Utilizing Passive Data Collection to Detect Anxiety and Depression

A Technical Report submitted to the Department of Computer Science

Presented to the Faculty of the School of Engineering and Applied Science
University of Virginia • Charlottesville, Virginia

In Partial Fulfillment of the Requirements for the Degree
Bachelor of Science, School of Engineering

Keshav Ailaney
Spring, 2021

Technical Project Team Members
Keshav Ailaney
Wei Wang
Aldrick Johan
Johan Ketkar

On my honor as a University Student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments

Signature  Date __5/16/21_____
Keshav Ailaney

Approved _____ Date ___5/16/21_____
Afsaneh Doryab, Department of Systems and Information Engineering

Utilizing Passive Data Collection to Detect Anxiety and Depression

ALDRICK JOHAN, University of Virginia
KESHAV AILANEY, University of Virginia
JOHAN KETKAR, University of Virginia
WEI WANG, University of Virginia

Using passive data collected from smartphones, daily behavior is modeled through images of two-dimensional data. Such is applied to k-means clustering in order to illustrate significant differences in behavior and through the use of a convolutional neural network, the images were utilized to predict both anxiety and depression among users based on PHQ and GAD scores. The experiment yielded results that were accurate up to 95% and 92% accuracy, respectively.

CCS Concepts: • **Deep Learning** → *Convolutional Neural Networks*; • **Clustering** → *Classification.*

Additional Key Words and Phrases: anxiety, depression, neural networks, clustering, smart phones

1 INTRODUCTION

Early detection of major depressive disorder or depression in an individual can lead to earlier treatment, significantly increasing the probability of recovery. However, the diagnosing and detection of depression is based on physical examination, mental evaluation, or lab testing. These methods require both time and financial resources from the patient, thus, contributing to the overall difficulties of the diagnosing process. The goal of this technical project is to utilize behavior modeling techniques to detect depression and anxiety from data collected passively from users' smartphones. Specifically, clustering and classification will be employed to complete the task. These methods will be used to distinguish the differences in users' behavioral data. This approach can greatly reduce the amount of time commitment required from the users and simplify the process of detecting depression.

Our contributions are twofold: to provide a methodology for converting behavioral data into images and to show the efficacy of using those images for behavior modeling. Accurate classifications of a user's mental state by utilizing passive data collection is valuable and can be used to enhance the lives of many. Furthermore, we believe that the methodology used to generate the images can be used to further improve behavior modeling methods. These images can be utilized with a variety of different models to deliver more accurate classifications. The report explores some of the use cases of the behavioral data images, along with prior works related to the topic of behavior modeling.

2 RELATED WORK

Machine learning algorithms have been applied in many behavioral modeling studies. For example, past attempts applied several classification and clustering techniques to model behavior for mental health [1]. The dataset used in this study was unlabeled and was first analyzed with clustering algorithms to generate group labels. The clustering

Manuscript submitted to ACM

algorithms used in this process included K-means, hierarchical, and K-medoids. The generated group labels were then validated by computing the Mean Opinion Score. With the labeled data, classification algorithms were performed to build a prediction model for mental health. The algorithms involved in this process included logistic regression, Naive Bayes, support vector machine (SVM), decision tree, KNN, ensemble (bagging), and random forest tree. The results indicated that KNN, SVM, and random forest trees had achieved similar performance. The accuracy score of all three models was about 90%. However, this project had not considered utilizing a neural network model for clustering or classification.

A different study models and classifies physical human behavior using body sensors and a variety of classification techniques, respectively [2]. This study attempts to utilize such sensors to provide a more convenient means of activity detection rather than commonly implemented visual based systems which require adequate lighting and equipment among other factors. Activities such as standing, sitting, walking, and specific forms of exercise were classified using Bayesian, decision trees, least squares method, k nearest neighbors, dynamic time warping, support vector machines, and artificial neural networks. Feature extraction, reduction, and cross validation were also implemented. The most effective model was Bayesian, but many yielded high accuracy rates. For example, the ANN yielded 96.2% accuracy. However, the usage of a CNN to model the behavior of users and sensor data was not considered; however, such an approach will exhibit in the following paper.

Utilizing the same AWARE framework as our research, another study constructed a feature extraction technique with a prediction accuracy of 85% for depression [3]. This study involved several different machine learning algorithms for a variety of purposes. For example, the dataset used in this project contained several feature sets, including call and location features, and randomized logistic regression was utilized to select the feature sets most relevant for training the models. logistic regression was also used along with Gradient Boosting Classifier in the model training and validation process. Lastly, an ensemble classifier, using AdaBoost with Gradient Boosting Classifier as the base estimator, was utilized for depression detection. Classification techniques were also employed in our study; however, a CNN model was utilized for such a task.

Depression and anxiety have also been predicted through means other than behavior modeling. Many papers, for example, utilize stress as a strong indicator. Through interviews, a research group [4] explores a strong correlation between both chronic and acute stress with major depressive episodes. A different group [5] also found such a relationship between acute stress and depression. However, perfectionism among adolescents was also a predictor of depression and anxiety found in this study. Other studies utilize scientific means, which seeks to identify specific genes related to major depression [6]. The study found a relationship between variations of the 5-HTT locus and increased sensitivity of stressful life events, triggering depression among individuals.

