Toward Computer-Mediated Emotional Monitoring and Burnout Mitigation for University STEM Students

London Bielicke (Mentor: Mathew Mauriello)

Keywords: HCI, stress, students, STEM, burnout, programming, cognitive load, AER

ABSTRACT

Recent research shows that many STEM University students face burnout and poor mental health, however, there is little research on why and when students feel stressors during tasks. Student's mental well-being is a recurring issue that is exacerbated by the pandemic and other stressors of the fast-paced modern work environment. Decreased state of well being from experiencing repeated stress and negative emotions can lead to burnout and decreased performance in students and employees. While the goal is to maximize quality, overworking students without allowing time to recuperate from stress may lead to less efficient work. Furthermore, stress from burnout will decrease the quality of the student's or employee's life outside of the workplace. To mitigate stress, it is important to know when stress reactions and changes in emotion occur. By understanding emotional and stress responses better we can make adaptive software that can improve user well-being. Many measurements of stress involve reflecting on past events, which does not allow us to see the subliminal factors that may contribute to these stress responses. We aim to track users' computers to create a tool that predicts emotion and stress changes in STEM students. Using these predictions, our final application will generate just-in-time interventions when our algorithm predicts that a user has been experiencing negative emotions or stress for too long as an attempt to prevent burnout in university STEM students.

INTRODUCTION

Preventing burnout in students is essential to optimize productive study time as well as enhance students' way of life. Mitigating stress is only possible when we know the cause of students' stress. To better understand the context in which stress occurs, we will use context features involving keyboard and mouse actions, active applications and browser tabs, and facial recognition to predict emotion and stress levels based on manually entered user labels. Using these labels, we will try to map different contextual features to stress and emotions on both a general and individual level. The predictions will be used to generate just-in-time interventions to reduce acute stress and mitigate future burnout. We will test different groups to see whether specific interventions such as going for a walk, or telling the user to take a break with their personal preferred form of relaxation is more effective.

SYSTEM DESCRIPTION

OVERVIEW

The Student Logging tool is comprised of 4 main components: key-logging, mouse-logging, context-logging and emotion-logging. The components collect data in realtime and are programmed in Python. The data collected are stored in CSV files and are later used to compute the 13 features used for stress and emotion prediction and are described in table 1. The 13 features are converted to a .aarf file including the manually entered user emotions as class labels. This file is used in Weka where we use different machine learning algorithms to best predict the user's stress and emotion levels at different time intervals.

KEY-LOGGER

The keylogger takes real-time updates from the user's keyboard interactions using the Keyboard library. The actions are stored in a locally stored csv file, where each line writes the timestamp of the action, the type of action (UP, DOWN), and the type of key (KEY, SPACE, BACKSPACE).

MOUSE-LOGGER

The mouse-logger takes real-time updates from the user's mouse interactions using the PYNPUT library. The actions are stored in a csv file, where each line writes the timestamp of the action, and the type of action (MOVE, CLICK, SCROLL). Move and click actions also write the coordinates of the mouse, while scroll actions log the scroll vector indicating the velocity of the scroll.

CONTEXT-LOGGER

Context logs consist of the user's system information and their interactions with applications and browser tabs on their computer. This component takes updates on one-minute intervals that log the timestamp, the application in focus, and the number of tabs open in Chrome.

EMOTION LOGGING

The emotion detection algorithm stores the predicted emotion as well as the confidence intervals of all seven emotions of the user at a given frame over the entire session. The emotion is detected every set amount of frames. We are using Python's deepface module to make our predictions.

Feature	Description	Output type
Clicks Per Minute	tracks the rate at which a user clicked their mouse over a given interval	float
Mouse Movement	tracks the total movement of the user's mouse over a given interval. Computed using euclidean distance between each change in curser coordinates (NOTE: measured in pixels)	float
Average Mouse Speed	tracks the average speed of the user's mouse (tracked only during periods of movement)	float
Average Scroll Velocity	tracks the average scroll velocity over a given interval, tracked only during periods of scrolling.	float
Changes in Direction	tracks the number of times the user changed directions of their scroll on a given interval (ie. scrolling down to scrolling up)	int
Mouse Click Length	tracks average mouse click duration, which is tracked from down click of mouse until the user releases the mouse button	float
Average Click Delay	tracks the average duration between clicks, which is tracks from up movement of mouse button to next down movement	float
Keys Per Minute	tracks the rate at which the user is typing	float
Average Keypress Length	tracks the duration that a user holds down a key on average, computed from keydown event to keyup event	float
Most Keys Down	tracks the most amount of keys that a user hekd down during a given interval	int
Changes in Emotion	tracks the amount of times a user's emotion changed over a given interval	int
Most Frequent Emotion	tracks the most frequent emotion that a user was labeled as over a given interval	String/Nominal
Number of Chrome Tabs	tracks the most number of chrome tabs that a user had open during a given interval	int
Active Window	tracks the most frequent window that a user had in focus over a given interval	String/Nominal
Active Window Changes	tracks the number of times the user focused on a different window over a given interval	int

Figure 1.

