CSE 535: Mobile Computing

Neuromusic Android Application

Naresh Balusu
ASU ID: 1210434755
nbalusu@asu.edu

Avinash Reddy Kaitha
ASU ID: 1210439630
akaitha@asu.edu

Gaurav Ramdasi
ASU ID: 1206467752
gramdasi@asu.edu

ABSTRACT

We have built a music app which plays songs based on the mood of user. The mood is detected by the Neurosky Wearable Sensor. we also provided several features like creating playlist based on moods, getting users preferences from Facebook, sharing playlists.

KEYWORDS

Android, Neurosky Mindwave, Matlab,Higuchi Fractal Dimension algorithm

INTRODUCTION

This is a mobile application which uses NeuroSky sensor data that helps in changing its context. We used android services to play music, to retrieve data from data bases, storing user preferences etc and SQLite databases to store data locally and MySql for Fog server. It learns from user reactively. To get sensor data we used thinkgear android APIs. The app should work on Android phones with Android API Level 19 + (KitKat and above).

IMPLEMENTATION AND ALGORITHMS USED

In this application, we have a neurosky device which is paired to the android phone via bluetooth and the device sends various signals to the device. We have used Facebook apis for user authentication. The basic idea is to predict the user's mood based on the signals received. So we have built a model for this in Matlab. We have tried to incorporate this code in to the android app by using Matlab coder to generate C code and then use android idk to call the functions from java. But this was taking a lot of memory and all the functions weren't portable. So we had to use a fog server. We post our signal data using php scripts and then call Matlab functions from the server using "exec" command in php.Matlab processes the input data based on the training data in the fog server and sends the result back to the mobile device. Now we display our predicted mood to the user. He can change it if he feels its wrong and this result will be added to our training data in the server. This way our training data set keeps growing and gives better results. We use musicovery APIs to get song names based on the current mood. Then we use soundcloud APIs to stream this music. Our app has a database which stores the user info and playlists.

We referred to two research papers to build our Mood prediction algorithm. First one is Song induced mood recognition system where they have analyzed the change in moods due to listening to songs. Here they classified the

signals based n type of song the user was listening .
The next one is Real time EEG Based Emotion Recognition. Here they propose a Fractal dimensional model for mood recognition. Using Higuchi Fractal Dimension Algorithm, they found the fractal dimension values from different electrodes. They selected particular electrodes in their sensor and called them arousal and valence. In our prediction algorithm, since we do not have multiple electrodes to implement the valence arousal model, we combined these two papers to build a new algorithm. Since we know listening to music affects our mood and the EEG signals, we will initially have a training phase. This happens when we run the app for the first time, It picks one song for each mood and user has to listen to all the songs while wearing the headset. Since we know the label and the FD value calculated from HFD, we add this to our fog server database. This will be our initial training data set. After training phase, we use Naive Bayes classifier to classify the new incoming data and then add that back to training dataset as mentioned earlier.

Higuchi Fractal Dimension Algorithm:

Let $X(1), X(2), \ldots, X(N)$ be a finite set of time series samples, the new time series is constructed as follows:

$$X_k^m : X(m), X(m+k), X(m+2k), \ldots, X(m+[\frac{N-m}{k}] \cdot k) \, . \qquad (1)$$

where $m = 1, 2, \ldots, k$, m is the initial time and k is the interval time.
Then, k sets of $L_m(k)$ are calculated as follows:

$$L_m(k) = \frac{\left\{ (\sum_{i=1}^{[\frac{N-m}{k}]} |X(m+ik) - X(m+(i-1) \cdot k)|) \frac{N-1}{[\frac{N-m}{k}] \cdot k} \right\}}{k} \, . \qquad (2)$$

$\langle L(k) \rangle$ denotes the average value over k sets of $L_m(k)$ and the relationship exists as follows:

$$\langle L(k) \rangle \propto k^{-D} \, . \qquad (3)$$

Finally, the fractal dimension can be obtained by logarithmic plotting between different k and its associated $\langle L(k) \rangle$ [17].

Complete Algorithm:
Training Phase:
Get sensor data from Neurosky
Send data to fog server with labels
Calculate FD values using HFD algorithm
Store the FD value with the mood label in the server.

Testing Phase:
Get sensor data from Neurosky
Send data to fog server
Calculate FD values using HFD algorithm
Classify the new value using Naive Bayes classifier
Send the FD value and predicted mood back to the user.
Update final FD value and the mood label to the fog server.