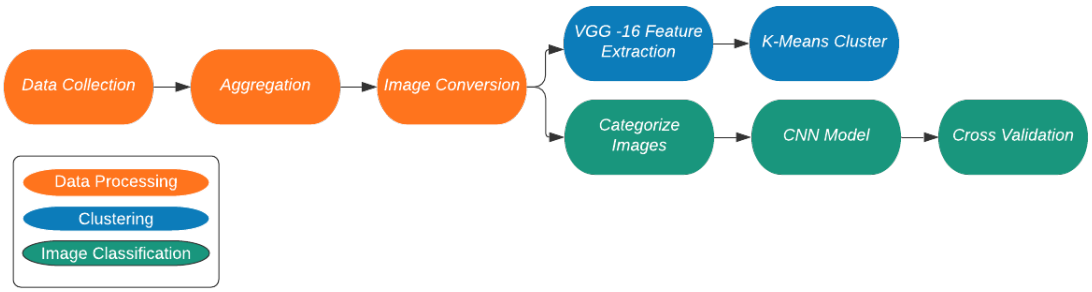

Fig. 1. Process overview

3 METHODS

This experiment was consisted of two main phases: data processing and machine learning. In the data processing phase, data was categorized, aggregated, and converted into images. Then, the images were labeled for supervised learning. In the machine learning phases, clustering and CNN models were constructed and trained with the images. An overview of the process is shown in figure 1.

3.1 Data Collection

Using AWARE [7], a framework to log user mobile activity for research, valuable data was collected by participants at Christopher Newport University in 2020 and 2021. Information relating to the user's calls, conversations, activities, location, Wi-Fi connection, and screen time were gathered by utilizing sensors and plug-ins available on the participants' devices [7] to track their behavior. The dataset contained mobile contextual information collected from 48 participants over a period of at most 24 weeks. For each sensor/plug-in type, there were certain unique features. For example, location data contained the 'double_longitude' feature and the call data contained the 'call_duration' feature. However, there were also some shared features in all types of data. These common features included 'name', 'timestamp', 'device_id', and 'device_label'. The 'name' feature identified the data type. Some example values for this feature were "calls" and "locations" for call data and location data, respectively. The 'timestamp' feature contained the number of milliseconds between 1970 and the time for when the data entry was recorded. The 'device_id' was a unique identifier assigned to each device while the 'device_lable' was an identifier assigned to each participant. Therefore, if a participant had two devices, data collected from these two devices would have different 'device_id' values but the same 'device_lable'.

As a means to understand their mental health, each user was asked to complete weekly PHQ-9 and GAD-7 surveys during the data collection process. These surveys would provide ground truth to determine their degree of depression and anxiety, respectively. PHQ-9 [8] is a self-administered survey that is consisted of 9 individual questions, and each question requires the respondent to provide a score between 0 and 3. In total, the PHQ-9 score ranges from 0 to 27, and the score is interpreted as follows: no depression (0 – 4), mild depression (5 – 9), moderate depression (10 – 14), moderately severe depression (15 – 19), and severe depression (20 – 27). The effectiveness of the PHQ-9 survey has been validated in two large studies, and it is half the length of many other evaluation methods [8]; thus, it is used as the ground truth for depression measures in this study. Similarly, the GAD-7 [9] is a self-report questionnaire for anxiety measures that is consisted of 7 items. For each item, the respondent would need to provide a score between 0 and 3,

corresponding to 'not at all', 'several days', 'more than half the days', and 'nearly every day', respectively. The total score ranges from 0 to 21, and the interpretations for the score are as follows: no anxiety (0 – 4), mild anxiety (5 – 9), moderate anxiety (10 – 14), and severe anxiety (15 – 21).

3.2 Data Preprocessing

The raw data collected using AWARE was processed before it was provided as inputs to the machine learning models. Initially, the raw data was grouped into separate comma-separated values (CSV) files based on sensor/plug-in type. For example, the location data collected from all participants was placed into a single CSV file. However, for our machine learning models, it was necessary to also categorize the data by the user or participant. This categorization can be achieved by utilizing the 'device_label' feature, which provided the participant identifier for each data entry. For each sensor/plug-in type, the data would be further separated based on the 'device_label' values. Data entries with different 'device_label' values would be placed into different CSV files. After this step was completed, each CSV file would contain data entries collected from one participant and one type of sensor/plug-in.

The next step in data processing would be to separate the data in each CSV file based on the date. The date for when a data entry was generated was determined from its 'timestamp' value. Like the previous grouping, data entries collected from different dates would be placed into different CSV files. As a result, the newly generated CSV files would contain one type of sensor/plug-in data collected from one participant and on one specific date. For example, one CSV file contained the location data collected from participant "203BN" on 2020/08/13. Lastly, the data entries within each CSV file would be sorted in ascending order according to the 'timestamp' values.

3.2.1 Data Cleaning. For each data entry, we would need to verify that the 'device_id' matched with the corresponding 'device_label'. In other words, the device identified by the 'device_id' must be owned by the participant identified by the 'device_label'. If there was a mismatched between the two values, this data entry would be removed from the dataset.