Table 1: Descriptions of all features collected

REAL-TIME ANALYSIS

Upon exiting the session, the logs are fed into a function which parses the data and runs analysis to compute the 13 features. These features are then used to display various graphs about their interactions and changes in activity over the session. The visuals are also stored on their computer for future reference. This is both for our use to analyze the data as well as for the user to notice patterns or subconscious tendencies and correlations in their behavior. Lastly I developed a dashboard for the user to interact with after their session. The dashboard includes highlights such as their most used windows and which emotion they were most often labeled as while using that window, the amount of active time spent during the session. Their most frequent emotion label over the entire session, and a graph visualizing their change in emotion.

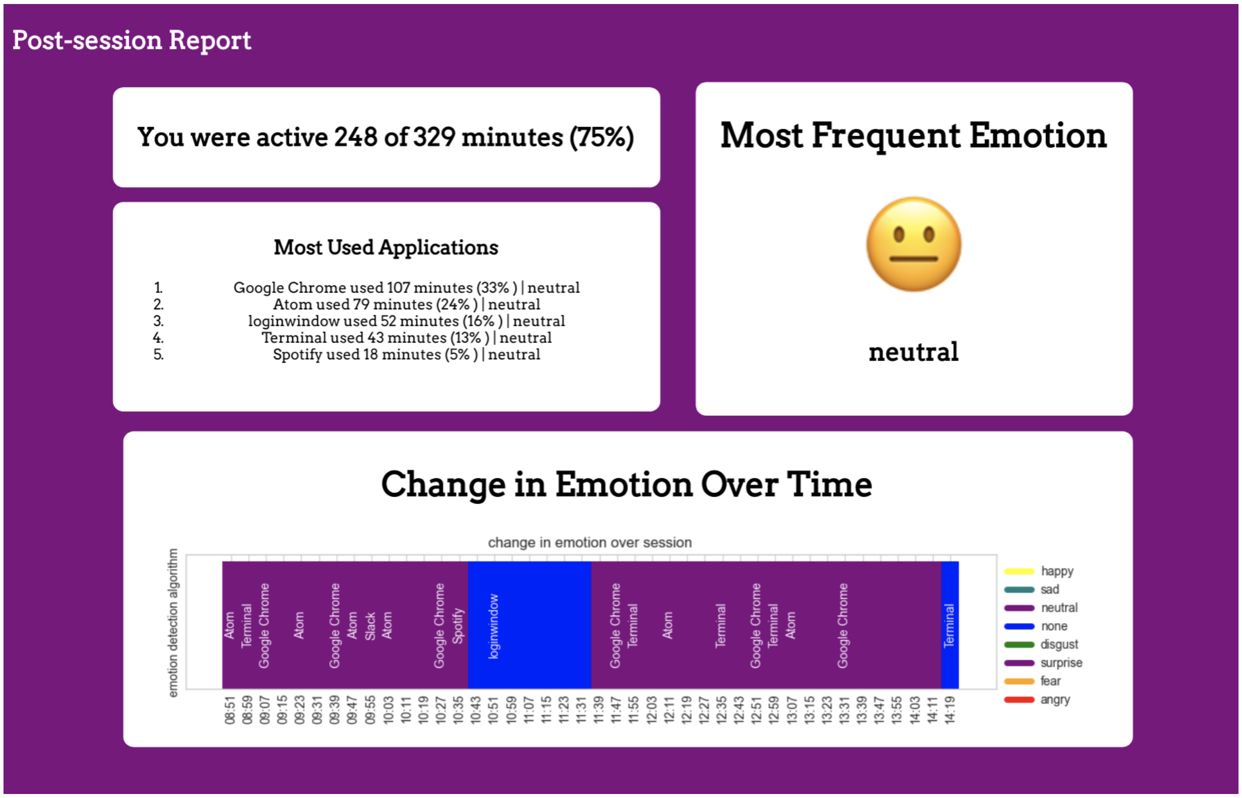


Figure 2. User dashboard generated post-session

RELATED WORK

Emotion detection utilizes facial recognition to predict emotion, however much of this data is trained on a non-representative population. This results in demographics such as non-white users and females receiving less accurate labels. We attempt to add other context features to help mitigate these biases in the data that we see from past research.

Other studies show that emotion detection is inaccurate unless given further environmental cues[1]. These studies reference the difference between one's smile while taking a photo and one's smile while laughing at a joke. Because we are testing STEM students during work, our environment is narrow and helps us better contextualize the change in emotion.

