| 262 | 36979.094 | IdleTime | 0 | 0 | 0 |
| 263 | 36979.095 | KeyUp | NumPad6 | 0 | 0 |
| 264 | 36979.114 | IdleTime | 0 | 0 | 0 |
| 271 | 36979.236 | IdleTime | 0 | 0 | 0 |
| 272 | 36979.239 | KeyDown | Return | 0 | 0 |
| 273 | 36979.239 | KeyPress | Return | 0 | 0 |
| 274 | 36979.246 | IdleTime | 0 | 0 | 0 |
| 277 | 36979.302 | IdleTime | 0 | 0 | 0 |
| 278 | 36979.308 | IdleTime | 0 | 0 | 0 |
| 279 | 36979.319 | KeyUp | Return | 0 | 0 |
| 280 | 36979.328 | IdleTime | 0 | 0 | 0 |
| 317 | 36979.916 | IdleTime | 0 | 0 | 0 |
| 318 | 36979.936 | IdleTime | 0 | 0 | 0 |
| 319 | 36979.953 | MouseMove | 466 | 319 | 0 |
| 320 | 36979.956 | IdleTime | 0 | 0 | 0 |

We extracted all the mouse and keystroke features identified in previous literatures, plus added other features that may be related to affect detection. These were grouped into three feature-sets: first set, include thirty (30) keystroke verbosity features such as typing speed in terms of keys pressed, typing speed in terms of the number of characters typed and some of its statistical equivalent, time taken for typing, number of times the backspace is pressed, the delete key is being pressed, and its combination (total error in typing), idle time, etc.; the second set include ninety two (92) keystroke durations and latency features of digraph keystroke (2G) and trigraph keystroke (3G) such as SUM_2G_1D2D (total duration between the 1st and 2nd down keys of the digraph), SUM_2G_1Dur (summation of all durations of the 1st key of the digraph), SUM_2G_1keylat (latency time between the 1st keyup and next keydown of digraph), SUM_3G_2D3D (total duration between the 2nd and 3rd keydown of the trigraph) , SUM_3G_Dur (total duration from the 1st keydown to the last keyup of the trigraph), SUM_3G_2keylat sum of the duration between the 2nd keyup and next keydown of the trigraph (), etc.; and the third set is composed of twenty eight (28) mouse features such as number of mouse movements, total distance move, average mouse speed, total number of left and/or right clicks, total number of double left and/or double right clicks, mouse scrolls, mouse activity duration, etc.

The extracted features were saved in a comma separated value (csv) file containing keyboard dynamic and mouse behavior features at every 15-second interval. This file was called the "incomplete dataset" since the affect is not yet labeled.

The third step was to divide the video logs into 15-second video time segments that correspond to mouse-key time segments in the incomplete dataset. This was done by observing the time in the screen logs. The video segment where the participant showed curiosity about being monitored through the camera or not seen in the video was marked "X" and the instance that corresponds with the time frame in the "incomplete dataset" was deleted. After the video segments were all prepared, affect labeling took place by mapping the observed affect in the video segment (high fidelity data) that has the same 15-second mouse-key time frame instance (low fidelity data) in the incomplete dataset (Figure 2).

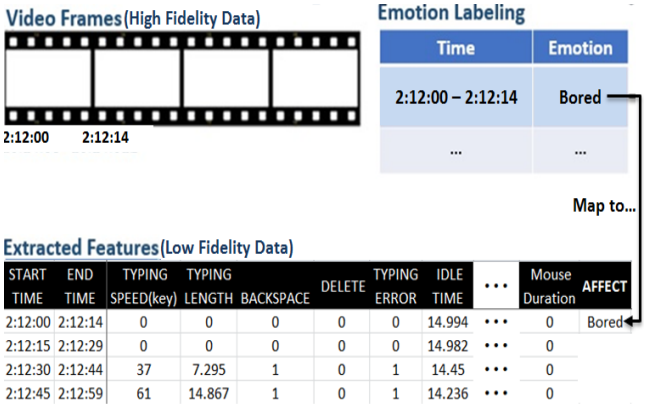


**Figure 2. Mapping High Fidelity data with the Low Fidelity data**

Determining of student's affect from the video segment was based on the modified coding scheme adopted from [3][10][29] and is presented in Table 2.