For activity data, any data entry with a low 'confidence' value would be drop. If the data was collected from an Android device, the 'confidence' value would range from 0 to 100. All data entries with a 'confidence' value less than 50 would be removed from the dataset. If the data was collected from an iOS device, the possible range for 'confidence' would be from 0 to 2. Data entries with a 'confidence' value less than 1 would be removed.

Additionally, data entries from each CSV file would be aggregated into hourly interval in later steps. After the aggregation, each CSV file would contain 24 rows of data entry, representing aggregated data from each hour of the day. However, for some dates, data for certain hours may be missing. If there was no data present for an hour, the data entry would be filled with the value 0.

3.3 Feature Extraction

New features were extracted from the ones provided by AWARE for the following types of data: call, audio, activity, and screen.

3.3.1 Call Data. For call data, there were five features extracted, including 'num_incoming', 'num_connected', 'num_dialing', 'num_disconnected', and 'num_call'. Most of these five features were extracted from 'call_type', which indicated the types of calls made from the participants' devices. If the 'call_type' value of a data entry was 1, its 'num_incoming' value would be set to 1 while the values for other extracted features, except 'num_call' would be set to 0. The same process would be done for other 'call_type' values. With value 2, the 'num_connected' value would be set to 1 for the entry. If the value was 3, 'num_dialing' would be set to 1. For value 4, 'num_disconnected' would be set to 1. For all

call data entries, the 'num_call' would always be set to 1. After data aggregation, these five features would reflect the number of incoming calls, connected calls, dialing or outgoing calls, disconnected call, and the total number of all types of calls.

3.3.2 Audio Data. Using the 'double_convo_start' and 'double_convo_end' features, we calculated the 'convo_length' value for each audio data entry. The 'convo_length' feature measured the length of each conversation recorded by the device. This value for this feature was computed by subtracting the 'double_convo_start' value from the double_convo_end' value. Another feature that was extracted for the audio data was 'num_convos'. The feature indicated the number of conversations recorded over a one-hour interval. The value of this feature would be set to 1 for any data entry with a 'convo_length' value greater than 0. Otherwise, it would be 0.

3.3.3 Activity Data. The activity data had different set of features depending on the device type. For example, a data entry generated by an iOS device would contain features like 'automotive' and 'cycling' while a data entry created by an Android device would not include these features. During the feature extraction process, new features would be created only for the Android activity data. The main goal was to ensure that all activity data would contain a same set of features.

There were 6 features created for Android activity data, including 'automotive', 'cycling', 'running', 'stationary', 'unknown', and 'walking'. These 6 features were present in all iOS activity data, which helped to indicate a participant's activity at the time of data collection. These features were extracted from the 'activity_name' feature in Android activity data. If the 'activity_name' was equal to "still", the value of 'stationary' would be set to one and the values of the other 5 features would be set to 0. This same process would be done for 'walking', 'cycling', 'automobile', 'running', and 'unknown' when the value of 'activity_name' was equal to "on_foot", "on_bike", "on_vehicle", "running", or "unknown", respectively. Each data entry should only have one of these features set to 1, and the values for the other five features must remain 0.

3.3.4 Screen Data. For screen data, there were two new features extracted: 'num_pickups' and 'total_screentime'. The 'num_pickups' feature was based on the 'screen_status' feature. The 'screen_status' feature indicated the state of the device. A 'screen_status' value of 0 would indicate that the device's screen was off, and 1 would indicate that the screen was on. A value of 2 and 3 would indicate locked or unlocked screen, respectively. For each data entry, if the 'screen_status' value was 1, the 'num_pickups' would be set to 1; otherwise, it would be 0. The 'total_screentime' was computed based on the 'screen_status' and 'timestamp' features. For each entry with a 'screen_status' of 1, we would find the nearest data entry before the current one with a 'screen_status' value of 0. Then, we would find the absolute value of the difference between the 'timestamp' values of these two entries. This difference would indicate the total amount of time in milliseconds that the screen was on. The difference in 'timestamp' values would become the 'total_screentime' value for the data entry with a 'screen_status' of 1. If no data entry with 'screen_status' of 0 was found before the current one, 'total_screentime' would be set to 0. For other entries with 'screen_status' not equal to 1, the 'total_screentime' would also be set to 0.

3.4 Data Aggregation

The data was aggregated into hourly intervals once the feature extraction process was completed. Based on the 'timestamp' values, we were able to determine the hour of the day at which a data entry was collected. During earlier data processing steps, the data entries were already separated based on their collection date; thus, for data entries in

each CSV file, we would aggregate all data entries generated at the same hour of the day. The aggregation methods used in this process included finding the sum, mean, mode, or number of entries collected during each hourly interval. The aggregation process is described below:

- Call: Find the sum for 'call_duration', 'call_type': 'num_incoming', 'num_connected', 'num_dialing', 'num_disconnected', 'num_calls'. Find the mode for 'call_type' and 'trace'.
- Location: Find the mode for 'provider'. Use mean for aggregation for these features: 'accuracy', 'double_altitude', 'double_bearing', 'double_latitude', 'double_longitude', and 'double_speed'.
- Audio: Find the mode for "datatype" and "inference". Find the mean value for these features: 'double_convo_end', 'double_convo_start', 'double_energy', and 'convo_length'. Find the sum for 'num_convos'.
- Activity: Find the sum for 'cycling', 'stationary', 'running', 'walking', 'automotive', 'unknown'. Compute the mean value for the 'confidence' feature.
- Screen: Find the sum for 'num_pickups' and 'total_screentime'.
- Wi-Fi: Count the number of unique 'bssid'. Find the mode for 'security' and 'ssid'. Find the mean value for "frequency" and 'ssid'.
- Sensor Wi-Fi: Count the number of unique 'bssid'. Find the mode for 'mac_address' and 'ssid'.