DATA ANALYSIS

At this time, we have not started pilot testing on our most updated application, however I have run data tested on myself and other lab members without the manually entered emotion labels. By running this data in Weka, we can start to identify redundant attributes and if there is a significant connection between the facial recognition emotion label and other context features. The biggest problem with our small sample dataset is that a vast majority of class labels were neutral. The J48 decision tree, Random Forest, and K-means clustering algorithms predicted emotion with high accuracy but low F-measure before preprocessing the data. To counteract this, I manipulated the data by merging emotion class labels into the categories neutral, negative, and positive. Although we had a small set of data to look at, the J48 decision tree gave us insight as to which features may lead to certain emotions. For example, instances with an average mouse movement greater than 1255 pixels per second more often led to a classification of Sad, whereas instances with less mouse movement were more often labeled as neutral. Outside of Weka, I also generated correlation matrices of the features. I also generated graphs for each session showing how the attributes changed over time. Lastly, I generated a dashboard for the

```
RandomForest

Bagging with 100 iterations and base learner

weka.classifiers.trees.RandomTree -K 0 -M 1.0 -V 0.001 -S 1 -do-not-check-capabilities

Time taken to build model: 0.12 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         131               92.9078 %
Incorrectly Classified Instances        10                7.0922 %
Kappa statistic                          0.6933
Mean absolute error                      0.063
Root mean squared error                  0.1768
Relative absolute error                 46.7406 %
Root relative squared error             69.9545 %
Total Number of Instances              141

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.975    0.350    0.944      0.975    0.959      0.688    0.905     0.971     neutral
                 0.000    0.007    0.000      0.000    0.000      -0.010   0.291     0.014     happy
                 0.333    0.015    0.500      0.333    0.400      0.387    0.971     0.480     negative
                 0.917    0.000    1.000      0.917    0.957      0.954    1.000     1.000     none
Weighted Avg.    0.929    0.301    0.916      0.929    0.922      0.688    0.907     0.939

=== Confusion Matrix ===

   a   b   c   d   <-- classified as
 118   1   2   0 |   a = neutral
   2   0   0   0 |   b = happy
   4   0   2   0 |   c = negative
   1   0   0  11 |   d = none
```

Figure 2: WEKA Random Forest results. 92.91% accuracy and a weighted F-measure of 92.2.

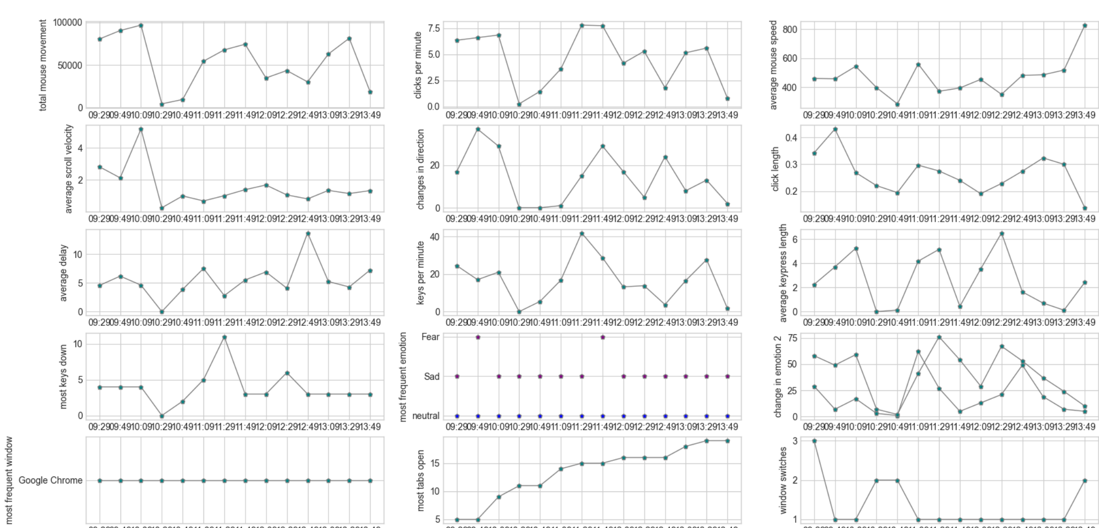
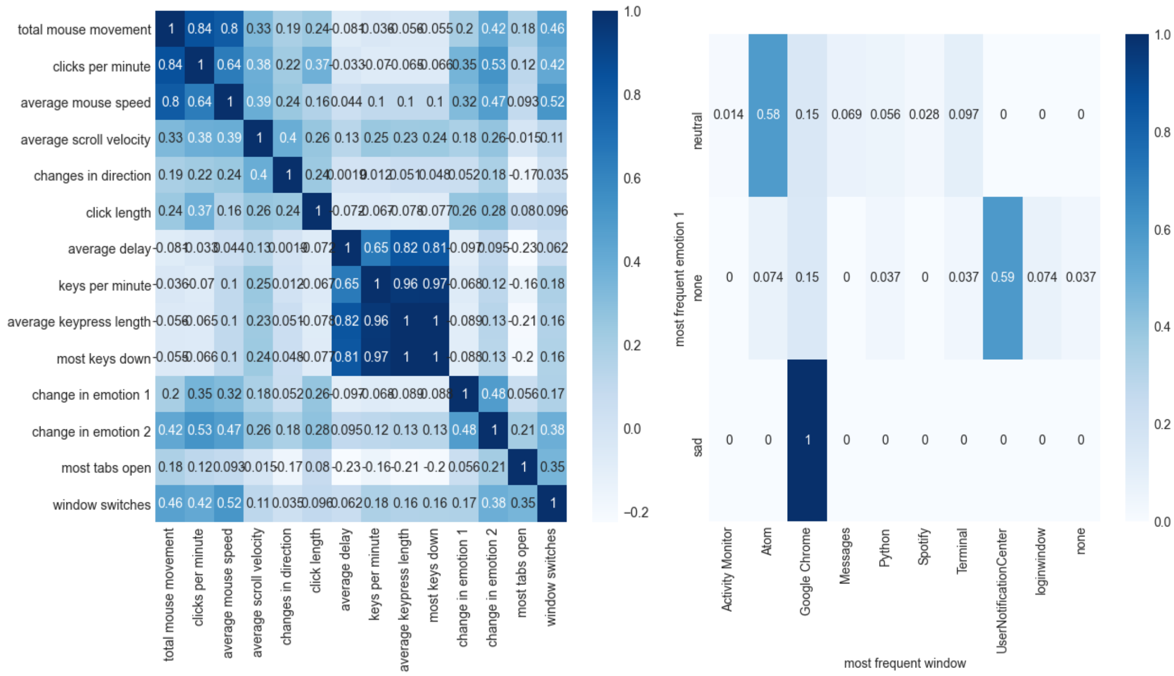