**Table 2. Affective State Criteria**

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

The scheme was modified to find the state of confusion (negative valence, positive arousal), boredom (negative valence, negative arousal), frustrated, and a special emotion state labeled as "others" which is not within the scope of the study. The state "others" was made under the premise that the emotion with respect to the time frame was found to be neither confused, bored, nor frustrated.

## 3.4 Model Development and Data Analysis Methods

When the datasets were established, we explored on these and tried 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 as J48, Decision Tree, and Random Forest. Each classifier were trained, validated and tested using (1) keystroke verbosity features alone, (2) keystroke duration and latency features of 2G and 3G alone, (3) all keystroke features which comprises verbosity plus duration and latency features, (4) mouse features alone, and (5) combined all keystroke and mouse features.

To select the acceptable affective models, the results of the classifiers were analyzed in terms of accuracy rate and kappa statistic.

The tree models were further analyzed to find the notable features that help out in the recognition of negative affect states of novice programming students and how these features are related to student's affect.

## 3.5 Results and Discussion

After conducting data processing of Section 3.3, four datasets were derived (see Table 3). Every fifth student in the list of participants was chosen as part of the test set.

**Table 3. The Different Datasets.**

| Dataset | Number of Participants for Training Set | Number of Participants for Test Set | Total Number of Participants |
|---|---|---|---|
| CS126L-AT2 | 14 | 4 | 18 |
| CS126L-BT1 | 13 | 4 | 17 |
| CS126L-BT2 | 16 | 4 | 20 |
| ALL CS126L | 43 | 12 | 55 |

As stated in Section 3.4, and by using RapidMiner, some classifiers were trained using gini index criterion and validated using Batch-X-Validation to allow student-level cross-validation. The main dataset (ALL CS126L) was used to find the classifier that gives the most acceptable model in terms of kappa statistic and/or accuracy rate.

As shown in Table 4, using keystroke duration and latency features on 2G and 3G alone, as well as mouse features alone do not provide a good model to detect negative affect since the kappa is very low (less than 0.2) which implies a slight agreement [30].

It was also observed that Decision tree classifier (highlighted row) consistently provide the highest kappa and accuracy. It implies that in this experiment, the Decision tree classifier gave the most acceptable model. Lastly, the kappa and accuracy of the other feature-sets (keystroke verbosity, all keystroke features, and combined all keystroke and mouse features) are statistically tied. And since the kappa is in moderate agreement [30], it implies that these feature-sets can be used to model negative affect detector.

Thus, 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 presented in Table 5.

**Table 4. Statistical measures for model performance using some well-known tree-based classification algorithms.**

| Feature-Set | Classifier | Depth of the tree | Kappa statistic | Accuracy rate (%) |
|---|---|---|---|---|
| Keystroke verbosity | J48 | N/A | 0.472 | 69.98 |
| | Random Forest | 7 | 0.093 | 59.3 |
| | Decision tree | 6 | 0.493 | 70.80 |
| Keystroke duration & latency | J48 | N/A | 0.093 | 54.98 |
| | Random Forest | 7 | 0.030 | 57.04 |
| | Decision tree | 10 | 0.103 | 56.72 |
| All keystroke features | J48 | N/A | 0.454 | 69.06 |
| | Random Forest | 5 | 0.035 | 57.33 |
| | Decision tree | 4 | 0.489 | 71.03 |
| Mouse features | J48 | N/A | 0.014 | 54.24 |
| | Random Forest | 3 | 0.000 | 56.65 |
| | Decision tree | 5 | 0.003 | 56.94 |
| Combined keystroke & mouse features | J48 | N/A | 0.434 | 67.78 |
| | Random Forest | 7 | 0.078 | 58.77 |
| | Decision tree | 4 | 0.490 | 71.06 |

**Table 5. Result in testing the models generated by the Decision tree classifier**

| Feature-Set | Depth of the tree | Kappa statistic | Accuracy rate |
|---|---|---|---|
| Keystroke verbosity | 7 | 0.564 | 74.08 |
| All keystroke features | 6 | 0.568 | 74.28 |
| Combined keystroke and mouse features | 6 | 0.567 | 74.23 |

Table 5 shows that in the testing phase, the kappa statistic and the accuracy rates significantly increased but are statistically tied. This verifies that the three (3) feature-sets can be used to model negative affect detectors of novice programming students.