For each type of data, some features were not included in the aggregated data, such as the 'timestamp' and 'device_label' features. These dropped features were not mentioned above.

3.5 Image Generation

Another step of the data processing was to convert the data into images. Once the raw data were separated into dates and aggregated into hourly intervals, each data value was scaled to be between 0 and 255, representing pixel values. Each data table contained 24 rows, corresponding to the 24 hours of a day, and the column dimension varied based on the data features available for each sensor/plug-in type. The dimension of the images would correspond with the dimension of the data tables. However, since the column dimension differed for each data type, images were resized to be 32 x 32.

Figure 2 contains an example images generated using the activity data collected from user 25BY, which is shown in table 1. The figure contains some color spots that are noticeably "lighter" compared to others. The color difference was caused by the original data values. During the image generation process, the data value was directly converted into pixel values. In this case, a higher original value would create a deeper color pixel. Additionally, it was apparent that most of the color spots formed columns. These columns corresponded to the features contained in the original dataset. As shown in table 1, the "stationary" feature contained the highest data values, which described the number of times the user was being still during a day; thus, the most clear column in figure 2 corresponded to the "stationary" feature. From the color difference, it was possible to determine which activity was more prevalent that other.

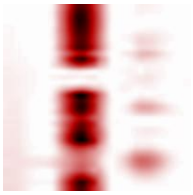

Fig. 2. Activity data image from user 25BY

cycling	stationary	running	unknown	walking	automotive
0	224	0	0	12	0
0	244	0	0	0	0
0	236	0	0	2	0
.	.	.	.	.	.
.	.	.	.	.	.
0	222	0	0	24	0
0	222	0	0	0	0

Table 1. Aggregated activity data from user 25BY

For supervised learning algorithms, the data should be labeled. In our study, the results of the weekly PHQ-9 and GAD-7 surveys were used to label the data. The PHQ-9 scores ranged from 0 – 27, and the scores were interpreted according to this guideline: minimal depression (0 – 4), mild depression (5 – 9), moderate depression (10 – 14), moderately severe depression (15 – 19), and severe depression (20 – 27). The GAD-7 scores ranged from 0 – 21 and were interpreted as follows: minimal anxiety (0 – 4), mild anxiety (5 – 9), moderate anxiety (10 – 14), and severe anxiety (15 – 21). Since the surveys were conducted on a weekly basis, all data collected between the previous survey and the current survey would be labeled using results from the current survey. For example, if a survey was submitted on 10/10/2020 and the previous survey was conducted on 10/03/2020, data that occurred between 10/10/2020 and 10/04/2020 would be labeled with the results from the 10/10/2020 survey.

3.6 Behavioral Image Clustering

In this study, a pre-trained neural network was used to extract a feature vector from the generated images. The feature vectors were used as inputs to the K-Means clustering algorithm to cluster based on individual behavior as well as group behavior.

3.6.1 Transfer Learning. The VGG-16 model from the Keras library was used to extract a feature vector from each generated image before clustering. VGG-16 is a pre-trained convolutional neural network that is considered state of the art for image recognition problems [10]. The VGG-16 model has 16 layers, however because in this study it was used as a feature extractor only, the final prediction layer was removed. Each generated image was reshaped to be of size 224 x 224 and used as an input for the VGG-16 model which outputted a vector of relevant features specific to one image. The feature vectors were stored and used as inputs for clustering methods described below.

3.6.2 Silhouette Score Analysis. K-Means clustering labels each data point with a cluster number. The number of unique clusters is an input parameter to K-Means and must be tuned to match the data. In both the clustering of individual behavior and the clustering of group behavior, the optimal number of clusters was determined by using silhouette score analysis. The silhouette score algorithm from the Scikit Learn metrics library was used in this study. Silhouette score analysis is performed by calculating the mean intra-cluster distance a and the mean nearest-cluster distance b for each image after clustering with a specific k value. A silhouette score is determined by taking the average of $\frac{b-a}{max(a,b)}$ for all samples. The candidates for the number of clusters, k, were $k \in [2, 9]$. Each instance of clustering, the data was clustered with every candidate k value and the silhouette score was calculated and stored. Only the clustering output with the parameter value k which corresponded to the highest silhouette score was reported in the results.

3.6.3 Clustering of Individual Behavior. Images were sorted into buckets where each bucket only contained images that corresponded to one user and one sensor type. For example, all images in one bucket belonged to the user with device label '203BN' and were generated from the 'audio' sensor. The VGG-16 model described above extracted a features vector for the images in each bucket. The Scikit Learn K-Means algorithm was then used to along with silhouette score analysis to produce a clustering of the data in each bucket. The results of this clustering were used to analyze the behavior patterns of each user individually per each sensor.