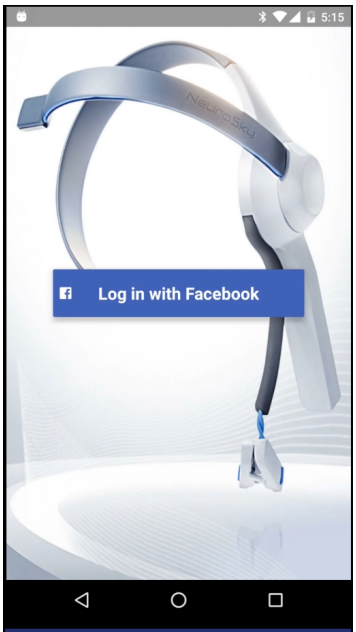
User Database:
The application has a user database which stores data acquired from Facebook APIs like name, email id, cover pic, display pic etc. It also stores playlist tracks.

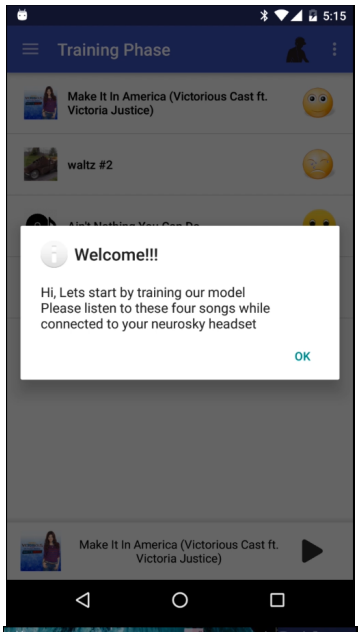
Fog Server Database:
There is a fog server with one table which stores the FD values and the mood label. This is used as the training data set for the Naive Bayes Classifier. This table keeps growing as the user uses the application.

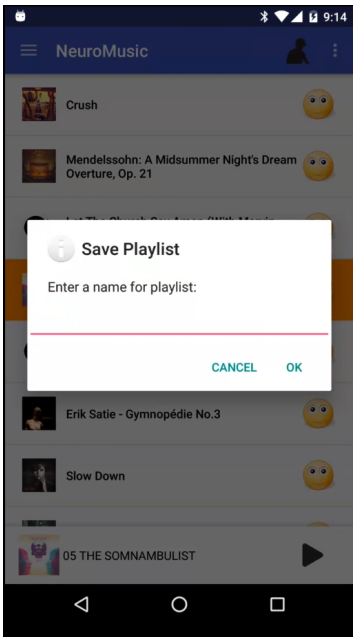
Matlab Implementation:

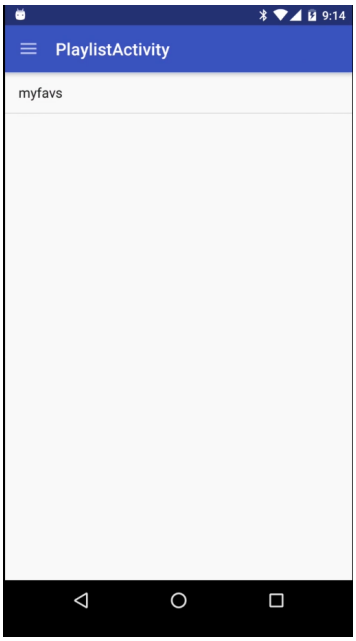
```
function train(D,mood)
dbname = 'eeg';
username = 'root';
password = '';
driver = 'com.mysql.jdbc.Driver';
dburl = ['jdbc:mysql://localhost:3306/' dbname];

conn = database(dbname, username, password, driver, dburl);
conn.message

 Result = hfd(D,256)
  colnames = {'FD','Mood'};
 data = {Result,mood};
 datainsert(conn,'Raw_FD',colnames,data);
```