Figure 3. Graphs generated post session displaying changes over given intervals



Figures 4 and 5. Correlation matrices visualizing relationships between different features.

LIMITATIONS AND FUTURE WORK

Privacy is an increasing concern, which may hinder people from utilizing an application which tracks and stores data about their interactions with work and their environment, especially since a face cam must be active to track the data. Furthermore, the facecam may be distracting during everyday use, and it may alter behavior observed in an environment where they are not normally tracked. This may play a factor in inaccurate labels during testing as well since the prompts may be distracting or alter their mood by making them aware of their current state. This plays further into the problem of manually entered emotion labels because humans are not always aware of their emotions, and their emotions may be more complex than one of the seven emotion labels.

However, this tool could be used to allow STEM students to become more aware of their interactions with their environment and emotions from only a few sessions. As a result, students may be more intentional and aware during other sessions when they are not being tracked given the data we provide with our tool. Furthermore, this data could be used by employers or teachers to understand when certain tasks or assignments increase stress or when burnout starts to occur.

While we had decent results from the random forest algorithm in WEKA, we need more data to better understand the connection between these features. If the data is heavily skewed in our final dataset, we will likely benefit from using a random subset of the neutral class, or oversampling of the less frequent class labels.

REFERENCES

[1] Harmanpreet Kaur, Daniel McDuff, Alex C. Williams, Jaime Teevan, and Shamsi T. Iqbal. 2022. “I Didn’t Know I Looked Angry”: Characterizing Observed Emotion and Reported Affect at Work. In Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems (CHI '22). Association for Computing Machinery, New York, NY, USA, Article 199, 1–18.

[2] Nunes, A. R. V. (2019). *Deep Emotion Recognition through Upper Body Movements and Facial Expression* (Doctoral dissertation, Master’s Thesis, Aalborg University).

[3] Qi, Y., Jia, W., & Gao, S. (2021, June). Emotion recognition based on piezoelectric keystroke dynamics and machine learning. In *2021 IEEE International Conference on Flexible and Printable Sensors and Systems (FLEPS)* (pp. 1-4). IEEE.

[4] Shah, S., Teja, J. N., & Bhattacharya, S. (2015). Towards affective touch interaction: predicting mobile user emotion from finger strokes. *Journal of Interaction Science*, *3*(1), 1-15.

[5] Yang, K., Wang, C., Gu, Y., Sarsenbayeva, Z., Tag, B., Dingler, T., ... & Goncalves, J. (2021). Behavioral and physiological signals-based deep multimodal approach for mobile emotion recognition. *IEEE Transactions on Affective Computing*.

[6] Zimmermann, P., Guttormsen, S., Danuser, B., & Gomez, P. (2003). Affective computing—a rationale for measuring mood with mouse and keyboard. *International journal of occupational safety and ergonomics*, *9*(4), 539-551.