3.6.4 Clustering of Group Behavior. Images were also sorted into buckets where each bucket only contained images that corresponded to one sensor. For example, all images in one bucket were generated from the 'audio' sensor but belonged to all users. The VGG-16 model described above extracted a features vector for the images in each bucket. The scikit learn K-Means algorithm was then used to along with silhouette score analysis to produce a clustering of the data in each bucket. The results of this clustering were used to analyze behavior patterns of all users per each sensor.

3.7 Behavioral Image Classification

A convolutional neural network, typically used for image classification, was also utilized to analyze the correlation between user activity and mental health. Although convolutional neural networks are not typically used for behavior modeling, they offer a constructive approach to learning that allows it to learn from a limited number of examples [11]. Specifically, the convolutional neural network (CNN) was used to determine the optimal set of input sensors for predicting a user's mental health. The CNN used for this research was made of the following layers: a batch normalization layer, a layer normalization layer, multiple max pooling layers, multiple 2D convolutional layers, a dropout layer, a flattening layer, a 'relu' activation layer, and a 'softmax' activation layer. The model was compiled using the 'nadam' optimizer and used sparse categorical cross entropy to calculate loss. For use in this model, the data was labeled using the user's depression or anxiety classification for that week. The data was in the form of 32 x 32 images, and each image represented one day of a user's activities collected from one sensor. Two testing methods were used to determine which sensors provided the best information for predicting the depression or anxiety levels of a user: a regular test using a 80%-20% train-test split of the data and then another test using leave-one-out cross-validation. For both of these tests, the CNN attempted to classify test data based on the data that was used to train it.

To begin, the 80%-20% split was tested for all sensor combinations. The model was trained using 80% of the available data and was tested using 20% of the available data. The available data for each test was the entirety of the data provided by each of the input sensors. Each combination of sensors was run for 20 epochs. Each sensor combination was tested with two sets of categories: the anxiety categories provided by the GAD-7 survey and the depression categories provided by the PHQ-9 survey. The final accuracy of the model was used to determine how effective the combination of sensors was at predicting a user's mental health in regards to anxiety or depression. The accuracy for these tests were low due to the relatively small dataset.

Consequently, leave-one-out cross-validation was used for testing. For these tests, there were some adjustments made to the method. First, the split of testing and training data was changed to use the leave-one-out method. This involved excluding one user's data from the training data, and using it for the test data. The accuracy for this test is noted and then the process is repeated with another user. Once all the users have been excluded once, the average accuracy from all the tests is calculated. This process was utilized to accommodate for the relatively small amount of training data and because it is a fairly accurate predictor of the accuracy of a model [12]. Another change was with the combinations of input sensors that were used. For the cross-validation tests, a maximum of three sensors were used as

an input. There were two reasons for this decision. First, during the testing of the 80%-20% split it was found that using four or more sensors degraded the performance of the model sharply. Second, using leave-one-out cross-validation was very time consuming due to the large amount of tests that had to be run for each combination of sensors. Furthermore, some sensors had separate versions for iOS and Android, such as 'ios_activity' and 'google_activity.' These sensors were considered as one sensor for the purposes of this test. The final change made to the method for cross-validation was reducing the number of epochs from 20 to 10. This change was made to reduce runtime and because there was not a large increase in accuracy between the 10th and 20th epoch.

4 RESULTS

4.1 Behavioral Image Clustering

4.1.1 Individual Clustering. All images belonging to a single user generated from a single sensor were clustered according to the methods above. Table 2 shows the average number of clusters reported across all users.

Table 2. Average Clusters

Sensor	# of Clusters
screen	1.92
ios_activity	2.05
google_activity	1.88
audio	2.26
audio_android	2.50
locations	3.44
calls	2.65

An example of the produced clustering is illustrated in the following figures, depicting the differences in two user's behavior. For user 49ZR in Figure 3, the high intensity of values found solely in the middle column of the first cluster represent the sedentary behavior of the user while the second cluster illustrates records of walking and automotive behavior as found in the last columns, respectively. However, the clustering of user 203BN depicts no clear differences and thus the basis of such clustering cannot be determined. Nevertheless, 203BN illustrates more consistent sedentary behavior in comparison to 49ZR.

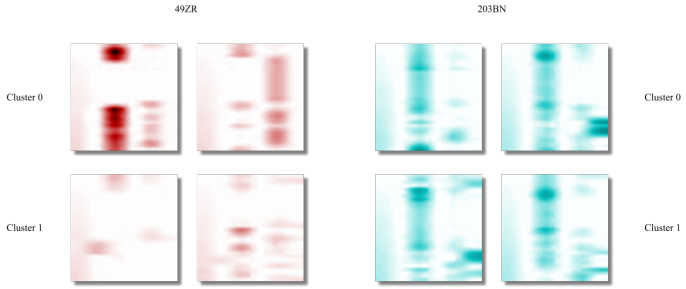

Fig. 3. Activity Clustering Comparison

The clustering results in figure 4 depicts audio differences based on the recorded hour of intense energy, the amplitude of the audio signal. This is evident for both users.