To determine the notable features that help out in the recognition of negative affective states and how these features are related to student's affect, the tree models generated from the above 3 feature-sets were analyzed. This was done by listing the unique inner nodes of the decision tree models. Initial result shows that some of the notable features are highly correlated. Hence, another experiment was again undertaken by removing some features that are highly correlated to other features. It was observed that the kappa and accuracy slightly improved (see Table 6). The table shows that the kappa in all the feature-sets are almost equal and the accuracies have slight differences. This implies that the notable features from the keystroke verbosity feature-set are enough to model a negative affect detector of novice C++ programming students. However, to improve slightly the prediction rate of the model, the MAX_3G_1Dur and SUM_2G_1Dur from the keystroke duration and latency of the digraph (2G) and trigraph (3G) feature-set, and MM_Total_X mouse feature should be added.

**Table 6. Result in testing the models generated when some correlated features were removed.**

| Feature -Set | Depth | Kappa | Accuracy | Notable Features |
|---|---|---|---|---|
| KV | 6 | 0.569 | 74.23 | Typing Error, Typing Variance, Idle Time, Total Key Events, F9 |
| All KF | 6 | 0.568 | 74.28 | Typing error, Typing variance, Idle time, Total keyevents, F9**, MAX_3G_1Dur, AVE_3G_2D3D**, and SUM_2G_1Dur |
| All F | 6 | 0.572 | 74.37 | Typing error, Typing variance, Idle time, Total keyevents, F9**, SUM_2G_1Dur, AVE_3G_2D3D**, and MM_Total_X |

*KVF – Keystroke verbosity features
*All KF – All keystroke features
*All F – Combined keystroke and mouse features
*F9 – the number times F9 key was pressed
*SUM_2G_1Dur – sum of all durations of the 1st key of the digraph
*AVE_3G_2D3D– the average duration time between the 2nd and 3rd keydown of the trigraph
*MM_Total_X – total distance travelled by the mouse along the x-axis
** Optional feature that when removed, it does not affect the performance of the model. However, it adds additional branches in the decision tree

Finally, to determine how student's affect related to keyboard dynamics and mouse behavior, the unique paths from the root of the decision tree using the combined keystroke and mouse features to its leaves were analyzed. The said decision tree model is shown in Figure 3.

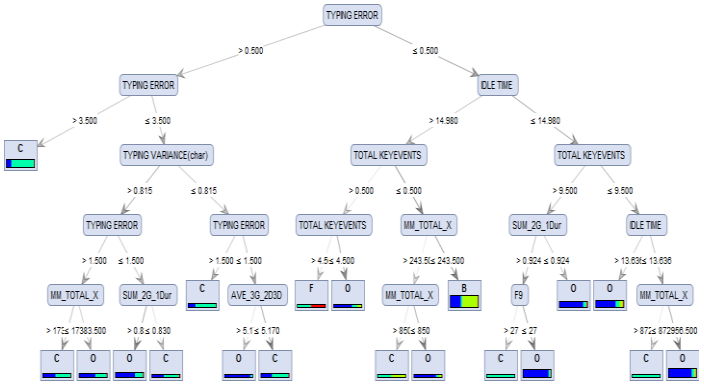


**Figure 3. A sample generated decision tree model.**

As shown in Figure 3, student affect is related in terms of 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), the number of keyevents (keydown + keypress + keyup) he/she executed in the keyboard, the student's typing variance (his/her typing varies with time), the number times F9 key (shortcut to compile and run the program) was pressed, the sum of all durations the student acted on the 1st key of the digraph, the average duration time between the 2nd and 3rd

keydown of the trigraph, and total distance he/she moved the mouse along the x-axis.

## 4. CONCLUSION

It must be noted that this study tries 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? (3) how is student's affect related to mouse behaviors? (4) will the combined features from keystrokes and mouse movements provide better predicting model than using keystroke features alone, or mouse behaviors alone? (5) to what extent do the models correctly predict novice programmer's affect? Hence, this section addresses the research questions as follows:

(1) the notable features from keyboard dynamics and/or mouse behaviors that help out in the recognition of negative affect states of novice programming students were presented in Table 6 where: the keystroke dynamics are the Typing Error, Typing Variance, Idle Time, Total Keyevents, SUM_2G_1Dur, AVE_3G_2D3D, and F9 while the mouse behavior is the total distance travelled by the mouse along the x-axis.