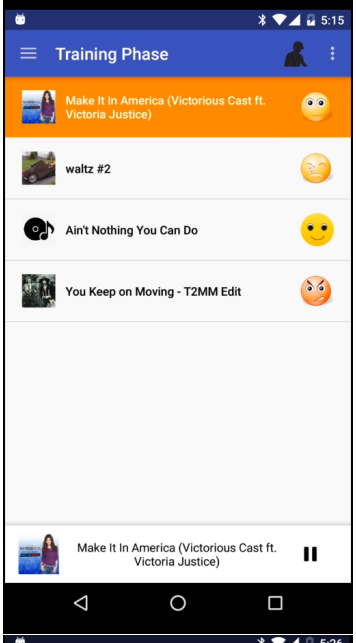
APPLICATION SCREENSHOTS

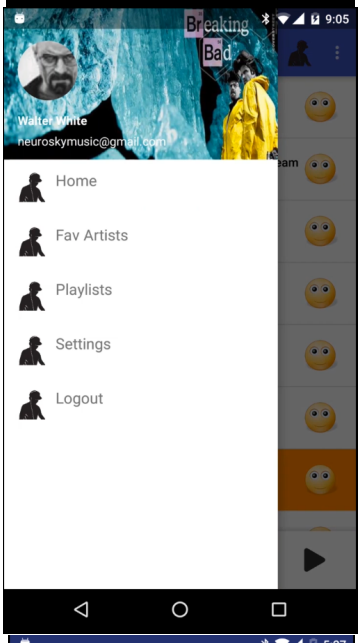


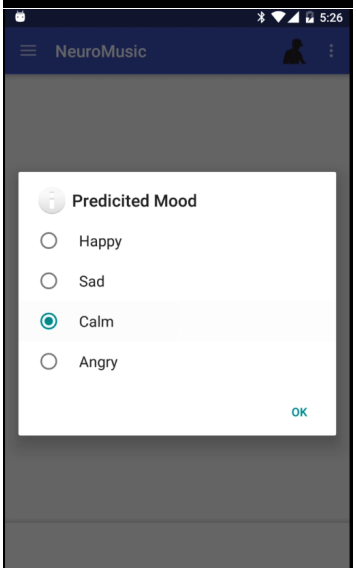


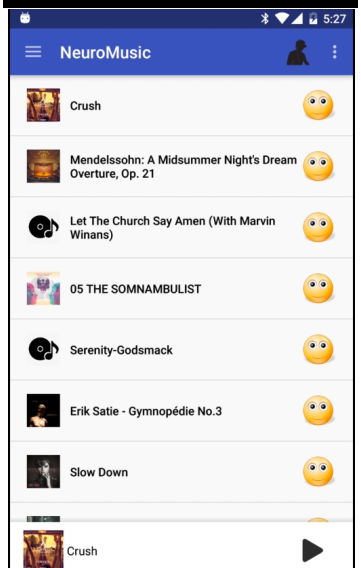












TASK COMPLETION STATUS

We have successfully completed all the 15 tasks and then completed 4 extra tasks. The contribution table is as follows:

S.No	Task	Assignee
1	**User's DataBase**	Avinash
2	**Sensor data**	Gaurav
3	**Music Metadata**	Naresh
4	**App database**	Naresh
5	**Data from Facebook**	Avinash
6	**Reactive Learning**	Naresh
7	**Music player:**	Gaurav
8	**Background Service**	Gaurav
9	**Playlist:**	Avinash
10	**Share playlist**	Gaurav
11	**Playing music based on mood:**	Naresh
12	**Offline play**	Avinash
13	**Recommendations**	Naresh
14	**Mood based radio with out sensors**	Gaurav
15	**Settings**	Avinash
16	**Learning from server to Matlab**	Avinash
17	**Inserting data from matlab to database**	Naresh
18	**Sending data from Matlab to Android**	Avinash
19	**Update training data from Android**	Gaurav

CONCLUSION

We have developed an application which can predict the mood of the user based on EEG signals. The neurosky mind wave device has only one sensor and sends erroneous data so predicting could be difficult. We can use other data like the alpha, beta and theta signals from the device to make our prediction model even better.

ACKNOWLEDGEMENTS

We thank our Professor Dr. Ayan Banerjee and our TA Junghyo Lee for all the support and valuable guidance during the building phase of the project. We would also like to thank Mr. Koosha Sadeghi for helping us while working with neurosky mind wave

REFERENCES

1. Yisi Liu, Olga Sourina, and Minh Khoa Nguyen.: Real-time EEG-based Emotion Recognition and its Applications

2. G.B. Janvale, B.W. Gawali, Rakesh S. Deore, Suresh C. Mehrotra, Sachin N. Deshmukh, Arun V. Marwale.: Songs induced mood recognition system using EEG Signals

3. http://developer.android.com

4. http://square.github.io/retrofit/

5. https://github.com/google/gson

6. https://github.com/umano/AndroidSlidingUpPanel

7. https://developers.facebook.com