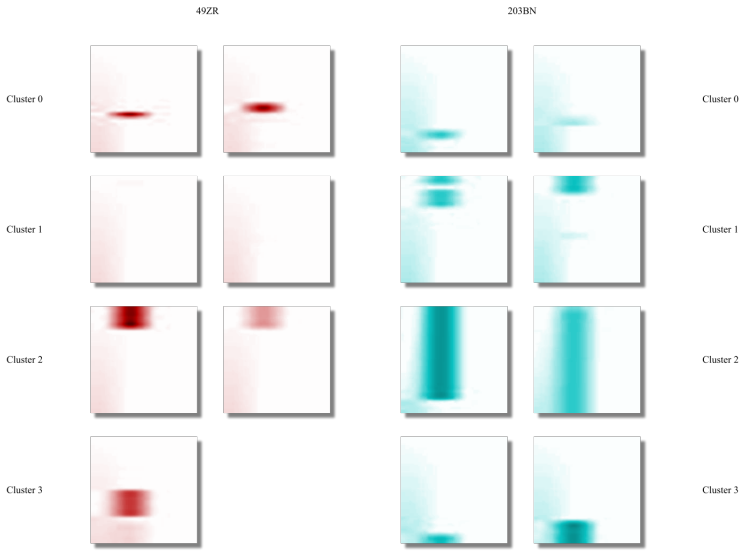

Fig. 4. Audio Clustering Comparison

Figure 5 illustrates User 49ZR and 203BN's calls behaviors in which slight differences can be found. User 49ZR clusters based on the number of calls present, found near the right edge of the images. User 203BN, however, clusters

based on the number of calls as present in the second cluster as well as the call type, shown in the third cluster. Both users have similar call durations in comparison to each other as well as across each user's images and thus, were not the basis of each user's clustering.

Fig. 5. Calls Clustering Comparison

Figure 6 shows the clustering based on the two user's location data. Although the values are high, seen by the intensity of the images, the clustering depicts differences based on available location recordings. Thus, the clustering did not utilize the values of location, but the presence of data and as a result, much cannot be determined.

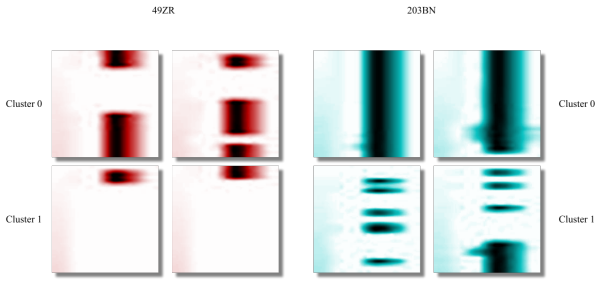

Fig. 6. Location Clustering Comparison

The final figure describes differences in the screen time of each user. The left column represents the number of times the user picked up their device while the right side shows the total screen time. User 49ZR exhibits increased screen time near the end of the day, as shown in the second cluster. However, no clear behavioral differences can be found for user 203BN.

Fig. 7. Screen Clustering Comparison

4.1.2 Group Clustering. All images from every user belonging to the same sensor were clustered according to the methods above. This was repeated for every sensor and the number of clusters with the highest silhouette score are reported in table 3. An example of such clustering is illustrated below in Figure 8 and the images found in the second row

Table 3

Sensor	# of Clusters	Silhouette Score
screen	2	0.2908
ios_activity	2	0.1235
audio	2	0.5191
locations	2	0.4196
calls	2	0.5389
audio_android	2	0.4624
google_activity	2	0.3221

represent the second of two clusters. The screen data depicts a difference in total screen time between the two clusters as screen time is scattered throughout the day in the first cluster while the second cluster shows more concentrated screen time near the end of the day. Activity data was clustered based upon the recordings of sedentary behavior as the second cluster contained more recordings of such behavior as opposed to the first. The audio data is evidently clustered based upon high energy levels found in the beginning of the day, present in the second cluster. Location data was clustered according to the consistency of longitude and latitude data. The second cluster contains a full column of data while the first cluster contains breaks in recordings. Finally, the calls data is clustered based on the call type, found in the second column of each image. Images found in the second cluster contain high call type values, representing missed calls present.