(2) as shown in Figure 3, student's affect is related to keyboard dynamics in terms of 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), the number of keyevents (keydown + keypress + keyup) he/she executed in the keyboard, the student's typing variance (his/her typing varies with time), the number times F9 key (shortcut to compile and run the program) was pressed, the sum of all durations the student acted on the 1st key of the digraph, and the average duration time between the 2nd and 3rd keydown of the trigraph.

(3) also shown in Figure 3, student affect related to mouse behaviors in terms of the total distance the student moved the mouse along the x-axis.

(4) Table 6 shows that the kappa of the three feature-sets are almost equal but the accuracy slightly increased when a mouse feature was added. Hence, combining keystrokes and mouse movement features provide slightly better predicting model than using keystroke features alone, and significantly better than using mouse behaviors alone as shown in Table 4.

(5) As shown in Table 6, the prediction rates of the models generated by Decision tree classifier using the three feature-sets are statistically tied to around 74.3%.

## 5. ONGOING AND FUTURE WORKS

We are still working on quantifying the relationship of the student's affect with the keystroke and mouse features by analyzing the weights of the edges of each decision tree.

Also, to extend this research, we will also try the following experiments: (1) divide the data set into students with low / medium / high incidences of boredom, confusion, frustration, and see how features differ among the three groups; (2) divide the data set by time and look at the data at first 1/3 of the observation period, the second 1/3 of the observation period, and the last 1/3 of the observation period and check if the features are "stable"; and (3) look at high-boredom / confusion / frustration students vs. low boredom / confusion / frustration students if their features differ or similar over time.

## 6. REFERENCES


[1] Affect. Encyclopedia of Mental Disorders. URL=http://www.minddisorders.com/A-Br/Affect.html..

[2] Combs, H. Psychiatry Clerkship. UW School of Medicine - Department of Psychiatry and Behavior Sciences. URL=http://depts.washington.edu/psyclerk/ glossary.html.

[3] Carlos, C.M., Delos Santos, J.E., Fournier, G. and Vea, L. 2013. Towards the Development of an Intelligent Agent for Novice Programmers through Face Expression Recognition. 13th Philippine Computing Science Congress (2013).

[4] March, J.D. 2010. Affective Priming in Music and Words. Master thesis. School of Graduate Studies, Department of Psychology, Memorial University of Newfoundland, Canada. URL=http://216123216456.info/jamiemarch/thesis/index.php?.

[5] Picard R.W. And The Medial Lab – Affective Computing Group. Affective computing. URL=http://affect.media.mit.edu/.

[6] Darling-Hammond, L., Orcutt, S., Strobel, K., Kirsch, E., Lit, I. And Martin, D., With Contributions From Comer, J. Feelings Count - Emotions and Learning. The Learning Classroom: Theory and Practice. URL=http://www.learner.org/courses/learningclassroom/support/05_emotions_learning.pdf

[7] Tsoulouhas, G., Georgiou, D., And Karakos, A. 2011. Detection of Learner's Affective State Based on Mouse Movements. Journal of Computing 3, 11 (2011), 9-18.

[8] Marchand, G. C., And Gutierrez, A. P. 2012. The role of emotion in the learning process: Comparisons between online and face-to-face learning settings. Internet and Higher Education 15 (2012), 150–160.

[9] Bosch, N., D'Mello, S., and Mills, C. 2013. What Emotions Do Novices Experience during Their First Computer Programming Learning Session? Proceedings of the 16th International Conference on Artificial Intelligence in Education (2013), 11–20.

[10] Rodrigo, M.M.T., Baker, R.S.J. D, Jadud, M.C., Amarra, A.C.M., Dy, T., Espejo-Lahoz, M.B.V., Lim, S.A.L., Pascua, S.A.M.S., Sugay, J.O., Tabanao, E.S. 2009. Affective and behavioral predictors of novice programmer achievement. Proceedings of the 14th annual ACM SIGCSE conference on innovation and technology in computer science education 41, 3 (2009), 156-160.

[11] Robins, A., Rountree, J., And Rountree, N. 2003. Learning and teaching programming: A review and discussion. Computer Science Education 13, 2 (2003), 137-172.

[12] D'Mello, S. and Graesser, A. 2012. Dynamics of affective states during complex learning. Learning and Instruction. 22, 2 (2012) 145–157.