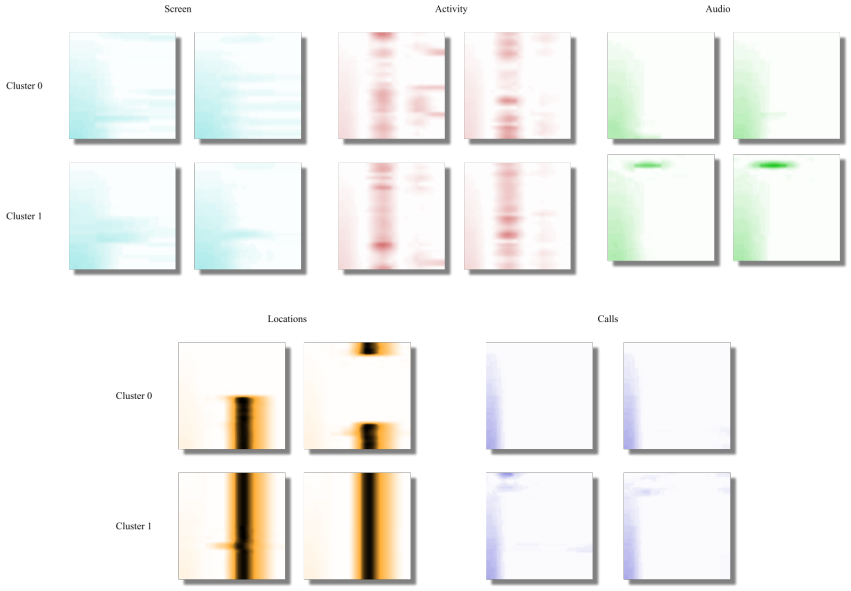

Fig. 8. Sample Clustered Images of All Sensors

When analyzing the results of this clustering it was observed that some users had a majority of images from a sensor belonging to the same cluster. This indicates that such a users behavior, as measured by the sensor was consistent between days. Other users had images evenly split between clusters, suggesting that their behavior was inconsistent between days.

Table 4 shows the portion of images belonging to a single cluster for two different users. User 49ZR's images were more likely to belong to the same cluster, meaning user 49ZR's behavior was more consistent across different days. User 203BN's images were more likely to be split evenly between clusters, meaning user 203BN's behavior was less consistent across different days.

Table 4. Portion of images belonging to dominant cluster

Sensor	49ZR	203BN
calls	.90	.90
locations	.99	.82
audio	.78	.58
ios_activity	.94	.53
screen	.78	.60

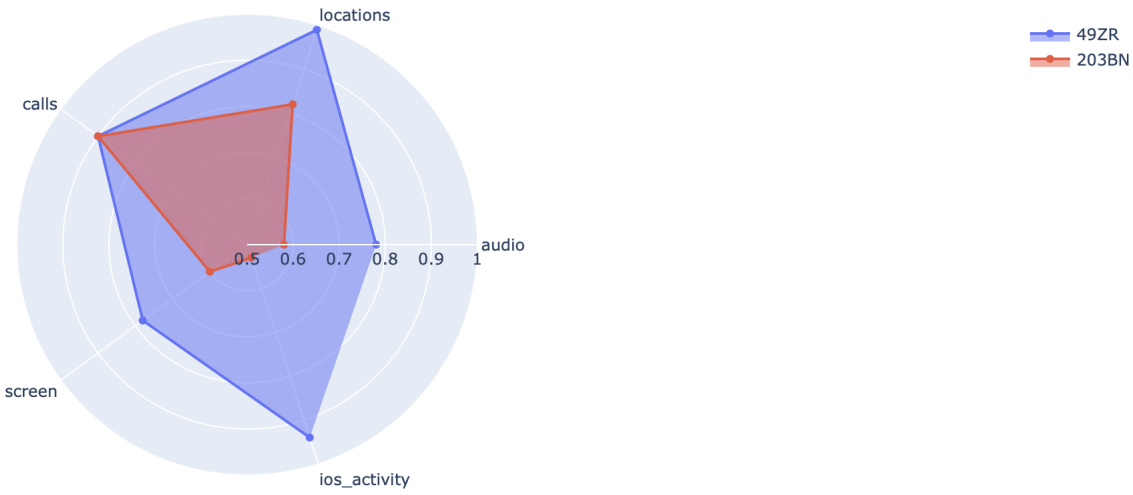

Fig. 9. Portion of images belonging to dominant cluster

4.2 Behavioral Image Classification

The results for both CNN approaches are shown in this section. The first approach utilized a 80%-20% split of the data into training and testing data. It was thought that using 80% of the available data for training would be enough to produce accurate results, however this was not the case. Consequently, leave-one-out cross-validation was then used to calculate the accuracy of the models, to accommodate for the low volume of training data.

4.2.1 Train-Test Split Results.

The results in table 5 and table 6 were acquired while using a 80%-20% split of train and test data to calculate the accuracy of the CNN. The five input combinations with the highest accuracy are shown in the table below in descending order. The results for both the GAD-7 categories and the PHQ-9 categories are shown.

Table 5. GAD Results

Rank	Inputs	Accuracy
1	google_activity	0.5652
2	audio_android, google_activity	0.5124
3	audio_android	0.4712
4	audio, google_activity	0.4702
5	ios_activity, google_activity, audio_android	0.4439

Table 6. PHQ Results

Rank	Inputs	Accuracy
1	google_activity	0.6232
2	audio_android, google_activity	0.5992
3	audio_android	0.5481
4	audio, audio_android, google_activity	0.5193
5	ios_activity, google_activity, audio_android	0.4849

4.2.2 *Cross-Validation Results.*

The results contained in table 7 and table 8 were acquired while using leave-one-out cross-validation to calculate the accuracy of the CNN. The 5 input combinations with the highest accuracies are shown in the table below in descending order. The results for both the GAD-7 categories and the PHQ-9 categories are shown. For these results, note that both ios_activity and google_activity are considered the same sensor. This also applies to audio_android and audio.