[13] Shuhidan, S., Hamilton, M. And D'souza, D. 2009. A Taxonomic Study of Novice Programming Summative Assessment. Proceedings of the Eleventh Australasian Conference on Computing Education 95 (2009), 147-156.

[14] Tsihrintzis, G.A., Virvou, M., Alepis, E., and Stathopoulou, I-O. 2008. Towards Improving Visual-Facial Emotion Recognition through Use of Complementary Keyboard-Stroke Pattern Information. Proceedings of the Fifth

International Conference on Information Technology: New Generations (2008), 32-37.

[15] Khanna, P. and Sasikumar, M. 2010. Recognising emotions from keyboard stroke pattern. International Journal of Computer Applications 11, 9 (2010), 1-5.

[16] Felipe, D. A., Gutierrez, K.I, Quiros, E.C., and Vea, L. 2012. Towards the Development of Intelligent Agent for Novice C/C++ Programmers through Affective Analysis of Event Logs. Proceedings of the International MultiConference of Engineers and Computer Scientists 1 (2012), 511-518.

[17] Epp, C., Lippold, M., and Mandryk, R. L. 2011. Identifying emotional states using keystroke dynamics. Proceedings of the Annual Conference on Human Factors in Computing Systems 189 (2011), 715–724.

[18] Tsui, W-H., Lee, P., and Hsiao, T-C. 2013. The effect of emotion on keystroke: An experimental study using facial feedback hypothesis. Proceedings of the 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (2013), 2870-2873.

[19] Shukla, P. and Solanki, R. 2013. Web Based Keystroke Dynamics Application for Identifying Emotional State. International Journal of Advanced Research in Computer and Communication Engineering 2, 11 (2013), 4489-4493.

[20] Solanki, R. and Shukla, P. 2014. Estimation of the User's Emotional State by Keystroke Dynamics. International Journal of Computer Applications 94, 13 (2014), 21-23.

[21] Bixler, R. and D'Mello, S. 2013. Detecting Boredom and Engagement During Writing with Keystroke Analysis, Task Appraisals, and Stable Traits. Proceedings of the International Conference on Intelligent User Interfaces (2013), 225–234.

[22] Schuller, B., and Rigoll, G. 2004. Emotion recognition in the manual interaction with graphical user interfaces. Proceedings of the IEEE International Conference on Multimedia and Expo 2, (2004), 1215-1218.

[23] Zimmermann, P., Guttormsen, S., Danuser, B., And Gomez, P. 2003. Affective computing - a rationale for measuring mood with mouse and keyboard. International Journal of Occupational Safety and Ergonomics 9, 4 (2003). 539-51.

[24] Rodrigues, M., Gonçalves, S., Carneiro, D., Novais, P., and Fdez-Riverola, F. 2013. Keystrokes and Clicks: Measuring Stress on E-learning Students. Advances in Intelligent Systems and Computing 220 (2013), 119-126.

[25] Lim, Y. M., Ayesh, A. And Stacey, M. 2014. Detecting cognitive stress from keyboard and mouse dynamics during mental arithmetic. Science and Information Conference (2014), 146-152.

[26] Khan, I.A., Brinkman, Wp., And Hierons, R. 2013. Towards Estimating Computer Users' Mood from Interaction Behaviour with Keyboard and Mouse. Frontiers of Computer Science 7, 6 (2013), 943-954.

[27] Salmeron-Majadas, S., Santos, O., and Boticario, J. 2014. An evaluation of mouse and keyboard interaction indicators towards non-intrusive and low cost affective modeling in an educational context. Procedia Computer Science 35, 691-700.

[28] Salmeron-Majadas, S., Santos, O., and Boticario, J. 2014. Exploring indicators from keyboard and mouse interactions to predict the user affective state. Proceedings of the 7th International Conference on Educational Data Mining (2014), 365-366.

[29] Dragon, T., Arroyo, I., Woolf, B.P., Burleson, W., Kaliouby, R., and Eydgahi, H. 2008. Viewing Student Affect and Learning through Classroom Observation and Physical Sensor. Proceedings of the 9th international conference on Intelligent Tutoring System (2008), 524-531.

[30] Viera, A.J. and Garrett, J.M., 2005. Understanding interobserver agreement: the kappa statistic. Fam Med, 37, 5 (2005), 360-363.