Table 7. GAD Results

Rank	Inputs	Accuracy
1	ios_activity, google_activity	0.9517
2	ios_activity, google_activity, screen	0.8820
3	screen	0.8809
4	screen, audio_android, audio	0.8348
5	ios_activity, google_activity, screen, audio_android, audio	0.8258

Table 8. PHQ Results

Rank	Inputs	Accuracy
1	ios_activity, google_activity	0.9287
2	screen	0.9074
3	ios_activity, google_activity, audio_android, audio	0.8685
4	ios_activity, google_activity, screen	0.8677
5	ios_activity, google_activity, screen, audio_android, audio	0.8528

5 CONCLUSION

Using a combination of VGG16 feature extraction and K-Means clustering on sensor data reveals differences in user's behavior patterns as well as the ability to recognize a user's tendencies. The silhouette scores in table 3 demonstrate which sensor's reveal the most significant behavioral differences in this data. For example, images in different clusters of the call data were more different than images in different clusters of the ios_activity data. Further analysis of a user's distribution of images across clusters is an effective method to determining if the user acts consistently or inconsistently across different days.

The results produced by the CNN show that it is a good predictor of anxiety and depression if these two conditions are met: enough data is provided to the model and the correct sensors are provided as an input. The CNN tended to provide more accurate predictions when more data was provided. This is seen when comparing the results in table 5 with the results in table 7. The results in table 5 utilize a 80%-20% split of training and test data while the results in table 7 uses the leave-one-out method. The leave-one-out method provides more data as it includes every user's data except for one, demonstrating the relationship between amount of data and accuracy. This relationship is not exclusive to this project. Models need data to inform their predictions, so it is not surprising that providing more information to the model allows it to perform better.

The second condition for accurate predictions is that the correct sensors have to be used as the input. This conclusion can be deduced by looking at any of the prior CNN results tables. Activity and audio data is found near the top of all the results in terms of accuracy. There could be many reasons for this correlation. However, the number of activities someone will partake in and the number of conversations they have will vary greatly depending on how that person feels. The model was able to pick up on these differences and used them to predict a person's mental state. Other sensors such as 'locations,' which provided location data, and 'calls,' which provided call data, did not demonstrate great usefulness when classifying data. The correlation between data collected from these sensors and a person's mental state was weaker. Once again, there could be a variety of reasons for this weak correlation, but the number of calls a person receives/makes and the locations they visit are not impacted greatly by how they feel. When these two conditions are fulfilled, a CNN can be used as a good predictor of the anxiety or depression levels of people.

As a result, modeling behavior data as images can be applied to clustering methods or CNNs for analysis or prediction, respectively. However, more success can potentially be found with more complete data. Many hours of data were missing and thus, when aggregated, rows of the images were blank. Other opportunities of future work include the application of transfer learning in behavior image classification, different clustering methods, and the use of behavior images for other machine learning applications.

6 REFERENCES

[1] Srividya, M., Mohanavalli, S. and Bhalaji, N., 2018. Behavioral Modeling for Mental Health using Machine Learning Algorithms. *Journal of Medical Systems*, 42(5).

[2] Altun, K., Barshan, B. and Tunçel, O., 2021. *Comparative study on classifying human activities with miniature inertial and magnetic sensors.*

[3] Chikersal, P., Doryab, A., Tumminia, M., Villalba, D., Dutcher, J., Liu, X., Cohen, S., Creswell, K., Mankoff, J., Creswell, J., Goel, M. and Dey, A., 2021. Detecting Depression and Predicting its Onset Using Longitudinal Symptoms Captured by Passive Sensing. *ACM Transactions on Computer-Human Interaction*, 28(1), pp.1-41.

[4] Hammen, C., Kim, E.Y., Eberhart, N.K. and Brennan, P.A., 2009. Chronic and acute stress and the prediction of major depression in women. *Depress. Anxiety*, 26, pp.718-723.

[5] O'Connor RC, Rasmussen S, Hawton K., 2010. Predicting depression, anxiety and self-harm in adolescents: the role of perfectionism and acute life stress. *Behav Res Ther*,. 48(1), 52-9.

[6] Kendler KS, Kuhn JW, Vittum J, Prescott CA, Riley B., 2005. The Interaction of Stressful Life Events and a Serotonin Transporter Polymorphism in the Prediction of Episodes of Major Depression: A Replication. *Arch Gen Psychiatry*, 62(5), pp.529–535.

[7] Ferreira, D., Kostakos, V. and Dey, A., 2015. AWARE: Mobile Context Instrumentation Framework. *Frontiers in ICT*, 2.

[8] Kroenke, K. and Spitzer, R., 2002. The PHQ-9: A New Depression Diagnostic and Severity Measure. *Psychiatric Annals*, 32(9), pp.509-515.

[9] Williams, N., 2014. The GAD-7 questionnaire. *Occupational Medicine*, 64(3), pp.224-224.

[10] Huilgol, P., 2020. Top 4 Pre-trained Models for Image Classification with Python Code. *Analytics Vidhya*.

[11] Spaanenburg L., Tehrani M.A., Kleihorst R., Meijer P.B.L. (2009) Behavior Modeling by Neural Networks. In: Alippi C., Polycarpou M., Panayiotou C., Ellinas G. (eds) Artificial Neural Networks – ICANN 2009. ICANN 2009. Lecture Notes in Computer Science, vol 5768. Springer, Berlin, Heidelberg.

[12] Evgeniou, T., Pontil, M. Elisseeff, A., 2004. Leave One Out Error, Stability, and Generalization of Voting Combinations of Classifiers. *Machine Learning*, 55, pp.71–97.